S/T3 08/17

Département de formation doctorale en informatique

École Doctorale IAEM Lorraine
UFR MIM

Contributions à la dissémination d'informations haptiques dans un environnement multimodal

THÈSE

présentée et soutenue publiquement le 2 décembre 2008

pour l'obtention du

Doctorat de l'Université Paul Verlaine — Metz
(spécialité informatique)

par

Thomas Pietrzak

Composition du jury

Directeur de thèse :	Maurice Margenstern	Professeur à l'Université Paul Verlaine – Metz
Encadrants :	Isabelle Pecci	Maître de conférences à l'Université Paul Verlaine – Metz
	Benoît Martin	Maître de conférences à l'Université Paul Verlaine – Metz
Rapporteurs :	Vincent Hayward	Professeur à l'ISIR – Paris
	Yacine Bellik	Maître de conférences HDR à l'Université Paris-Sud – Orsay
Examinateurs :	Bertrand David	Professeur à l'École Centrale de Lyon
	Yves Guiard	Directeur de recherche CNRS à l'ENST – Paris

Laboratoire d'Informatique Théorique et Appliquée — EA 3097

LITA
Laboratoire d'informatique théorique et appliquée



AVERTISSEMENT

Cette thèse est le fruit d'un long travail approuvé par le jury de soutenance et disponible à l'ensemble de la communauté universitaire élargie.
Elle est soumise à la propriété intellectuelle de l'auteur au même titre que sa version papier. Ceci implique une obligation de citation, de référencement dans la rédaction de tous vos documents.
D'autre part, toutes contrefaçons, plagiats, reproductions illicites entraînent une poursuite pénale.
Enfin, l'autorisaton de diffusion a été accordée jusqu' à nouvel ordre.

- Contact SCD Metz : daniel.michel@scd.univ-metz.fr

Ecrire au doctorant:

Code de la Proriété Intellectuelle. articles L 122. 4
Code de la Proriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Table des matières

Introduction xvii

CHAPITRE I L'interaction et la déficience visuelle 1

1. Introduction . . . 1
2. Les interfaces . . . 2
   - 2.1. Modèles d'architectures logicielles des systèmes interactifs . . . 2
     - 2.1.1. Modèles à couches . . . 2
     - 2.1.2. Modèles à agents . . . 3
     - 2.1.3. Conclusion . . . 6
   - 2.2. Multimodalité . . . 6
     - 2.2.1. Les espaces de conception . . . 7
     - 2.2.2. Coopération entre modalités . . . 11
     - 2.2.3. Conclusion . . . 11
3. L'haptique . . . 12
   - 3.1. La somesthésie . . . 14
     - 3.1.1. Les mécanorécepteurs . . . 15
     - 3.1.2. Les thermorécepteurs . . . 16
     - 3.1.3. Les nocicepteurs . . . 16
     - 3.1.4. Conclusion . . . 16
   - 3.2. Taxonomie des périphériques haptiques . . . 16
     - 3.2.1. Les périphériques tactiles . . . 17
     - 3.2.2. Les périphériques à retour de force . . . 21
     - 3.2.3. Conclusion . . . 26
   - 3.3. Interactions haptiques . . . 27
     - 3.3.1. Simulations . . . 27
     - 3.3.2. Navigation d'interfaces . . . 28
     - 3.3.3. Exploration . . . 28
     - 3.3.4. Les icônes . . . 29
     - 3.3.5. Conclusion . . . 33
   - 3.4. Conclusion . . . 34
4. Les systèmes pour déficients visuels . . . 34

4.1. Méthodes de présentation et d'exploration 35
4.2. Exploration non visuelle 36
4.2.1. Objets 36
4.2.2. Graphiques 37
4.2.3. Interfaces 43
4.2.4. Scènes 3D 43
4.3. Systèmes haptiques pour l'enseignement 44
4.3.1. MICOLE 45
4.3.2. Autres systèmes 47
4.4. Conclusion 47
5. Conclusion 48

CHAPITRE II Icônes tactiles **51**

1. Introduction 51
2. Le concept d'icônes 52
2.1. Définitions 52
2.2. Paramètres et dimensions des Tactons 53
3. Les Tactons mono-dimensionnels 54
3.1. Méthodologie 55
3.2. Expérience 1 : comparaison de Tactons statiques, clignotants et vagues 56
3.3. Expérience 2 : plus de picots, et formes grossissantes 59
3.4. Expérience 3 : tests in situ 61
3.5. Expérience 4 : amélioration des vagues 63
3.6. Expérience 5 : Tactons mixtes 65
3.7. Conclusion 67
4. Les Tactons multi-dimensionnels 68
4.1. Conception 69
4.2. Méthodologie 71
4.3. Résultats 72
4.4. Discussion et conclusion 74
5. Exploration de formes géométriques 75
5.1. Modes d'exploration 75
5.2. Méthodologie 78
5.3. Résultats 81
5.4. Discussion 85
5.5. Conclusion 88
5.6. Travaux futurs. 88
6. Analogie musicale 89
7. Conclusion 92

CHAPITRE III Icônes kinesthésiques **97**

1. Introduction 97

2. Le concept de PICOB 98
2.1. Définitions 98
2.2. Implémentation 99
3. Les impulsions guidées 100
3.1. Conception 100
3.2. Méthodologie 101
3.3. Expérience 1 : reconnaissance de direction 102
3.4. Expérience 2 : reconnaissance d'amplitude 103
3.5. Expérience 3 : reconnaissance simultanée de direction et d'amplitude 105
3.6. Conclusion 108
4. Les impulsions semi-guidées 109
4.1. Conception 109
4.2. Méthodologie 110
4.3. Expérience 1 111
4.4. Expérience 2 113
4.5. Conclusion 115
5. Les séquences d'impulsions : PICOB 116
5.1. Conception 116
5.2. Méthodologie 117
5.3. Résultats 120
5.4. Discussion 122
6. Conclusion 124

CHAPITRE IV Applications multimodales **127**

1. Introduction 127
2. Architecture multimodale et collaborative MICOLE 128
2.1. Le système multi-agents 129
2.1.1. Les agents 130
2.1.2. Le bus 131
2.1.3. Les agents threadés 134
2.1.4. Le réseau 134
2.2. Le framework multimodal 135
2.2.1. Les agents d'entrée 136
2.2.2. Les agents de sortie 138
2.2.3. Les principaux agents 139
3. Logiciel d'exploration de schémas de circuits électriques 145
3.1. Premier prototype 146
3.1.1. Conception 146
3.1.2. Évaluation informelle 149
3.2. Second prototype 150
3.2.1. Modifications 150

3.2.2. Évaluation 151
3.3. Application finale 157
3.3.1. Modifications 158
3.3.2. Utilisation de l'architecture MICOLE 158
3.3.3. Techniques d'interaction 162
3.3.4. Évaluation 164
4. Conclusion 174

Conclusion **177**

Bibliographie **187**

Index **201**

ANNEXE A Questionnaires **205**

ANNEXE B Traces d'exploration **217**

Table des figures

CHAPITRE I L'interaction et la déficience visuelle 1

1. Modèle de Seeheim . . . 3
2. Modèle Arch . . . 4
3. Modèle MVC . . . 4
4. Modèle PAC et hiérarchie PAC . . . 5
5. Modèle PAC-Amodeus . . . 6
6. Espace de conception de Frohlich (Entrée) . . . 8
7. Espace de conception de Frohlich (Sortie) . . . 9
8. Espace CASE . . . 10
9. Extension à CASE . . . 10
10. Espace CARE . . . 12
11. Les mécanorécepteurs de la peau . . . 15
12. Souris Virtouch et VTPlayer . . . 17
13. Tactos, Tactiball, Tactipen et Tactitab . . . 18
14. Trois prototypes de VITAL . . . 18
15. Cybertouch . . . 19
16. STReSS, STReSS^2 et ABV . . . 20
17. Homonculus sensoriel . . . 20
18. Tongue Display Unit . . . 21
19. Gant Teletact et ComTouch . . . 21
20. Joystick à retour de force et volant à retour de force . . . 22
21. Virtuose et Cyberforce . . . 24
22. Pantograph, Pencat/Pro et souris Wingman Force Feedback . . . 25
23. Omega, Delta et Falcon . . . 25
24. Haptic Master, Cybergrasp et Haptic Workstation . . . 25
25. La vache haptique . . . 27
26. Tactor VBW32 et C2 . . . 30
27. Le haptic back display . . . 31
28. Périphérique utilisé pour les haptic icons . . . 32
29. Plage Braille . . . 35
30. Visage construit à partir de formes de base . . . 38

31. Exploration de camemberts avec Tac-tiles 40
32. Le système Tacis 42
33. Le Moose 43
34. Le système Homere 44
35. L'alphabet Braille 45

CHAPITRE II Icônes tactiles **51**

1. Tactons radiaux et diagonaux 54
2. VTPlayer et ses matrices de picots 55
3. Capture d'écran du logiciel de tests 55
4. Ensembles de Tactons utilisés dans l'expérience 1 56
5. Expérience 1 : erreurs et temps de réponse 58
6. Ensembles de Tactons utilisés dans l'expérience 2 59
7. Expérience 2 : erreurs et temps de réponse 60
8. Expérience 3 : erreurs et temps de réponse 62
9. Ensembles de Tactons utilisés dans l'expérience 4 63
10. Expérience 4 : erreurs et temps de réponse 65
11. Ensembles de Tactons utilisés dans l'expérience 5 65
12. Expérience 5 : erreurs et temps de réponse 67
13. Formes de Tactons utilisés pour les Tactons multi-dimensionnels 70
14. Erreurs de reconnaissance des dimensions pour chaque condition 73
15. Transmission d'information pour les conditions S_2, S_3 et T_4 74
16. Motifs utilisés pour l'exploration de formes géométriques 76
17. Guidage sur un segment 77
18. Formes proposées 80
19. Erreurs, confiance et temps d'exploration par condition 82
20. Erreurs par utilisateur par condition 82
21. Indice de confiance par utilisateur 83
22. Temps d'exploration par utilisateur 84
23. Confiance et temps d'exploration par réponse 85
24. Deux stratégies d'exploration de la forme 21 en condition Pv 87
25. Correspondance entre une image et un accord 91
26. Correspondance entre un Tacton dynamique et une mesure 91
27. Correspondance entre un Tacton statique et un point d'orgue 92

CHAPITRE III Icônes kinesthésiques **97**

1. Longueur et Amplitude d'une impulsion de direction et de forme fixée 98
2. PHANToM Desktop et OMNI 99
3. Longueur et Amplitude d'une impulsion de direction et de forme fixée 100
4. Expérience 1 : erreurs pour les différentes valeurs d'amplitude 103
5. Expérience 2 : erreurs 104

6. Expérience 3 : erreurs . . . 106
7. Expérience 3 : erreurs des conditions à 3 amplitudes . . . 107
8. Expérience 3 : information transmise . . . 108
9. Impulsion semi-guidée . . . 110
10. Expérience 1 : erreurs de direction . . . 112
11. Expérience 1 : information transmise . . . 112
12. Expérience 2 : erreurs de direction et d'amplitude . . . 114
13. Expérience 2 : information transmise . . . 114
14. Symétrie des séquences . . . 117
15. Séquences d'impulsions utilisées . . . 118
16. Nombre de fois que chaque séquence a été lue . . . 122
17. Confusion entre séquences ayant le même nombre d'allers et de retours . . . 123

CHAPITRE IV Applications multimodales **127**

1. Système de communication entre agents . . . 132
2. Exemple de message envoyé par un agent . . . 133
3. Exemple de binding créé par un agent . . . 134
4. Exemple d'agent threadé . . . 135
5. Fonctionnement du réseau . . . 136
6. Agents d'entrée et sortie . . . 137
7. Syntaxe des périphériques d'entrée . . . 137
8. Agent input pour la souris . . . 138
9. Syntaxe des périphériques de sortie . . . 139
10. Agent output pour le PHANToM . . . 139
11. Messages pour la sortie sur le PHANToM . . . 140
12. Messages pour la création de PICOB . . . 141
13. Message pour le point de guidage . . . 141
14. Messages pour la gestion de la liste de points de guidage . . . 142
15. Messages des entrées de la VTPlayer . . . 143
16. Messages des sorties de la VTPlayer . . . 143
17. Messages des entrées clavier et souris . . . 144
18. Messages pour la lecture de sons . . . 144
19. Messages pour la synthèse vocale . . . 145
20. Exemple de schéma de circuit à représenter . . . 146
21. Source d'un circuit et sa représentation visuelle . . . 147
22. Représentation visuelle des fils et des nœuds . . . 148
23. Représentation visuelle des composants . . . 148
24. Aimantation du schéma . . . 149
25. Représentation haptique des composants . . . 149
26. Aimantation du circuit avec les composants . . . 150
27. Indications directionnelles avec des PICOB guidés . . . 151

28. Temps d'exploration par circuit 155
29. Problème de téléportation dans un circuit 156
30. Exploration typique du circuit 9 157
31. Messages pour les curseurs de l'application de circuits électriques 159
32. Messages pour chaque objet l'application de circuits électriques 160
33. Représentations haptiques des objets 160
34. Ajout de positions intermédiaires 162
35. Formes de circuits de l'application finale 166
36. Circuits utilisés dans les tests finaux 166
37. Un utilisateur et un enseignant 167

Liste des tableaux

CHAPITRE I L'interaction et la déficience visuelle 1

1. Propriétés des mécanorécepteurs de la peau 14
2. Spécifications des périphériques PHANToM [157] 23
3. Résumé des spécifications des périphériques de pointages à retour de force 26

CHAPITRE II Icônes tactiles 51

1. Correspondance entre les durées des images et les figures de notes 90

CHAPITRE III Icônes kinesthésiques 97

1. Expérience 2 : moyenne des matrices de confusion 105
2. Utilisateurs 118
3. Séquences à reconnaître et les propositions 119
4. Réponses des utilisateurs dans la série 1 120
5. Réponses des utilisateurs dans la série 2 121

CHAPITRE IV Applications multimodales 127

1. Schémas utilisés dans l'expérience 152
2. Utilisateurs 153
3. Les circuits à reconnaître et les propositions 153
4. Réponses des utilisateurs 154
5. Résultats pour chaque circuit 155
6. Techniques d'interaction 163
7. Utilisateurs de l'application finale 165
8. Circuits explorés par les utilisateurs 168
9. Erreurs de reconnaissance de la forme des circuits 170
10. Erreurs de reconnaissance de composants 170
11. Confusions entre composants 171

Remerciements

Mon travail de thèse a été réalisé au sein tdu thème Interaction Homme Machine de l'équipe Calculs, Modèles et Interfaces du Laboratoire d'Informatique Théorique et Appliquée de l'Université Paul Verlaine – Metz. Je tiens à remercier en premier lieu Maurice Margenstern, mon directeur de thèse et directeur du LITA pendant la majeure partie de ma thèse qui a montré un intérêt constant pour mon travail. Je remercie aussi très chaleureusement mes deux encadrants : Benoît Martin et Isabelle Pecci. Grâce à eux j'ai pu travailler sur un sujet passionnant, et j'ai pu participer au projet MICOLE dans lequel j'ai pu travailler avec des experts de mon domaine. Grâce à eux j'ai aussi pu publier mes travaux à plusieurs reprises, et les exposer dans diverses conférences sur trois continents. Ils m'ont appris le métier, et je leur en suis profondément reconnaissant.

Je remercie vivement mes deux rapporteurs Yacine Bellik et Vincent Hayward, ainsi que mes examinateurs Bertrand David et Yves Guiard pour avoir accepté d'évaluer mon travail. C'est un immense honneur pour moi d'avoir pu leur exposer mes travaux, et j'espère en avoir été digne.

Je remercie aussi mes collègues de travail, que ce soit Kamel Chelghoum, Nicolas Noble et Jérôme Wax du thème IHM, ou les divers collaborateurs avec qui j'ai pu travailler directement, à savoir Andrew Crossan et Stephen Brewster de l'Université de Glasgow, et Roope Raisamo, Janne Jarvi et Rami Saarinen de l'Université de Tampere. Je salue l'ensemble des partenaires du projet MICOLE avec qui j'ai eu de passionnantes discussions lors des diverses réunions auquelles j'ai eu la chance d'assister.

Je suis partagé entre les remerciements et les excuses envers ceux qui ont eu la tâche difficile de partager le même bureau que moi au quotidien, à savoir Nicolas et Jérôme déjà cités, mais aussi Serge Gaspers, Thomas Veneziano, Julien Schleich, et plus récemment Vincent Demange. Je n'oublie évidemment pas les autres doctorants avec qui j'ai passé des heures sur des débats en tout genres, que ce soit au restaurant universitaire ou pendant la pause café. Merci à vous Mathieu Liedloff, Michaël Rao et David Michel.

Je remercie ensuite l'ensemble du personnel de l'Université. D'une part je suis reconnaissant envers les enseignants à qui je dois mes connaissances en informatique, et avec qui j'ai pu ensuite collaborer en tant qu'enseignant. D'autre part je souligne le travail primordial du personnel non enseignant, et en particulier Martine Salm, Catherine Noël, Damien Aignel et Sarah Fellini, sans qui

rien de tout cela ne serait possible.

Je remercie sincèrement l'ensemble des enseignants qui m'ont tant appris tout au long de ma scolarité. J'ai appris de nombreuses choses passionnantes grâce à eux, et je ne les en remercierai jamais assez.

J'exprime toute ma gratitude envers mes amis, qui m'ont toujours soutenu, aidé et distrait quand il le fallait. Je me sens privilégié pour avoir autant d'amis sur lesquels je peux compter. Par ordre alphabétique, merci à Nicolas Bertrand, Anne Buchou, Émilie Chosseler, Stéphane et Sandrine Enginger, Guillaume François, Adrien et Aurélie Lazarus, Mélanie Martin, Sylvain Michot, Éric Nicolas, Vincent, Mylène et Jean-François Noal, Michaël Oliveira, Rossella Pignanelli, Valérie Radek, Romain Stankiewicz, Didier Ziegler. Que ceux que j'ai oubliés me pardonnent.

Je vais conclure par ceux qui me sont le plus cher, à savoir ma famille. En particulier je ne trouverai jamais de mots assez forts pour décrire la gratitude que j'ai envers mes parents et mon frère. Tout ce que je suis et ce que j'ai fait je le leur dois. Je leur dois tout, et rien au monde ne peut rembourser la dette que j'ai auprès d'eux. C'est avec beaucoup d'émotions que je tiens à les remercier tout particulièrement, car c'est grâce à eux que j'ai pu réaliser mon rêve.

Pour finir je vais rendre hommage à mon grand père. Il était passionné de science et aurait été très fier d'assister à l'aboutissement de mes études. C'est avec regret que j'ai fini quelques mois trop tard pour qu'il puisse assister à cet aboutissement qui l'aurait beaucoup ému. C'est avec tout l'amour que j'ai pour lui que je lui dédie tout ce travail.

À mon grand-père,

Introduction

Étudie, non pour savoir plus, mais pour savoir mieux.

Sénèque

La navigation d'interfaces est généralement une tâche essentiellement visuelle. En effet les informations, qu'elles soient importantes ou non, sont le plus souvent affichées sous forme de textes ou de graphiques. La modalité auditive est utilisée occasionnellement pour des notifications. L'intérêt des interactions non-visuelles a plusieurs motivations. Premièrement le canal visuel peut être surchargé, et donc l'ajout d'informations supplémentaires de manière visuelle ne fait que pénaliser la compréhension des informations déjà existantes. Dans ce cas il faut tenter d'utiliser une autre modalité afin de soulager le canal visuel qui est déjà surchargé. Deuxièmement il est possible que la surface d'affichage soit limitée, et donc que la quantité d'information pouvant être affichée soit limitée. Dans ce cas il faut donc bien trouver une solution pour transmettre plus d'informations à l'utilisateur. L'utilisation d'une autre modalité devient nécessaire. Il est parfois conseillé de suppléer l'affichage visuel lorsque la vision est déjà utilisée par une autre tâche. L'exemple du quotidien est la conduite à l'aide d'un système GPS. Le conducteur doit utiliser sa vision pour regarder la route. Les systèmes de GPS possèdent donc une synthèse vocale pour indiquer le chemin à suivre. C'est un bon exemple où la vision est utilisée en seconde modalité : certains systèmes GPS possèdent un écran pour afficher la carte routière avec le chemin à suivre au cas où les informations auditives ne sont pas suffisantes. Troisièmement il est nécessaire de trouver une autre modalité d'affichage lorsque le rendu visuel n'est pas possible pour l'utilisateur. C'est le cas pour l'utilisation d'un système dans l'obscurité, ou pour un utilisateur possédant une déficience visuelle.

Dans plusieurs des cas énoncés ci-dessus, le rendu auditif peut être utilisé. Quand les informations sont basées sur le langage naturel, il est possible d'utiliser une synthèse vocale. S'il s'agit de représenter des objets ou des actions, le système peut soit utiliser une synthèse vocale pour prononcer le nom de l'objet ou de l'action, soit le représenter par une icône auditive. Cependant il y a des situations où le retour sonore est soit inutilisable soit déconseillé. Il peut s'agir d'environnements bruités, ou au contraire d'environnements où le silence est requis. Le travail exposé dans cette thèse vise à concevoir et utiliser des techniques d'interaction mettant à contribution le sens du toucher afin d'afficher des informations de manière non-visuelle dans de telles situations. Le sens du toucher est stimulé d'une part par le retour tactile et d'autre part par le retour de force. On parle alors de retour haptique, et par extension d'interactions haptiques. Bien que toutes les situations évoquées

ci-dessus soient intéressantes à étudier, nous nous concentrerons sur l'utilisation de l'haptique dans des systèmes destinés à des utilisateurs déficients visuels. En effet le travail exposé dans les chapitres suivants a été réalisé au sein du projet européen MICOLE [113], qui avait pour but d'étudier les environnements multimodaux et collaboratifs pour l'intégration d'enfants déficients visuels. Nous utiliserons donc les techniques d'interaction que nous allons développer dans des applications destinées à des utilisateurs déficients visuels.

Contexte

Les études préliminaires menées au stade initial du projet européen ont permis d'identifier les besoins des enfants déficients visuels en terme d'outils pédagogiques. Parmi les personnes déficientes visuelles, nous distinguons celles possédant une vision résiduelle et les non voyants. Les domaines les plus demandeurs d'outils sont les mathématiques, la géographie, et de manière générale toutes les matières utilisant des schémas.

Le problème des mathématiques est que la technique utilisée pour représenter les expressions mathématiques est un braille étendu. Le premier problème est qu'il existe plusieurs normes pour représenter les symboles mathématiques. Celles-ci ont été définies par les constructeurs de matériel. Le second problème est que le braille n'est utilisable que de manière linéaire. Or nous utilisons parfois deux dimensions pour représenter des objets mathématiques. Les fractions par exemple utilisent une seconde dimension. Même s'il est possible d'écrire les fractions de manière linéaire, il n'est pas évident de les manipuler de la sorte si elles sont de grande taille. Le problème de la géographie et des schémas est que ce sont des représentations fortement visuelles. En classe, les schémas sont dessinés sur du papier spécial où les traits sont mis en relief grâce à un stylo thermique ou un poinçon. Les enfants explorent ces schémas avec leurs 10 doigts. Cependant certaines techniques telles que la perspective sont exclusivement visuelles. Les schémas utilisant la perspective sont donc incompréhensibles par une personne non voyante. Les enfants possédant une vision résiduelle peuvent utiliser des schémas agrandis avec un fort contraste. Cependant tout comme pour les schémas tactiles, il est nécessaire d'effectuer des simplifications. Ces techniques possèdent plusieurs inconvénients.

Le premier est qu'elles ne sont utilisées ou utilisables que par des enfants déficients visuels. Or ceci est pour ces enfants une source d'exclusion. Ces enfants sont souvent intégrés dans des classes avec des enfants ne possédant pas de déficience visuelle, et souhaitent utiliser les mêmes outils que leurs camarades. La création d'un système multimodal serait une solution car il permettrait d'afficher les informations nécessaires à la fois de manière visuelle et de manière haptique.

Le second inconvénient est la nature statique des schémas en relief. En effet, souvent l'enseignant est obligé d'aider l'enfant en le prenant par la main. Ceci est parfois aussi gênant pour l'enseignant que pour l'élève. Il en résulte que certains enseignants s'abstiennent donc d'aider les élèves. L'utilisa-

tion d'un guidage permettrait d'aider l'élève à explorer le schéma sans avoir besoin d'être en contact direct avec lui.

Objectifs

Le travail exposé dans cette thèse a pour but de créer des techniques d'interaction utilisant le retour tactile et le retour de force pour coder des informations, puis de les intégrer dans une application multimodale. Nous avons divisé ce but en trois objectifs, qui sont traités dans les trois derniers chapitres de cette thèse.

Le premier objectif consiste à créer des icônes tactiles utilisant une matrice de picots. Le retour tactile a déjà été utilisé pour créer des icônes vibrotactiles. Une de ces techniques, appelée Tactons, consiste à créer des vibrations selon plusieurs paramètres définis [26, 25, 30, 31, 32, 29]. Les différents paramètres sont structurés de manière à permettre de transmettre plusieurs informations à l'utilisateur à l'aide d'un seul Tacton. Nous nous inspirerons de ces travaux pour concevoir des Tactons utilisant des matrices de picots à la place de vibreurs. Ainsi nous espérons créer des icônes tactiles utilisant plusieurs paramètres. Certaines études font état de telles icônes tactiles, sans fournir toutefois de structure analogue à celle des Tactons [152, 19, 90, 112, 118]. La méthode présentée par ces études consiste à créer des icônes de manière empirique et de leur attribuer un sens via des suggestions obtenues suite à des tests utilisateur. Notre approche sera opposée : nous fixerons la signification de nos icônes, et nous vérifierons par des tests que les utilisateurs parviennent à interpréter le sens correctement.

Le deuxième objectif était de créer des icônes utilisant le retour de force. Nous utiliserons la même méthode que précédemment pour créer ces icônes de manière structurée. Des icônes basiques ont été développées par Maclean et Enriquez [53, 102, 54]. Ces icônes utilisent un périphérique ne possédant qu'un degré de liberté en rotation. Les possibilités offertes par cette technique sont donc limitées, mais elles montrent que le retour de force peut être aussi utilisé pour coder des informations. Nous allons donc créer des icônes utilisant les trois dimensions de l'espace, tout en leur donnant une structure similaire à celle des Tactons. Ces deux techniques d'interaction que nous allons étudier dans les chapitres suivant ont été conçues dans la phase d'études fondamentales du projet européen MICOLE.

Le troisième objectif de cette thèse vise à utiliser les résultats des études fondamentales obtenus lors des expérimentations précédentes. Pour atteindre cet objectif, nous allons dans un premier temps concevoir une architecture multimodale permettant de faciliter le développement d'applications multimodales et collaboratives. Cette architecture doit inclure, entre autres, les techniques d'interaction décrites ci-dessus, ainsi que d'autres techniques développées par les autres partenaires du projet européen MICOLE. Le partenaire finlandais avait déjà développé une architecture multimodale [153]. Cependant la structure de cette architecture était très rigide, et avait été conçue pour les besoins d'une application d'exploration du système solaire. Le système de communication entre agents est basique, et l'aspect collaboratif n'est pas pris en compte. L'architecture développée dans

les chapitres suivants remédie à ces problèmes. En second lieu nous allons utiliser cette architecture pour créer une application utilisant les techniques d'interaction ci-dessus pour afficher des informations de manière non-visuelle, et vérifier par des tests utilisateur que ces techniques s'intègrent avec succès dans une application concrète.

Structure de la thèse

Dans cette thèse nous introduisons les concepts de multimodalité et d'haptique. Nous définissons ensuite deux techniques d'interaction haptique permettant de coder des informations. Enfin nous concevons une architecture multimodale et une application multimodale basée sur cette architecture. Le document est composé de 4 chapitre dont voici un aperçu :

Chapitre I : l'interaction et la déficience visuelle

Dans ce premier chapitre nous allons tout d'abord introduire la notion d'interface, à travers les principaux modèles d'interfaces. Cette partie a pour but de décrire quelques manières de structurer une interface, et d'en décrire les principaux composants. Ceci fait nous aborderons la notion de multimodalité, qui est centrale dans l'ensemble des travaux de ce document. Le concept de multimodalité est défini et expliqué à travers les principaux espaces de conception. Nous entrerons ensuite dans les détails du sujet qui nous intéressent dans cette thèse : les interactions haptiques en sortie. Nous allons donc aborder dans un premier temps l'haptique au niveau sensoriel, c'est-à-dire du côté humain. Après cela nous évoquerons quelques technologies permettant de stimuler le sens du toucher. Nous insisterons sur les périphériques similaires à ceux que nous allons utiliser par la suite. Enfin nous nous intéresserons à des systèmes existants destinés aux utilisateurs déficients visuels. En préambule nous examinerons des études sur l'exploration non-visuelle, avant de se pencher sur des études sur des systèmes dédiés aux utilisateurs déficients visuels en particulier.

Chapitre II : Icônes tactiles

Le deuxième chapitre traite de la conception d'une technique d'interaction permettant d'afficher des informations sous forme d'icônes tactiles, appelées Tactons. Nous commencerons cette étude par un rappel sur le concept de Tacton, puis nous étendrons cette notion avec nos propres dimensions au cas des matrices de picots. Par la suite nous détaillerons la conception et l'évaluation de Tactons statiques et dynamiques mono-dimensionnels au travers de 5 expériences. La conception de ces Tactons se fait incrémentalement au cours de ces expériences. Le but est de trouver de bons ensembles de Tactons pour représenter des directions, et ce en utilisant trois méthodes. La première méthode consiste à utiliser des Tactons statiques, et les deux suivantes consiste à utiliser des Tactons dynamiques : des vagues et des clignotements. Ensuite nous utiliserons les résultats de ces expériences afin de créer des Tactons multi-dimensionnels. L'expérience décrite dans cette partie vise à déterminer la quantité d'informations que nous pouvons transmettre à l'utilisateur grâce à ces Tactons. Nous présenterons ensuite une méthode permettant d'explorer des formes géométriques grâce à un guidage utilisant les Tactons directionnels précédemment expérimentés. Nous exposerons une étude comparant cette méthode à une méthode de présentation traditionnelle se contentant de transcrire les pixels en picots

levés. Enfin nous évoquerons une analogie entre nos définitions de Tactons avec la notation musicale. Cette analogie laisse une question ouverte intéressante sur la correspondance entre ces Tactons et des icônes auditives telles que les earcons [22, 28].

Chapitre III : Icônes kinesthésiques

Ce troisième chapitre a pour objectif de présenter la conception et l'évaluation d'une autre technique d'interaction appelée PICOB, utilisant cette fois le retour de force pour présenter des informations grâce à des icônes kinesthésiques. Les PICOB sont des séquences d'impulsions senties par l'utilisateur grâce à un bras articulé à retour de force. Nous commençons par présenter le concept d'impulsions par ses paramètres et ses modes de lecture. Dans le mode de lecture guidé, le système contrôle totalement le déplacement de l'impulsion alors que dans le mode semi-guidé, l'utilisateur est partiellement contraint, et utilise la dimension laissée libre afin de ressentir l'impulsion. Nous présentons ensuite une série d'expériences visant à tester la reconnaissance d'impulsions guidées. Après cela nous détaillons une deuxième série d'expériences, dédiées à l'évaluation d'impulsions semi-guidées. Enfin la dernière section décrit l'évaluation de séquences d'impulsions, en vue de leur utilisation dans une application détaillée dans le chapitre suivant.

Chapitre IV : Applications multimodales

Dans ce dernier chapitre nous commençons par décrire l'architecture multimodale que nous avons conçue en collaboration avec l'université de Tampere en Finlande dans le cadre du projet européen MICOLE. Dans un premier temps nous décrivons la structure d'agents et le bus logiciel que ces derniers utilisent afin de communiquer. Nous détaillons ensuite le framework multimodal que nous avons développé. Ce framework définit une structure pour l'ajout de périphériques d'entrée et de sortie. Nous y décrivons une librairie d'agents incluse dans cette architecture, qui permet d'utiliser aisément les techniques d'interaction décrites dans les chapitres précédents. Après la description de cette architecture, nous discutons du développement incrémental d'une application d'exploration de schémas de circuits électriques. Cette application utilise un bras articulé à retour de force pour explorer les schémas, ainsi qu'un périphérique tactile en sortie. Nous aborderons tout d'abord la conception et l'évaluation informelle du premier prototype. Nous étudierons ensuite le second prototype basé sur le précédent et les améliorations suggérées par les tests informels de ce dernier. Enfin nous nous intéresserons au portage de cette application sur l'architecture, en mettant en évidence les nouvelles techniques d'interaction. Pour finir, ce chapitre présente une évaluation de cette application avec des utilisateurs déficients visuels.

Chapitre I

L'interaction et la déficience visuelle

La bibliographie se fait après et non
avant d'aborder un sujet de recherche.
Jean Perrin

Résumé

Ce chapitre constitue un état de l'art sur les interactions, les modalités et les applications que nous allons détailler dans les chapitres suivants. Nous commençons par étudier les interfaces au sens large, grâce aux principaux modèles d'interface. Ensuite nous nous intéressons à la notion de multimodalité à travers quelques espaces de conception et de caractérisation de modalités. Nous expliquons ensuite la notion d'haptique, qui est la modalité étudiée dans les chapitres suivants. Nous l'étudions d'abord du point de vue humain grâce à la somesthésie. Puis nous l'abordons du point de vue machine avec tout d'abord un éventail de périphériques haptiques, et ensuite quelques interactions utilisant cette modalité. Enfin nous nous intéressons aux systèmes multimodaux pour utilisateurs déficients visuels, et en particuliers ceux utilisant l'haptique. Pour cela nous évoquons des méthodes de présentation et d'exploration d'interface. Nous continuons par des études sur l'exploration non visuelle, et nous terminons par des système utilisant l'haptique destinés à l'enseignement.

1. Introduction

Les systèmes informatiques sont en constante évolution depuis les dernières décennies. Tant que les interfaces étaient textuelles, les déficients visuels pouvaient les explorer entièrement et efficacement à l'aide de tablettes tactiles et de synthèses vocales. L'arrivée des interfaces graphiques a compliqué l'exploration des interfaces pour ces utilisateurs. Le développement des interfaces multimodales permet de communiquer de plusieurs façons avec un système informatique. Ainsi il est possible d'utiliser des canaux sous-utilisés tels que l'haptique afin de transmettre de l'information aux utilisateurs déficients visuels. L'affichage d'informations en utilisant l'haptique peut se faire en utilisant le retour tactile ou le retour de force. Dans ce chapitre nous allons commencer par décrire quelques modèles d'interfaces afin de souligner les principaux problèmes liés à leur conception. Nous enchaînerons sur la notion de multimodalité afin de situer les travaux des chapitres suivants au sein d'une interface de manière générale. Ensuite nous étudierons l'haptique, qui est la modalité utilisée

dans nos travaux. Enfin nous aborderons les systèmes pour utilisateurs déficients visuels, qui est le domaine d'application que nous avons ciblé tout au long de nos travaux.

2. Les interfaces

Les premiers systèmes informatisés laissaient peu de place à l'interaction : les données étaient engluées dans le programme, et le rendu du résultat était sommaire. L'évolution des systèmes informatiques a rendu nécessaire l'exploration de nouveaux moyens d'acquérir et d'afficher les données. En effet les systèmes permettent de traiter des quantités de données de plus en plus importantes. De plus la variété des informations augmente aussi : texte, son, parole, images, vidéos, etc. Il est devenu primordial de formaliser la manière de concevoir, d'implémenter et d'évaluer les interfaces afin de les rendre plus accessibles et plus efficaces.

2.1. Modèles d'architectures logicielles des systèmes interactifs

Un gros effort a été effectué par la communauté internationale d'interaction homme machine afin de développer des modèles d'interfaces [39]. En effet on peut décomposer tout système interactif en deux grandes parties : le *noyau fonctionnel* qui contient la partie fonctionnelle du programme et l'interface qui permet à l'utilisateur d'interagir avec ce noyau. Ces modèles ont pour but de clarifier les interactions et plus précisément la partie interface des systèmes afin qu'elle relie efficacement l'utilisateur au noyau fonctionnel. On distingue deux types de modèles d'interface. D'une part les modèles à couches effectuent les transitions entre l'utilisateur et le système par analyse lexicale, syntaxique et sémantique. D'autre part les modèles à agents se basent sur le comportement des langages de programmation objet. Le sujet de cette thèse n'étant pas centré sur les modèles d'interface, nous n'allons exposer que les principaux modèles de chaque type. Ainsi nous aurons une vue d'ensemble sur le sujet, sans toutefois entrer dans les détails qui sortent du cadre de l'étude présente.

2.1.1. Modèles à couches

Le modèle de Seeheim (figure 1) est le premier modèle d'interface à couches qui s'est répandu. Il a été conçu lors d'un workshop organisé à Seeheim en 1983 sur les systèmes de gestion d'interface utilisateur (User Interface Management Systems ou UIMS en anglais) [135, 66]. L'idée de ce modèle est de diviser l'interface en trois couches. La première couche est la présentation : elle gère les entrées et les sorties. C'est donc la partie directement en contact avec l'utilisateur à travers les périphériques. D'un point de vue linguistique cette coouche s'occupe de la partie lexicale. La deuxième couche contient le contrôleur de dialogue. Elle a pour tâche de gérer le séquencement des entrées et des sorties. Ce modèle étant conçu à l'époque des interfaces textuelles, le contrôleur de dialogue doit gérer la successions des écrans à afficher. D'un point de vue linguistique, il travaille sur la partie syntaxique de l'interface. Enfin la troisième couche est l'interface du noyau fonctionnel. C'est cette couche qui communique avec le noyau fonctionnel du programme. Elle permet d'afficher les données du noyau sous forme compréhensible par l'utilisateur, et permet d'interpréter les actions de l'utilisateur afin de les répercuter sur le noyau. Cette couche gère la partie sémantique de l'interface.

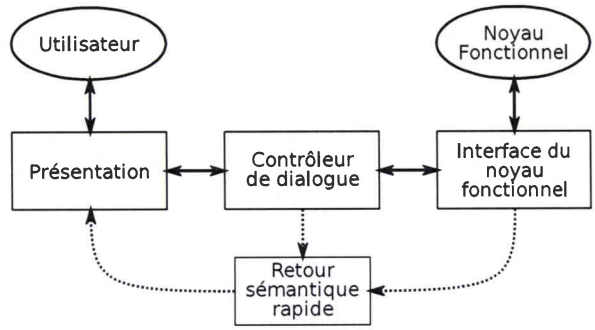


Figure 1 – Modèle de Seeheim

Le modèle de Seeheim est devenu une référence à la fois pour la création d'interfaces (surtout textuelles) et pour la création d'autres modèles d'interfaces. Le modèle Arch (figure 2) est une évolution de Seeheim [4]. Il est composé de 5 parties, issues des composants du modèle de Seeheim. La partie présentation du modèle de Seeheim a été divisée en deux parties : l'interaction et la présentation. La première contient les widgets et les communications avec les périphériques. Elle est dépendante de la plate-forme. La deuxième assure le lien entre l'interaction et le contrôleur de dialogue. Elle est indépendante de la plate-forme. Ainsi cette modification permet de créer des systèmes plus facilement portables car les parties dépendantes de la plate-forme sont bien identifiées. Le contrôleur de dialogue est identique à celui du modèle de Seeheim : il gère l'ordonnancement des interactions. L'adaptateur du domaine a un rôle similaire à celui de l'interface du noyau fonctionnel du modèle de Seeheim. Il réalise les tâches qui dépendent du noyau fonctionnel, mais qui ont lieu en dehors du noyau fonctionnel. Le modèle Arch précise aussi le type d'objets échangés entre les différents composants. Les objets du domaine représentent les données venant du noyau fonctionnel. Les objets de présentation contiennent les informations provenant des événements produits par l'utilisateur. Enfin les objets d'interaction sont les widgets eux même et leurs propriétés. Les objets d'interaction dépendent d'une boîte à outils spécifique alors que les objets de présentation sont indépendants.

2.1.2. Modèles à agents

Les agents sont issus de l'Intelligence Artificielle. Nigay distingue deux types d'agents. D'une part il y a les *agents intelligents*, qui ont un comportement adaptatif. D'autre part les agents il y a les *agents réactifs*, dont le comportement est entièrement programmé et n'évolue pas [120]. Sauf mention contraire, en Interaction Homme-Machine nous utilisons des agents réactifs.

Dès les premières interfaces graphiques et l'invention des langages objet au Xerox Parc, des modèles d'interface à agents ont été créés. Reenskaug [149] définit le modèle *Thing-Model-View-Editor* (chose-modèle-vue-éditeur). Une *chose* représente le système à implémenter. Il s'agit du problème auquel s'intéresse l'utilisateur, et que le système est censé résoudre. Un *modèle* constitue une abstraction interne du problème. Il s'agit d'une collection de données et d'opérations sur ces données qui correspondent à la réalisation de la tâche. Un modèle peut être composé de sous-modèles, qui représentent un raffinement du problème général. Une *vue* est une représentation

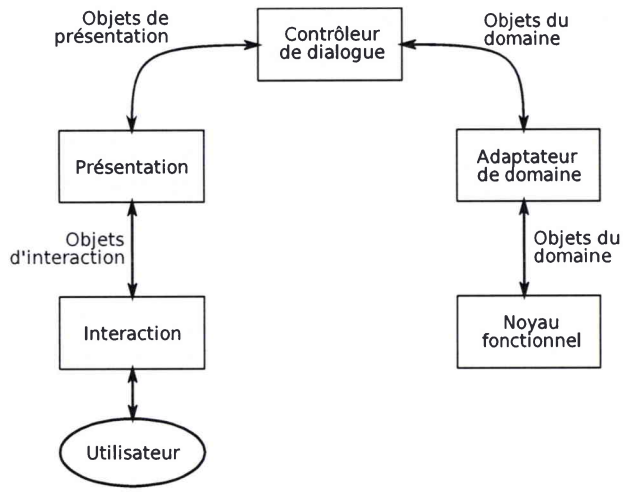


Figure 2 – Modèle Arch

partielle ou totale d'un modèle. Elle peut mettre en avant ou masquer certaines parties du modèle, selon le rôle qui lui est défini. À cette époque le concept de vue est encore limité car Reenskaug ne considère que les sorties écran et imprimante. Notons que la vue s'occupe aussi de certaines tâches agissant sur le modèle, mais reliées à la vue. Enfin l'*éditeur* permet à l'utilisateur d'agir sur une ou plusieurs vues. Ceci comprend des commandes, des menus, etc.

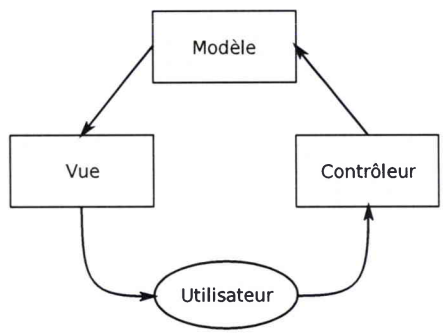


Figure 3 – Modèle MVC

Cette architecture a été remaniée pour donner le modèle *Model-View-Controler* (MVC, figure 3) [148], initialement utilisé par le langage Smalltalk, et encore utilisé de nos jours pour des applications écrites dans des langages objet comme C++ ou JAVA. On remarque tout d'abord que la notion de chose n'apparaît plus dans le modèle. Ensuite on note que la structure est similaire à celle de l'architecture précédente. Les vues conservent leur rôle de représentation du modèle. Il est toutefois précisé que la vue et le modèle lui correspondant partagent un langage commun afin que la vue puisse maintenir une consistance avec le modèle. Les contrôleurs sont une généralisation de la notion d'éditeur. Ils permettent à l'utilisateur d'arranger les vues comme il le souhaite, et fournissent

aussi des moyens d'agir sur les données comme des menus. Les contrôleurs sont responsables de la capture des informations venant de l'utilisateur, et de transmettre les informations requises aux vues concernées. Ainsi les vues ne reçoivent jamais directement les informations des périphériques tels que le clavier ou la souris, mais à travers les contrôleurs. Enfin la notion d'éditeur persiste, mais en tant que contrôleur particulier. En effet certaines vues possèdent un contrôleur de type éditeur, qui a pour rôle de modifier les données de la vue.

Le modèle *PAC* (Présentation-Abstraction-Contrôleur) de Coutaz [38] utilise une autre approche (figure 4). D'un côté il y a l'*abstraction* qui correspond au modèle dans MVC, c'est-à-dire la formalisation sur la machine du problème réel. D'un autre côté il y a la *présentation* qui correspond au dialogue entre l'utilisateur et le système. Ce composant mêle aussi bien les entrées que les sorties. Il correspond à la fois à la vue et au contrôleur de MVC. Enfin le *contrôleur* a cette fois-ci pour rôle de maintenir la consistance entre l'abstraction et la présentation. Ainsi si l'abstraction est modifiée en interne par le cœur du système, le contrôleur doit veiller à ce que les affichages correspondent bien à l'abstraction mise à jour. À l'inverse si l'utilisateur agit sur l'interface le contrôleur doit s'assurer que les opérations appropriées sur l'abstraction sont effectuées. Les applications implémentées sur ce modèle sont composées d'agents, appelés *agents PAC*, qui possèdent les trois facettes : abstraction, contrôle et présentation. La communication entre agents s'effectue par le biais des contrôleurs de chaque agent selon une structure hiérarchique (figure 4 à droite).

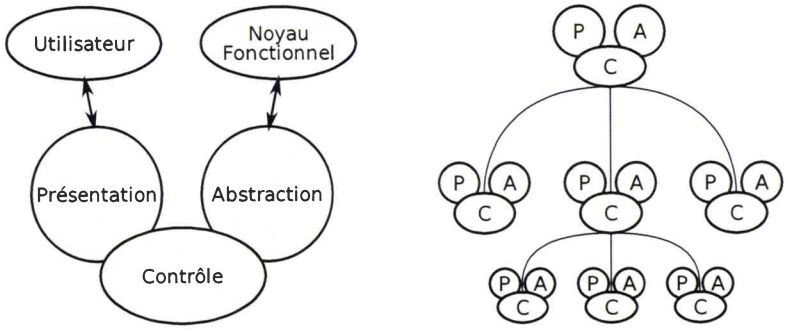


Figure 4 – Modèle PAC et hiérarchie PAC

Cette décomposition d'un programme permet de bien séparer les données et leur traitement de l'interface. Cependant des problèmes peuvent surgir en tentant de rendre interactif un système qui ne l'est pas [50]. En effet une application de simulation ayant déjà ses propres agents communiquants, mais non interactifs se retrouveraient dans un système PAC sous la forme d'une collection d'abstractions. Dans ce cas la communication ne se ferait plus à travers les contrôleurs, mais par les abstractions. *PAC-Amodeus* [41], une extension de PAC permet de contourner ce problème (figure 5). Il s'agit d'un modèle hybride qui combine le modèle Arch et le modèle PAC. Les agents PAC sont placés dans le contrôleur de dialogue. Le composant de présentation dialogue avec les facettes présentation des agents PAC, et l'adaptateur de domaine dialogue avec les facettes abstraction des agents PAC.

Ce modèle hybride permet de combiner les qualités de portabilité du modèle Arch à la finesse de description des interactions offerte par le modèle PAC.

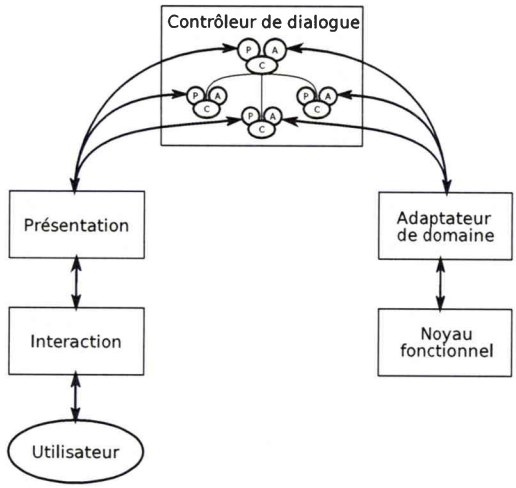


Figure 5 – Modèle PAC-Amodeus

2.1.3. Conclusion

Les modèles d'interface présentés permettent de décrire la structure de l'interface d'un système. Les deux types de modèles,à couches et à agents, ont leurs avantages. Les modèles à couches comme Seeheim ou Arch permettent de détailler l'évolution de l'information transitant entre l'utilisateur et le noyau fonctionnel du programme. Les modèles à agents tels que MVC et PAC visent plutôt à décrire chaque interaction en tant que tâche. Le cas particulier de PAC-Amodeus combine les deux approches afin d'en tirer les avantages des uns et des autres. Le travail exposé dans les chapitres suivants consiste essentiellement en des techniques d'interaction en sortie, utilisant le retour haptique. D'un point de vue des modèles de Seeheim et Arch nous nous placerons majoritairement dans les composants de présentation et d'interaction. D'un point de vue des modèles MVC et PAC nous nous concentrerons sur les vues et la présentation.

2.2. Multimodalité

Les systèmes multimodaux se caractérisent par les techniques d'interaction qu'ils mettent en œuvre. Nigay et Coutaz [123] définissent une technique d'interaction comme étant un couple $\langle p, L \rangle$ formé d'un périphérique p et d'un langage d'interaction L. Nigay [121] définit un périphérique comme étant un artefact du système qui acquiert (périphériques d'entrée) ou retourne (périphériques de sortie) des informations. La souris, le clavier ou le microphone sont des exemples de périphériques d'entrée. L'écran, la tablette braille ou le haut parleur sont des périphériques de sortie. Un langage d'interaction est défini par Nigay comme un langage utilisé par le système ou l'utilisateur pour échanger de l'information. Un langage est un ensemble d'expressions respectant des règles, dont les lexèmes sont des éléments de base résultant de l'interaction avec un périphérique. Un texte ou

un son sont des lexèmes générés par des périphériques. Le déplacement de la souris ou un mot prononcé sont des lexèmes générés par l'utilisateur. Le langage naturel et la manipulation directe sont des exemples de langages d'interaction.

Un système multimodal utilise plusieurs techniques d'interaction. Nous avons vu en quoi se caractérise une technique d'interaction, il nous reste à étudier comment elles se combinent. Bolt fut l'un des premiers à réfléchir sur la manière de combiner les techniques d'interaction à travers le concept de « *Put-That-There* » (« met ça là ») [23]. Le système qu'il a défini utilise les commandes vocales, le suivi du regard et les gestes. L'exemple classique consiste à déplacer un objet à l'aide d'une commande vocale, tout en désignant l'objet en question d'un geste. Cette étude expose d'autres opérations comme la création d'objets. Les propriétés non saisies sont positionnées à une valeur par défaut. Il est possible de déplacer un objet oralement vers une position relative à un autre objet : « déplace ça à droite du carré vert ». Le système prévoit aussi une modification des objets, que ce soit en donnant directement la propriété et la valeur souhaitée, ou en précisant de copier une valeur d'un autre objet. Enfin on peut aussi supprimer et nommer des objets. Toutes ces opérations se font en combinant le geste et la parole.

Le choix de la manière d'utiliser des techniques d'interaction dans un système est une tâche complexe. Des recherches ont donc été menées sur ce sujet. Elles ont abouti d'une part sur les espaces de conception d'interfaces multimodales et d'autre part sur les modèles de caractérisation de la coopération entre modalités. Nous allons en étudier quelques uns afin d'afficher les diverses problématiques liées à la création d'interfaces multimodales. Cependant cette thèse n'ayant pas pour sujet les espaces de conception et de caractérisation d'interactions multimodales, nous n'aborderons que les plus importants, sans entrer dans les détails techniques non pertinents dans notre contexte. De plus, étant donné que dans les chapitres suivants nous nous intéresserons quasi-exclusivement aux modalités en sortie, nous n'évoquerons que des espaces permettant de caractériser des techniques d'interaction en sortie.

2.2.1. Les espaces de conception

L'espace de Frohlich.

Frohlich fut le premier à créer un espace qui évoque simultanément les techniques d'interaction en entrée (figure 6) et les techniques d'interaction en sortie (figure 7) [3]. Ce modèle est centré utilisateur, et vise à utiliser les interactions du monde réel dans un environnement virtuel. Il décrit deux structures distinctes pour les interactions d'entrée et pour les interactions de sortie. Ces structures reposent sur cinq notions. Ici le *mode* représente la manière dont l'utilisateur agit sur le système : par le langage ou par une action. Les deux autres notions sont les *canaux de communication*, qui forment les interfaces de l'utilisateur et de la machine. En entrée les *canaux de communication utilisateur* sont la voix et le mouvement, et correspondent aux *canaux de communication système* qui sont l'audio, le visuel et l'haptique. En sortie, les canaux de communication utilisateur sont l'ouïe, la vue et le toucher, et correspondent aux mêmes canaux de communication système qu'en entrée. L'odorat et le goût ne sont pas considérés dans ce modèle. Ensuite le *média* correspond à la manière de représenter les données. Les médias correspondant au mode de langage sont la parole, le texte

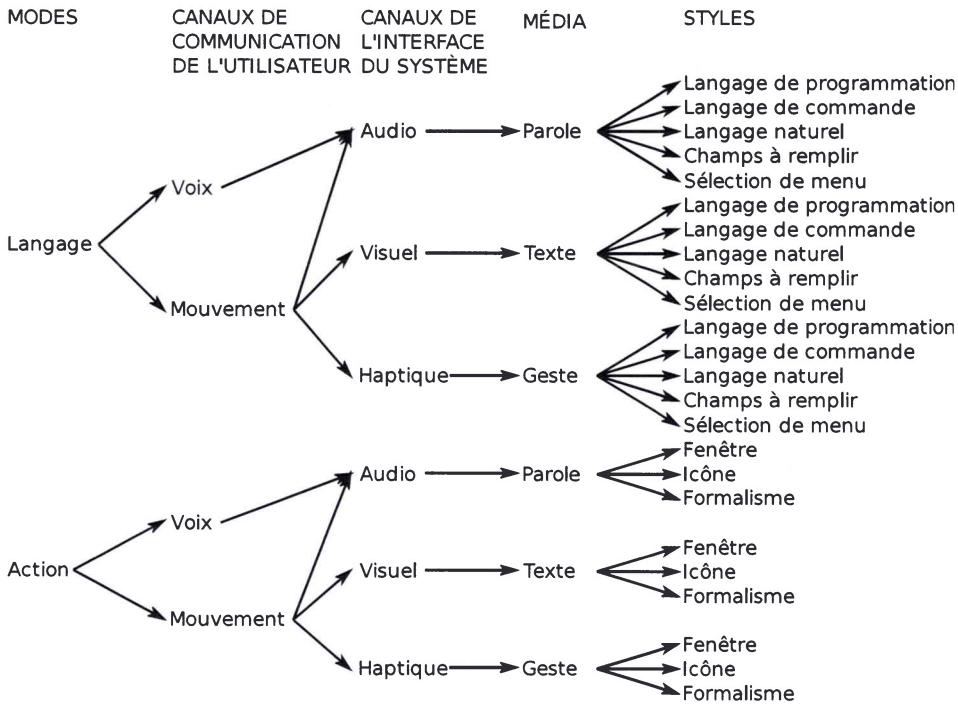


Figure 6 – Espace de conception de Frohlich (Entrée)

et le geste, et ceux correspondant au mode d'action sont le son, le graphique et la motion [124]. Enfin les *styles* d'interface sont des catégories d'interaction. Les styles de langage sont les langages de programmation, les langages de commandes, les langages naturels, les champs à remplir et les sélections de menu. Les styles d'action sont les fenêtres, les icônes et les formalismes. Cet espace est cependant limité : il ne permet pas de représenter toutes les interactions, ne précise pas comment sont ordonnancées les techniques d'interaction, leur niveau de fusion, etc.

La théorie des modalités de Bernsen.

Bernsen a élaboré un espace de conception dédié aux interactions en sortie [20]. Il définit les modalités de manière générative, c'est-à-dire que chaque modalité est dérivée de modalités génériques pures. Celles-ci sont formées d'un couple composé d'un média et d'un profil. Le média peut être graphique, sonore ou haptique. Le profil s'appuie sur quatre propriétés :

- Dynamique ou statique : présence ou absence de la dimension temporelle dans la représentation.
- Linguistique ou non linguistique : présence ou absence d'éléments lexicaux, syntaxiques, sémantiques ou pragmatiques dans la représentation.
- Analogue ou non analogue : présence ou absence de ressemblance entre la représentation et l'objet réel qu'elle représente.
- Arbitraire ou non arbitraire : absence ou présence de système sémantique pré-existant sur laquelle la représentation est basée.

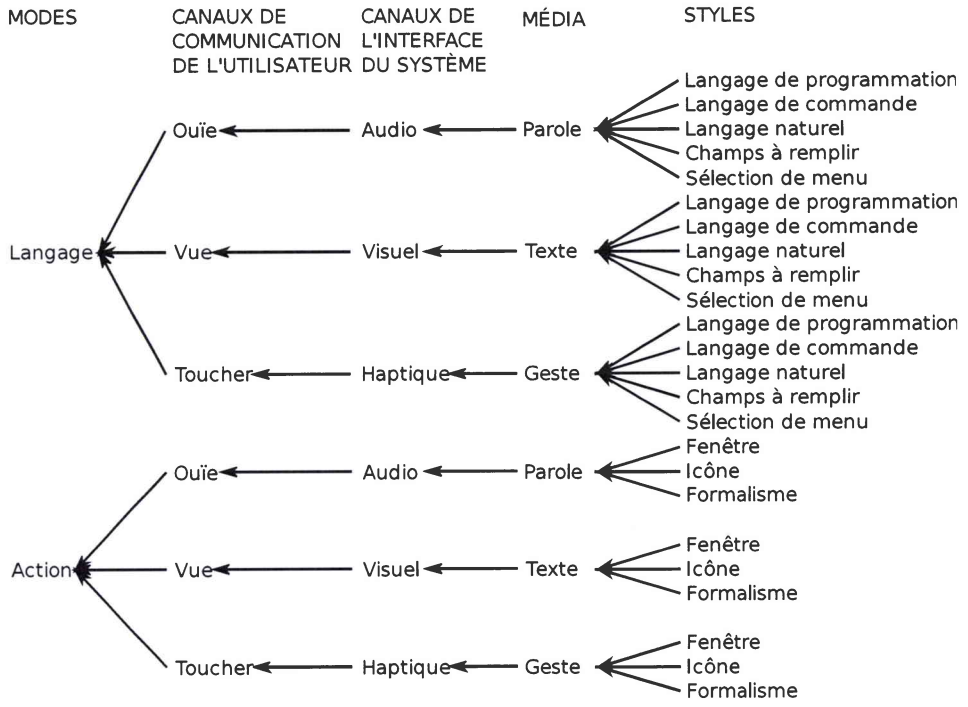


Figure 7 – Espace de conception de Frohlich (Sortie)

Parmi les 48 modalités génériques pures obtenues à l'aide de toutes les combinaisons, Bernsen en élimine certaines qu'il juge incohérentes. Par exemple les modalités linguistiques et analogiques ne peuvent pas être arbitraires. Cependant il élimine aussi les modalités du toucher statiques. Or comme nous le verrons dans les chapitres suivants, il est possible que le temps ait une imfluence ou non dans une représentation utilisant le sens du toucher. Il serait donc intéressant de ne pas restreindre cette classification pour permettre cette distinction. Cette restriction s'explique par le fait que les systèmes interactifs utilisant le sens du toucher se sont développés principalement après ces travaux de Bernsen.

L'espace CASE.

L'espace CASE est un premier espace simple et centré système [34] (figure 8). Il a été créé collectivement lors des journées IHM'91. Les techniques d'interactions sont classées selon deux axe. Le premier axe traite de l'usage, avec deux valeurs : séquentiel et parallèle. Il s'agit de la manière d'ordonnancer les techniques d'interaction. Le second axe traite de l'interprétation et de la génération des techniques d'interaction. Les deux valeurs sont : combiné et indépendant, ce qui permet de décrire le niveau de fusion entre les techniques d'interaction. Les quatre combinaisons possibles dans cet espace sont Alterné pour les interactions séquentielles et combinées, Synergique pour les interactions parallèles et combinées, Exclusif pour les interactions séquentielles et indépendantes, et enfin Concurrent pour les interactions parallèles et indépendantes.

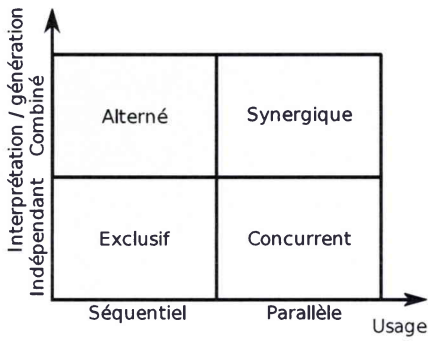


Figure 8 – Espace CASE

Quelques années plus tard, Nigay et Coutaz proposent une extension à cet espace en ajoutant un troisième axe [122] (figure 9). Ce troisième axe définit le niveau d'abstraction. La même donnée peut être traitée de différentes manières : une voix enregistrée peut être considérée comme un simple son, mais aussi comme un texte si le système sait interpréter le langage. Cet axe ne fournit cependant que deux valeurs, selon que la donnée ait une signification (« sens ») ou qu'elle soit brute (« pas de sens »). Un système utilisant des interactions ayant un sens est dit multimodal, et dans le cas contraire multimédia. Nigay [121] reporte une autre distinction entre multimodal et multimédia, où le terme multimodal désigne les interactions en entrée et multimédia les interactions en sortie. Dans la suite de cette étude nous utiliserons la première définition de multimodal, à savoir les interactions ayant un sens. De nombreuses extensions ont été créées depuis. Certaines ont pour but d'affiner le modèle, comme MSM (Multi-Sensori-Moteur) [40] qui ajoute trois axes pour le sens de l'interaction, le nombre de dispositifs et le contexte. D'autres modélisent des interactions particulières. Par exemple Bailly et al. [12] proposent 2M, un espace dédié aux interactions bi-manuelles.

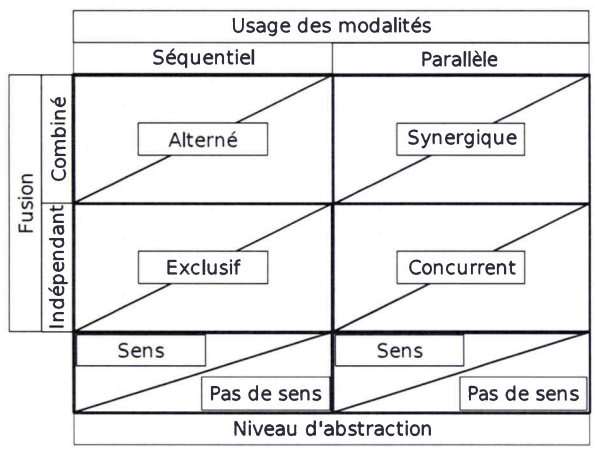


Figure 9 – Extension à CASE

2.2.2. Coopération entre modalités

L'espace TYCOON.

Martin définit six types de coopération au sein de l'espace *TYCOON*, en prenant cette fois-ci un point de vue à la fois du côté système et du côté utilisateur [109, 107]. Les propriétés ci-dessous permettent à la fois de caractériser et de concevoir les techniques d'interaction d'un système :

- *Transfert* : quand une modalité produit une information qui est utilisée par une autre modalité on dit alors qu'elles coopèrent par transfert.
- *Équivalence* : quand une information peut être traitée indifféremment par plusieurs modalités on dit alors qu'elles coopèrent par équivalence.
- *Spécialisation* : quand un type d'information est toujours traité par la même modalité on dit alors que les modalités coopèrent par spécialisation.
- *Redondance* : quand une information est traitée par plusieurs modalités à la fois on dit alors que les modalités coopèrent par redondance.
- *Complémentarité* : quand les différentes parties d'une information sont traitées indépendamment par plusieurs modalités on dit alors que les modalités coopèrent par complémentarité.
- *Concurrence* : quand différentes informations indépendantes sont générées simultanément par plusieurs modalités on dit alors que les modalités coopèrent par concurrence. Ce type de coopération a été rajouté dans une étude ultérieure [108].

Les propriétés CARE.

Coutaz *et al.* [42] utilisent cette approche à la fois orientée utilisateur et système, nommé *CARE* (figure 10). Il affine les spécifications au niveau de la fission et fusion des informations, ainsi qu'au niveau des relations temporelles entre les interactions. Avant de définir les catégories d'interactions, les auteurs définissent quelques notions. Un *état* est un vecteur de valeurs caractérisant l'état courant du système. Un *agent* est un utilisateur, un système ou une partie d'un système capable d'initier une action. Un *but* est un état qu'un agent veut atteindre. Les modalités sont classifiées en quatre catégories :

- *Complémentaires* : toutes les modalités sont nécessaires pour atteindre un but. Elles peuvent être utilisées de manière séquentielle ou parallèle.
- *Assignée* : une seule modalité permet d'atteindre un but. Dans ce cas l'utilisateur n'a pas le choix de la modalité.
- *Équivalentes* : n'importe quelle modalité permet d'atteindre le but souhaité.
- *Redondantes* : les modalités sont équivalentes et utilisées de manière séquentielles ou parallèles. La différence avec modalités équivalentes est que dans le cas des modalités équivalentes l'utilisateur n'utilise qu'une seule des modalités de l'ensemble.

2.2.3. Conclusion

Les espaces présentés permettent à la fois de décrire les interactions multimodales, et d'exposer les divers problèmes liés à leur création. D'un côté les espaces comme celui de Frohlich adoptent le point de vue de l'utilisateur. Ils permettent de modéliser la manière dont l'utilisateur agit sur le système.

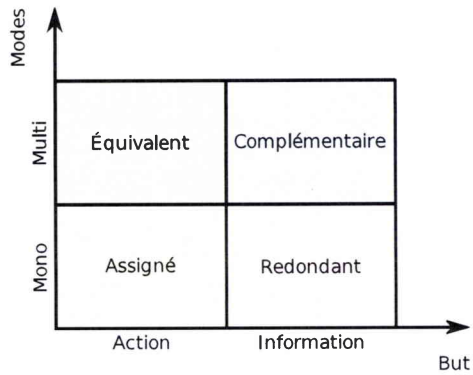


Figure 10 – Espace CARE

D'un autre côté les espaces comme CASE adoptent le point de vue du système. Ils définissent l'organisation des interactions par le système. Or lors de la conception d'un système multimodal il est aussi important de se préoccuper des deux points de vue. C'est pourquoi les espaces comme TYCOON ou CARE permettent de modéliser la combinaison des modalités en prenant en compte les deux aspects. Sans toujours en retenir la finesse de description, ces espaces ont l'avantage de se préoccuper à la fois de l'utilisateur et du système. Nous avons choisi d'utiliser l'espace CARE lorsque nous décrirons l'intégration de techniques d'intégration dans des systèmes multimodaux.

Comme nous avons pu le constater il existe de nombreuses modalités, que ce soit en entrée ou en sortie. Le travail présenté dans la suite sera focalisé sur les interactions en sortie, et en particulier sur celles utilisant le retour haptique. Il convient donc de décrire l'haptique, que ce soit du point de vue sensoriel, du point de vue des périphériques permettant d'utiliser l'haptique dans un système, ainsi que de l'utilisation de l'haptique et notamment dans les systèmes pour déficients visuels.

3. L'haptique

L'être humain possède cinq sens pour appréhender le monde qui l'entoure. Il est communément admis que mis à part chez les enfants de bas âge, le sens le plus utilisé est la vue. Nous pouvons appréhender l'espace autour de nous grâce à ce sens, nous reconnaissons les objets et les personnes dans une grande variation de conditions (position, distance, luminosité, etc.). Ce sens est si prépondérant qu'il peut influer sur la perception d'autres sens comme le sens auditif par exemple [110]. Ce dernier est le second sens le plus utilisé chez l'humain. Nous l'utilisons couramment comme moyen de communication, si bien que de nombreux langages ont été développés et sont utilisés quotidiennement. Les personnes atteintes de déficience visuelle compensent en partie leur handicap avec ce sens. Par exemple les aveugles développent un « sens de l'obstacle » qui leur permet de détecter la présence ou l'absence d'obstacles [95]. Le sens suivant est l'odorat. Il permet d'apprendre des informations importantes sur l'environnement proche. Par exemple il est utilisé pour attirer l'attention en cas de fuite de gaz, car le gaz de ville n'a pas d'odeur à l'origine : un parfum désagréable est ajouté afin que l'on puisse détecter une fuite et ainsi réagir de manière plus

ou moins appropriée. Le toucher est le quatrième sens le plus utilisé. Bien que ce sens permette d'acquérir beaucoup d'informations, il est peu utilisé. Une raison simple est que contrairement à la vue, l'ouïe et l'odorat, le toucher nécessite un contact physique pour transmettre une information. La communication tactile est donc très intime, ce qui peut expliquer les réticences à utiliser ce sens. Le dernier sens est le goût. À l'instar du toucher, ce sens nécessite un contact physique. Par contre l'information ne peut être recueillie que par la bouche, ce qui peut expliquer que ce sens est utilisé de manière limitée. De plus le fait que l'ingestion de certaines substances peut conduire à des complications médicales voire à la mort limite les sujets sur lesquels on peut exercer ce sens. La rareté de l'utilité de ce sens en comparaison des autres fait que le goût est un sens encore plus intime que le toucher. De ce fait la communication par le goût reste à usage réservé.

Les connaissances et la technologie actuelle font que ces sens ne sont pas tous utilisés équitablement dans les environnements virtuels. Il est aisé de créer de nombreux effets visuels grâce aux divers écrans et projecteurs. La plupart des interfaces sont d'ailleurs majoritairement visuelles, et le retour visuel est souvent le seul retour proposé. Le retour sonore est lui aussi facile à mettre en oeuvre grâce à des haut-parleurs. Le goût et l'odorat sont deux sens similaires. Les techniques connues permettant de les utiliser consistent en des systèmes libérant ou synthétisant des molécules connues. D'une part la synthèse des molécules n'est pas aisée, et d'autre part il n'existe pas de goût ou d'odeur de base permettant de recréer toutes les autres à l'instar des couleurs. Ces sens sont donc très peu utilisés dans les interfaces. Enfin le sens du toucher est difficile à reproduire car il regroupe une vaste quantité de sensations. Il est difficile pour un périphérique haptique de créer la totalité des sensations haptiques possibles, il faut donc utiliser des combinaisons de périphériques pour augmenter la palette de d'effets. De plus ces périphériques possèdent de nombreux désavantages : poids, encombrement, résolution, prix, etc.

Le travail présenté ici est centré sur le sens du toucher. C'est pourquoi nous allons dans un premier temps étudier son mode de fonctionnement. La science qui s'intéresse à l'étude du toucher est l'*haptique*. Le mot « haptique » vient du grec « haptesthai » qui signifie « toucher et manipuler » [155]. Le sens du toucher est souvent divisé en deux catégories. La première concerne ce que l'on ressent par les mécanorécepteurs de la peau : le sens *tactile*, ou *cutané*. La deuxième concerne ce que l'on ressent par les muscles, les tendons et les articulations : le sens *kinesthésique*. Oakley et al. [129] proposent des définitions précises pour les principaux termes utilisés en haptique :

Définition 1 *Haptique* : relatif au sens du toucher.

Définition 2 *Proprioceptif* : relatif aux sensations sensorielles sur l'état du corps (inclus les sensations cutanées, kinesthésiques et vestibulaires).

Définition 3 *Vestibulaire* : afférent à la perception de la position, l'accélération et la décélération de la tête.

Définition 4 *Kinesthésique* : sensation du mouvement. Relatif aux sensations issues des muscles, des tendons et des articulations.

Définition 5 *Cutané* : afférent à la peau elle-même ou à la peau en tant qu'organe sensoriel. Ceci inclut les sensations de pression, de température et de douleur.

Définition 6 *Tactile* : afférent au sens cutané, mais plus spécifiquement à la sensation de pression que de température ou de douleur.

Définition 7 *Retour de force* : relatif à la production mécanique d'information ressentie par le système kinesthésique humain.

Le toucher est aussi bien utilisé comme modalité d'entrée, avec des écrans sensitifs par exemple, qu'en sortie avec notamment les matrices de picots. Cependant dans cette étude nous nous intéresserons uniquement aux interfaces haptiques en sortie, bien que certains périphériques à sortie haptique permettent aussi d'acquérir des informations. On peut aussi souligner l'existence du retour *pseudo-haptique* [91, 92]. Il s'agit de donner une illusion haptique avec un autre sens. Par exemple en ralentissant le curseur de la souris lorsque l'on passe sur un bouton et que l'on accélère lorsqu'on le quitte : l'utilisateur a l'impression d'être monté sur le bouton, alors qu'aucun effet haptique n'a été utilisé.

3.1. La somesthésie

La *somesthésie* désigne l'ensemble des sensations que l'on peut ressentir à partir des récepteurs sensitifs du système *somatosensoriel*. Ces récepteurs peuvent être classés de plusieurs manières, nous utiliserons une classification par emplacement afin d'identifier les sensations tactiles et les sensations kinesthésiques. La *peau* est le plus gros organe du corps humain. Elle a une superficie qui peut faire jusqu'à $2m^2$ et peser entre 3 et $5kg$ [89]. Elle est constituée de deux parties : le *derme* (interne) et l'*épiderme* (externe). On trouve des récepteurs sensitifs dans ces deux parties. L'*hypoderme* est une troisième couche qui relie la peau aux *muscles* et aux *os*, qui contient des mécanorécepteurs, mais qui n'est pas considérée comme faisant partie de la peau. Les *tendons* sont des cordons qui relient les os aux muscles.

Type	**FAI**	**FAII ou PC**	**SAI**	**SAII**
Terminaison nerveuse	Meissner	Pacini	Merkel	Ruffini
Stimulus	Déplacement de la peau	vibrations de hautes fréquences	bords, points, coins, courbes	étirements de la peau
Fréquences	$1 - 300Hz$	$10 - 1000Hz$	$0 - 100Hz$	$0 - ?Hz$
Précision spatiale	$3mm$	$> 10mm$	$0{,}5mm$	$> 7mm$

Table 1 – Propriétés des mécanorécepteurs de la peau, adaptées de [29]

3.1.1. Les mécanorécepteurs

Il existe plusieurs types de récepteurs sensoriels [88, 101]. Parmi eux il y a les *mécanorécepteurs* (figure 11), dont une partie se trouve dans la peau, alors que d'autres se trouvent dans les muscles et les tendons. Leurs propriétés sont résumées dans la table 1. Il s'agit de nerfs spécialisés possédant des terminaisons à la place des dendrites qui leur donnent des particularités. On peut classer les multiples mécanorécepteurs existants par rapidité d'adaptation et par taille. Les corpuscules de Ruffini (récepteurs *SAII*) ont une adaptation lente et sont étendus. Ils permettent de sentir des pressions et des étirements de la peau. Les disques de Merkel (récepteurs *SAI*) sont aussi à adaptation lente, mais sont étroits. Ils répondent à des pressions tactiles avec une haute résolution. Ils sont mis à contribution pour lire du Braille par exemple. Les corpuscules de Pacini (récepteurs *FAII* ou *PCs*) sont à adaptation rapide et sont étendus. Ils nous informent sur les pressions fortes et les vibrations à fréquence élevée. Enfin les corpuscules de Meissner (récepteurs *FAI*) sont à adaptation rapide et sont étroits. Les informations qu'ils nous fournissent permettent de sentir les changements de texture, et les vibrations lentes.

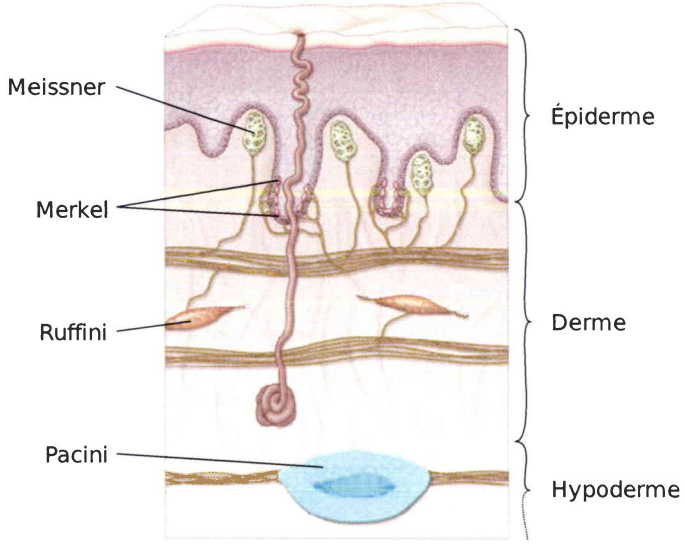


Figure 11 – Les mécanorécepteurs de la peau [64]

D'autres catégories de mécanorécepteurs sont responsables des sensations kinesthésiques, que l'on appelle aussi *propriocepteurs*. Les *fuseaux neuromusculaires* sont des fibres musculaires modifiées qui sont sensibles à l'allongement des muscles. Les *organes tendineux de Golgi* sont des mécanorécepteurs situés entre les muscles et les tendons. Ils réagissent aux variations de la force contractile des muscles. Ces récepteurs permettent à un individu de connaître la position relative des différentes parties du corps. L'ensemble des mécanorécepteurs permettent d'acquérir une grande quantité d'informations sur un objet : forme, taille, poids, texture, etc.

3.1.2. Les thermorécepteurs

Les *thermorécepteurs* sont un autre type de récepteurs sensoriels [88]. Ils se trouvent dans la peau et permettent de ressentir la température cutanée. Il y a deux types de récepteurs : les récepteurs au froid qui sont sensibles aux températures entre 10° et 35°, et les récepteurs au chaud qui sont sensibles aux températures entre 30° et 46°. Ces récepteurs sont situés dans la peau si bien que la sensation de température est souvent assimilée à une sensation tactile, alors qu'elle est habituellement reconnue comme un sens à part.

3.1.3. Les nocicepteurs

La sensation de douleur est due à des stimuli de *nocicepteurs*. Il existe trois types de nocicepteurs : certains sont des mécanorécepteurs et permettent de détecter les douleurs dues aux déformations et aux pressions intenses. D'autres sont des thermorécepteurs et signalent des températures inhabituelles. Enfin il y a les récepteurs polymodaux qui sont autant sensibles à la température qu'aux autres douleurs. Ces récepteurs ont des terminaisons libres. Tout comme la sensation de température, la sensation de douleur est souvent classifiée comme une sensation tactile alors que c'est aussi un sens considéré comme à part. Notons que les différents nocicepteurs se trouvent aussi bien dans la peau que dans les muscles ou les tendons. Il existe deux autres catégories de récepteurs : les *photorécepteurs* et les *chemorécepteurs*, mais ils ne procurent aucune sensation tactile.

3.1.4. Conclusion

La combinaison de ces sensations permet de sentir une palette variée de sensations tactiles [69] : piquement, douceur, humidité, glissement, frottement, aspérité. Il est aussi possible de ressentir des propriétés locales des objets : formes, coins, creux et bosses. Notre étude ne s'intéresse qu'à l'haptique en sortie. Nous allons donc étudier les principales catégories de périphériques permettant de donner un retour haptique.

3.2. Taxonomie des périphériques haptiques

L'utilisation de l'haptique comme modalité de sortie dans les interfaces consiste à utiliser des périphériques de sortie qui stimulent différents récepteurs sensoriels afin de procurer à l'utilisateur des sensations tactiles et kinesthésiques. Les technologies utilisées pour stimuler ces récepteurs sont variées. Il existe beaucoup de classifications dans la littérature. Hayward *et al.* [69] répertorie les périphériques par type, alors que Lécuyer [91] recense les périphériques tactiles par sensation et les périphériques à retour de force par technologie. La classification ci-dessous différentie les périphériques qui procurent des sensations tactiles et celles qui procurent des sensations kinesthésiques. Pour chacune de ces deux catégories, les périphériques sont regroupés par technologie utilisée.

Il serait illusoire de faire un aperçu exhaustif des périphériques haptiques existants tellement leur diversité est grande. C'est pourquoi nous allons examiner ceux que nous utiliserons par la suite, ainsi que quelques autres fonctionnant sur un principe similaire.

3.2.1. Les périphériques tactiles

Les périphériques tactiles permettent de ressentir plusieurs types d'informations. Parmi elles on retrouve les alphabets comme le Braille, des textures, des vibrations, etc. Il existe de nombreuses technologies permettant d'afficher ce genre d'informations.

Les matrices de picots.

Ces appareils sont les évolutions des livres en Braille. Il s'agit de matrices d'aiguilles (appelées picots) contrôlables individuellement afin de pouvoir afficher le motif souhaité. Généralement ces dispositifs n'autorisent que deux états pour chaque picot : *levé* et *baissé*. Les technologies utilisées sont coûteuses, et donc la taille et la résolution de ces afficheurs sont souvent limitées. Virtouch Ltd a développé deux souris utilisant ce principe. La première est appelée *Virtouch* (figure 12 à gauche). C'est un prototype, avec trois matrices de 4×8 picots. Elle possède aussi huit boutons pour la navigation. Elle a été améliorée sur plusieurs points, ce qui a donné la *VTPlayer* (figure 12 à droite). La taille a été réduite considérablement, même si elle reste toujours volumineuse en comparaison avec une souris conventionnelle. Cette réduction de taille a dû contraindre à enlever une matrice, et à ne mettre que 4×4 picots sur les deux restantes. Les picots sont des cylindres émoussés de $1mm$ de diamètre, se levant d'$1mm$, et étant espacés d'environ $2mm$. Enfin le nombre de boutons a été réduit à quatre. Cependant cet appareil n'a pas eu le succès commercial escompté. Une des raisons est le public visé très restreint. Ensuite les logiciels fournis avec n'étaient pas de grande qualité, et enfin selon Jansson [84] la navigation avec une souris n'est pas adaptée aux utilisateurs aveugles.



Figure 12 – Souris Virtouch et VTPlayer

Hanneton *et al.*. [67] ont créé un dispositif appelé *Tactos* avec une tablette graphique pour l'exploration et deux matrices de 2×4 picots pour le rendu tactile (figure 13 à gauche). Le but de ce dispositif est de permettre à des utilisateurs de reconnaître des formes géométriques. La méthode utilisée consiste à transcrire les pixels noirs en picots levés et les pixels blancs en picots baissés. Les auteurs on identifié plusieurs méthodes chez les utilisateurs afin d'explorer les formes. Ziat *et al.* ont conçu des techniques de zoom afin d'améliorer l'exploration [179]. Ces travaux seront détaillés dans

la section 3.3.3. De plus dans le chapitre II nous proposerons une autre technique d'exploration de formes géométriques, que nous comparerons à la technique consistant à transcrire les pixels.

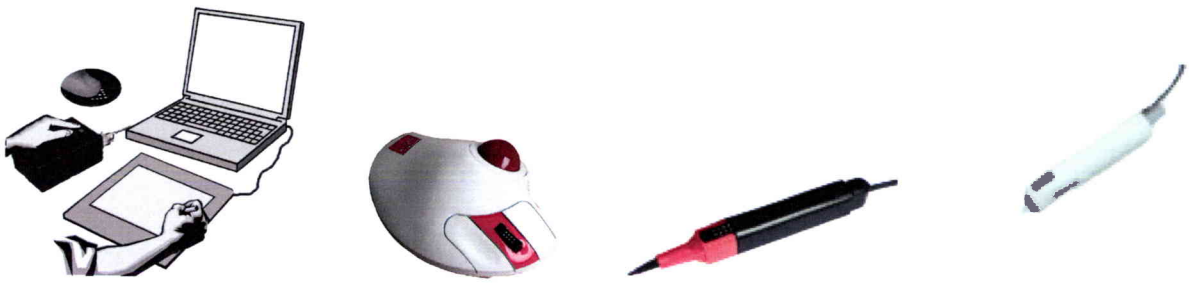


Figure 13 – Tactos, Tactiball, Tactipen et Tactitab

En s'inspirant de Tactos, Lecolinet et Mouret [90] ont fabriqué trois périphériques en utilisant des matrices de 2×4 picots (figure 13 à droite). Le premier est *TactiBall*, un trackball ayant la matrice à la place de la molette. Le second est *TactiPen*, un stylet pour PDA constitué d'un feutre surligneur dont le contenu a été remplacé par une matrice. Le troisième est *TactiTab*, un stylet pour tablette graphique. Le but est de donner une dimension tactile aux interfaces. Le point positif de ces dispositifs est leur variété dans le sens où ils sont intégrés à d'autres dispositifs déjà répandus. Le point négatif est la faible résolution de la matrice, notamment dans le sens de la largeur.

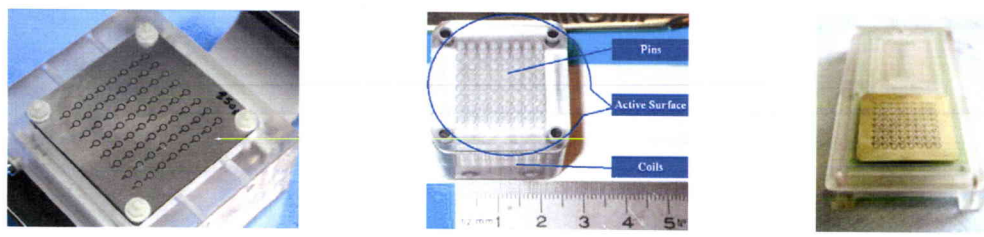


Figure 14 – Trois prototypes de VITAL

Benali-Khoudja et Hafez ont utilisé un autre procédé pour construire une série de matrices de picots appelées *VITAL* [16]. Le principe est d'utiliser des bobines pour créer un champ magnétique afin de lever le picot. La conception a été réalisée de manière a simplifier la fabrication et à permettre la production de masse. Le premier prototype (figure 14 à gauche) dispose d'une membrane magnétique en surface, dont des pastilles disposées au dessus de la matrice de bobines vibrent lorsqu'un champ magnétique est créé par la bobine du dessous. Le deuxième prototype (figure 14 au centre) permet aux bobines d'actionner des picots grâce à des cylindres de transmission [18]. Enfin la troisième génération de VITAL (figure 14 à droite) utilise des bobines sur plusieurs niveaux de circuits imprimés afin d'augmenter le nombre de boucles. En augmentant le nombre de boucles il est possible de baisser le courant à utiliser et ainsi simplifier l'électronique du dispositif.

Les dispositifs vibreurs.

Les dispositifs les plus courants sont les vibreurs. Il existe plusieurs techniques pour produire des vibrations. L'une d'entre elle consiste à monter une came décentrée sur l'axe d'un moteur. Ce système est utilisé dans la vie de tous les jours : les téléphones portables l'utilisent pour remplacer les sonneries auditives. Les jeux vidéo utilisent aussi ce système dans les manettes de jeux afin, d'augmenter le réalisme et l'immersion du joueur dans l'environnement. Le joueur reçoit des vibrations lorsqu'il heurte un obstacle par exemple. Ce système est aussi utilisé dans le gant *CyberTouch* d'Immersion [76] (figure 15 à gauche), qui possède un vibreur par doigt, et un pour la paume. La bande passante se situe entre $0Hz$ et $125Hz$.



Figure 15 – Cybertouch, Optacon et VBW32 Tactor

Certains périphériques utilisent les vibrations afin d'afficher des matrices de points à l'instar du Braille, mais sans lever de picots. L'*Optacon* (figure 15 au centre) est un des premiers appareils fonctionnant sur ce principe [99]. Il est constituée d'une petite caméra que l'utilisateur déplace sur le texte qu'il souhaite lire, et d'une matrice tactile qui affiche la transcription de ce qui est capté par la caméra. La matrice est constituée de 6×24 picots qui vibrent individuellement si le pixel correspondant est sombre.

D'autres périphériques utilisent le même principe qu'un haut parleur. Le vibreur *VBW32 Tactor* de Tactaid [161] (figure 15 à droite) vibre entre $100Hz$ et $800Hz$. Il est placé directement sur la peau pour stimuler les récepteurs sensitifs. Des études ont été menées afin de trouver différentes fréquences différentiables, notamment pour des utilisateurs déficients visuels [126]. D'autres études utilisant ce genre de périphériques seront discutées plus en détails dans la section 3.3.4.

Lévesque et Hayward ont étudié une technique permettant de donner l'impression de sentir des creux et des bosses en étirant ou en compressant la peau des doigts [96]. Cette technique a abouti au périphérique *STReSS* [132] (figure 16 à gauche), une matrice de lames piézoélectrique. La propriété des matériaux piézoélectriques est de se déformer sous l'action d'un courant électrique et inversement. Le principe est donc d'écarter ou de rapprocher les lames pour étirer ou comprimer la peau. Le même principe a été utilisé par Hayward pour construire l'*ABV* (figure 16 à droite, Afficheur de Braille Virtuel, *VBD* en anglais) [68]. Cet afficheur permet de faire sentir des picots virtuels et donc

permet de simuler du Braille [97]. Plus tard le *STReSS* a été amélioré et a donné le *STReSS²* [172] (figure 16 au centre). Cette version est plus facile à fabriquer, plus modulaire et plus compacte.

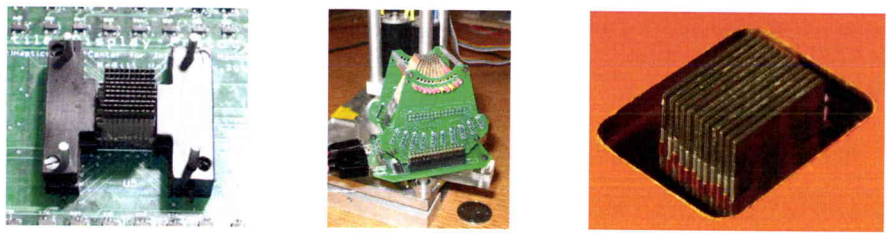


Figure 16 – STReSS, STReSS² et ABV

Les appareils à charge électrique.

Certains périphériques utilisent des impulsions électriques pour stimuler les récepteurs sensitifs. Bach-y-Rita est un précurseur dans le domaine, et a lancé l'idée de substitution sensorielle avec l'idée que l'on voit avec le cerveau et non avec les yeux [10]. Le principe est que l'image perçue par les yeux ne va pas plus loin que la rétine. Au delà, l'image n'est qu'un signal électrique transitant sur le système nerveux vers le cerveau. Ainsi pour substituer la vision à des personnes non voyantes, Bach-y-Rita a cherché comment produire le même type de signal. L'utilisation du sens tactile est devenu un choix intéressant, car il est peu utilisé, et peu de gens ont des déficiences totale pour ce sens. Il s'agit de trouver une zone pour placer un appareil destiné à stimuler les récepteurs sensitifs.



Figure 17 – Homonculus sensoriel, Musée d'Histoire Naturelle, Londres

L'homonculus sensitif (figure 17) est une représentation du corps humain où la taille des organes est proportionnelle à la taille des parties du cortex responsables de leur sensation tactile. On peut constater que les deux zones les plus sensibles sont les mains et la bouche, en particulier la langue. Sachant cela, Bach-y-Rita a construit le *Tongue Display Unit* (TDU, figure 18) [9], une matrice de 7×7 électrodes (12×12 dans une version plus récente) qui se place sur la langue. Cet appareil peut autant être utilisé par des non voyants que par des chirurgiens qui ont besoin de leurs mains pour opérer, et de leurs yeux pour observer le patient [151].

Kajimoto décide d'utiliser une autre région du visage : le front [86]. Il reprend l'idée de l'Optacon en combinant une caméra avec un dispositif tactile, formant le *Forhead Retina System* (FRS).



Figure 18 – Tongue Display Unit

L'image acquise par la caméra est seuillée et une impulsion électrique est envoyée sur les électrodes représentant un pixel noir. La matrice a une taille de 32×16 points. Il est cependant difficile de reconnaître une forme plus compliquée qu'une ligne avec cet appareil, et la sensation devient vite désagréable.

Les appareils à pression.

La pression sur une large surface est un aspect tactile que nous n'avons pas encore exploré dans cette étude. Il existe des périphériques permettant de sentir ce genre de sensations. Le gant *Teletact* d'Advanced Robotics Research Ltd. (figure 19 à gauche) en est un exemple. Il s'agit d'un gant possédant des coussins gonflables. L'utilisateur peut sentir des pressions en fonction du gonflement et dégonflement de ces coussins.

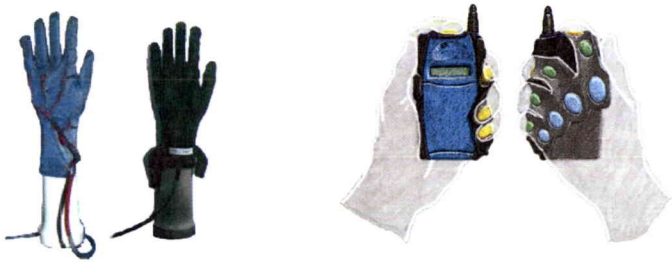


Figure 19 – Gant Teletact et ComTouch

Le *Comtouch* est un autre périphérique utilisant les pressions (figure 19 à droite) [37]. Il s'agit d'un système augmentant les fonctionnalités habituelles d'un téléphone portable avec un système de communication tactile. Il est composé d'une partie émettrice que l'utilisateur peut serrer, et une partie réceptrice qui serre la main de l'interlocuteur distant.

3.2.2. Les périphériques à retour de force

Certains périphériques permettent de ressentir des objets virtuels. Une grande partie d'entre eux sont aussi des *périphériques de pointage*, c'est-à-dire des périphériques d'entrée permettant de déplacer un curseur. Le nombre de *degrés de liberté* (DOF) désigne le nombre de coordonnées acquises par les différents capteurs du périphériques, ou le nombre d'axes sur lesquels sont rendues les forces. Voici une liste sélective de types de périphériques à retour de force, agrémentée de quelques exemples.

Contrôleurs de jeux.

Les contrôleurs de jeux à retour de force sont les périphériques à retour de force les plus répandus parmi le grand public. Ils sont à la fois beaucoup moins chers et beaucoup moins précis que leurs homologues utilisés dans le milieu de la recherche. Les deux principaux périphériques à retour de force de cette catégorie sont les joysticks et les volants à retour de force.

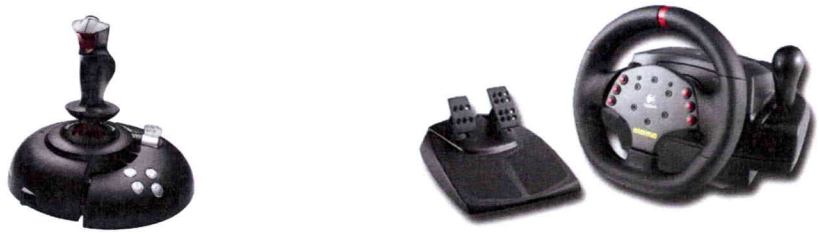


Figure 20 – Joystick à retour de force et volant à retour de force

Les *joysticks* ont de deux à quatre degrés de liberté en entrée : avant/arrière, droite/gauche, parfois le roulis et gaz. Seuls les deux premiers de ces degrés de liberté ont un retour de force. Les *volants* ont un degré de liberté, que ce soit en entrée ou en sortie. Les pédales, leviers de vitesses ou autres satellites ne sont que de simples commandes. Ces types de périphériques sont déclinés en de nombreux modèles chez les principaux fabriquants de contrôleurs de jeu dont les principaux sont Logitech [100], Saitek [154], Thrustmaster [163] et Microsoft [114].

Bras articulés.

Les bras articulés permettent de contrôler la position d'un ou plusieurs points dans l'espace. Chaque axe est relié à un moteur pas à pas pour rendre une force. À l'inverse ils possèdent des capteurs pour connaître la position sur l'axe. Les périphériques de ce type ont souvent $3DOF$ (translations dans l'espace) ou $6DOF$ (translations et rotations dans l'espace). Selon la gamme, ils ont un retour de force sur $3DOF$ ou $6DOF$.

Un des périphériques les plus répandus de cette catégorie est le *PHANToM* (table 2). Il a été conçu par Massie et Salisbury [111], et est commercialisé par la société Sensable [157]. Un stylet ou un dé à coudre est monté sur un bras articulé pour contrôler un curseur. Les spécifications techniques des différentes versions sont détaillées dans la table 2. Ils ont tous $6DOF$ en entrée et $3DOF$ en sortie, sauf les versions Premium 1.5 et 3.0 qui peuvent avoir $6DOF$ en sortie. Il est possible d'ajouter $1DOF$ correspondant au pincement sur ces derniers grâce à une extension prévue à cet effet. Un des désavantages de ces périphériques est qu'ils ne possèdent qu'un seul point de contact. Cependant il est possible de contourner ce problème en combinant plusieurs PHANToM [173]. L'espace de travail s'étend de $1{,}34dm^3$ pour la version OMNI à $198dm^3$ pour la version Premium 3.0. Ainsi l'espace de travail des versions OMNI, Desktop et Premium 1.0 correspond environ au mouvement du poignet, celui de la version Premium 1.5 correspond au mouvement de l'avant bras et enfin celui de la version Premium 3.0 correspond au mouvement du bras entier. Sur la plupart des périphériques à retour

Version	OMNI	Desktop	Premium 1.0
Photo			
Espace de travail	$160 \times 120 \times 70mm$	$160 \times 120 \times 120mm$	$254 \times 178 \times 127mm$
Résolution de position	$0{,}055mm$	$0{,}023mm$	$0{,}03mm$
Force maximale	$3{,}3N$	$7{,}9N$	$8{,}5N$
Force en continu	$0{,}88N$	$1{,}75N$	$1{,}4N$
DOF retour de force	3	3	3

Version	Premium 1.5	Premium 1.5 High Force	Premium 3.0
Photo			
Espace de travail	$381 \times 267 \times 191mm$		$838 \times 584 \times 406mm$
Résolution de position	$0{,}03mm$	$0{,}007mm$	$0{,}02mm$
Force maximale	8,5	$37{,}5N$	$22N$
Force en continu	$1{,}4N$	$6{,}2N$	$3N$
DOF retour de force	3/6	3/6	3/6

Table 2 – Spécifications des périphériques PHANToM [157]

de force, il est courant de distinguer la force maximale et la force en continu. La force maximale correspond à la plus grande force que le périphérique peut produire en pic, alors que la force en continu représente la force que le périphérique peut exercer sur une longue durée. La version OMNI est la plus faible avec une force maximale de $3{,}3N$ pour une force en continu de $0{,}88N$, alors que la version Premium 1.5 High Force atteint $37{,}5N$ en pointe et $6{,}2N$ en continu. Dans les chapitres III et IV nous utiliserons les versions OMNI et Desktop.

Garrec *et al.* ont construit un périphérique similaire au PHANToM, appelé *Virtuose* [65, 62] (figure 21 à gauche). Selon les versions il possède $3DOF$ ou $6DOF$ en entrée et en sortie. La version 6D Desktop est formée de 3 bras articulés, formant un espace de travail sphérique de $120mm$ de diamètre. Il peut exercer une force maximale de 7 à $10N$ pour une force continue de 1,4 à $3N$. Ses caractéristiques sont comparables à la version Desktop du PHANToM. La version 3D15-25 possède un espace de travail sphérique de $250mm$ de diamètre sur $3DOF$, avec une force maximale de $15N$ et

une force continue de $5N$. En comparaison avec la gamme PHANToM, il se positionne entre les deux versions Premium 1.5. Le Virtuose 6D35-45 a un espace de travail de $450mm$ de diamètre, exerce une force maximale de $35N$ et une force continue de $10N$. Il se situe entre le PHANToM Premium 1.5 High Force et le PHANToM Premium 3.0. Enfin la version 6D40-40 offre un espace de travail de $400mm$ de diamètre, pour une force maximale de $100N$ et une force continue de $30N$. L'espace de travail de cette version est restreinte par rapport à celui du PHANToM Premium 3.0, mais il exerce des forces supérieures à toutes les versions du PHANToM.



Figure 21 – Virtuose et Cyberforce

Immersion a développé le *CyberForce* (figure 21 à droite) qui est un bras articulé qui contraint la main de l'utilisateur avec $3DOF$ pour la force et $6DOF$ pour la capture de translation et de rotation. Sa zone d'action est d'environ $30cm \times 30cm$ sur un arc de 133°. La force maximale est de $8{,}8N$, et la force continue n'est pas précisée par le constructeur.

Les pantographes.

Le principe du pantographe a été utilisé pour contrôler la position d'un point grâce à deux angles. Ramstein et Hayward ont créé un périphérique utilisant cette technique, appelé *Pantograph* (figure 22 à gauche) [144]. Il n'a que $2DOF$ et son espace de travail est de $10cm \times 16cm$, avec une force maximale de $10N$. Une version plus récente possède un espace de travail plus restreint : $10{,}4cm \times 6{,}6cm$ [35]. Il atteint une précision de déplacement à $10\mu m$. D'autres périphériques, destinés au grand public, ont été construits à partir de ces prototypes. Le *Pencat/Pro* (figure 22 au centre) est un stylet monté sur un pantographe, qui a trois degrés de liberté, dont deux avec retour de force. La *souris Logitech iFeel* et la *souris Wingman Force Feedback* (figure 22 à droite) sont des souris montées sur un pantographe. Elles ont aussi $2DOF$.

Le principe du pantographe a été réutilisé dans des périphériques à trois, six et sept degrés de liberté, commercialisés par la société Force Dimension [59]. Une poignée est accrochée à trois pantographes, et des extensions permettent d'utiliser les rotations et le pincement. L'*Omega* (figure 23 à gauche) a un espace de travail cylindrique de $16cm$ de diamètre et $11cm$ de profondeur, pour une force continue de $12N$. La force maximale n'est pas précisée par le constructeur. Ce périphérique existe en version $3DOF$ (translations uniquement), $6DOF$ (translations et rotations) et $7DOF$ (translations, rotations et pincement). La force maximale en pincement est de $8N$. Le *Delta* (figure 23 au centre) est une version haute gamme dont l'espace de travail est de $36cm$ de diamètre et $30cm$ de



Figure 22 – Pantograph, Pencat/Pro et souris Wingman Force Feedback

profondeur, pour une force continue de $20N$. Il existe en version $3DOF$ et $6DOF$. Enfin le *Falcon* (figure 23 à droite), commercialisé par Novint [128], est une version grand public dont l'espace de travail est un cube de $10cm$ de côté, avec une force maximale de $8{,}9N$. Dans sa version actuelle il ne possède que $3DOF$.



Figure 23 – Omega, Delta et Falcon

Systèmes pneumatiques.

Les systèmes pneumatiques ont l'avantage de créer des forces plus importantes. Moog FCS [116] a créé l'*Haptic Master* [164] (figure 24 à gauche), qui est un bras monté sur un système pneumatique à $3DOF$, avec un espace de travail de $80dm^3$. Grâce à sa technologie basée sur un système pneumatique, il peut exercer une force maximale de $250N$ et une force continue de $100N$. Ce gain de force par rapport au PHANToM est appréciable étant donné que son rayon d'action correspond au mouvement du bras. Cependant un des désavantages de ce système est son encombrement.

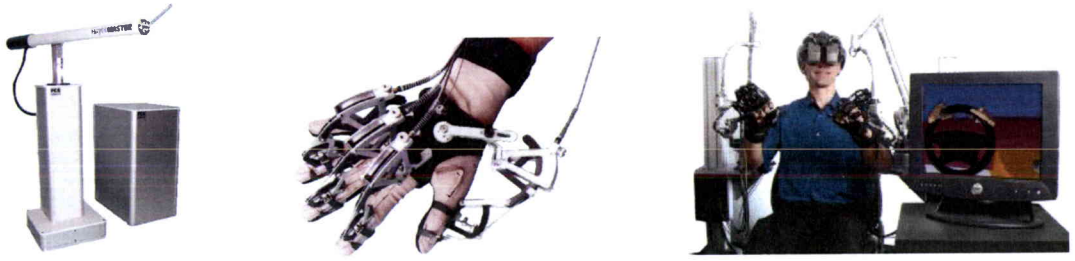


Figure 24 – Haptic Master, Cybergrasp et Haptic Workstation

Exosquelettes.

Les exosquelettes sont des périphériques à retour de force destinés à être portés. Ils permettent de contraindre les mouvements afin de donner une illusion de contact, ou d'ajouter leur puissance à celle des articulations. Il en existe une grande variété, nous ne présenterons que le *CyberGrasp* d'Immersion (figure 24 au centre). Il s'agit d'un exosquelette qui se fixe à la main, et qui permet de contrôler la position des doigts. La force maximale est de $12N$ par doigt. Il peut être utilisé en combinaison avec le CyberTouch pour ajouter un ressenti tactile, et le CyberForce pour contraindre la position de la main elle même. L'*Haptic Workstation* (figure 24 à droite) est un montage incluant tous ces périphériques pour les deux mains.

3.2.3. Conclusion

Nous avons évoqué dans cette section un aperçu de périphériques tactiles et à retour de force. Le sens tactile peut être simulé grâce à plusieurs technologies. En premier nous avons évoqué les matrices de picots. Le principe est de lever des picots afin de créer des pressions sur le doigt. L'information est représentée sous forme de grille, et peut servir à afficher du texte grâce à un alphabet *ad hoc* tel que le Braille. Il est aussi possible de représenter des informations sous forme de grille à l'aide d'appareils vibrants, cependant ils ne sont pas d'utilisation courante. Les appareils vibreurs les plus courants tels que les téléphones portables et les manettes de jeux se contentent d'un rendu tactile primitif. Nous avons aussi évoqué des méthodes moins répandues. L'une d'entre elles consiste à délivrer des décharges électriques pour afficher des informations sous forme de grille. Les appareils utilisant cette technique sont utilisés sur des parties très sensibles de la peau telles que la langue. Enfin nous avons évoqué une autre technique, consistant à comprimer des parties de la peau pour simuler le contact avec des textures.

Périphérique	**PHANToM**	**Virtuose**	**Omega Delta**	**Haptic Master**
Espace de travail	$16 \times 12 \times 7cm$ à $83{,}8 \times 58{,}4 \times 40{,}6cm$	Sphère de $12cm$ à sphère de $45cm$	Ø $16cm \times 11cm$ Ø $36cm \times 30cm$	$80dm^3$
Résolution de position	$0{,}007mm$ à $0{,}055mm$	Non précisé	$< 0{,}03mm$ $< 0{,}01mm$	$0{,}004mm$ $0{,}012mm$
Force maximale	$3{,}3N$ à $37{,}5N$	$7N$ à $100N$	Non précisé	$250N$
Force en continu	$0{,}88N$ à $6{,}2N$	$1{,}4N$ à $30N$	$12N$ $20N$	$100N$
DOF retour de force	3 à 6	3 à 6	3 à 7	3

Table 3 – Résumé des spécifications des périphériques de pointages à retour de force

La table 3 résume les principales caractéristiques des périphériques de pointage à retour de force de plus de $3DOF$ dont nous avons discuté. L'espace de travail, la résolution, la force maximale et en continu, ainsi que le nombre de degrés de libertés sont des points clés qui déterminent le choix du périphérique à utiliser. La plupart de ces périphériques possèdent une gamme de produits assez large pour couvrir différents types de besoins. En plus de ces appareils, nous pouvons souligner les exosquelettes tels que le Cybergrasp ainsi que le Cyberforce qui sont utilisés conjointement pour des applications de simulation principalement. Enfin le pantographe et ses dérivés permettent de bénéficier du retour de force sur $2DOF$, ce qui peut être suffisant selon l'utilisation.

3.3. Interactions haptiques

L'haptique a d'ores et déjà de nombreuses applications. Le rendu haptique peut être utilisé pour augmenter l'immersion dans une environnement de réalité virtuelle [93]. Il est aussi possible d'améliorer l'efficacité des interfaces visuelles et auditives en proposant de nouvelles techniques d'interaction [5, 143]. Enfin il est possible de remplacer une modalité quand celle-ci est indisponible pour diverses raisons : obscurité, surface d'affichage restreinte, environnement bruyant, handicaps visuels et auditifs, etc [165]. Nous allons nous intéresser d'une part à deux situations courantes d'utilisation de l'haptique : les simulations et la navigation d'interfaces. D'autre part nous allons étudier des techniques d'interaction du même type que celles que nous allons étudier dans les chapitres suivants, c'est-à-dire l'exploration d'une scène, et le codage d'information sous forme d'icônes en utilisant l'haptique.

3.3.1. Simulations

Le retour haptique est beaucoup utilisé dans le milieu médical. Des logiciels de simulations ont été développés afin de former des spécialistes. Par exemple Baillie, Crossan *et al.* [11, 43] ont travaillé sur un simulateur de palpation rectale de vache (figure 25), afin de former les vétérinaires bovins à la détection de certaines maladies et le suivi de la gestation. Williams *et al.* [174] ont développé un simulateur de palpation de dos pour ostéopathes. Dans les deux cas l'étudiant explore avec un PHANToM, et un spécialiste peut l'observer et ainsi le conseiller. Ce système inclut un mode de guidage permettant de rejouer des trajectoires.



Figure 25 – La vache haptique

La société SRI [160] a développé le système MEDFAST (Medical Emergency Forward Area Surgical Telepresence) [85] pour l'armée américaine. C'est un système de téléprésence pour effectuer des opérations chirurgicales à distance. Le principe est d'avoir un robot articulé dans un hôpital mobile (mobile army surgical hospital, MASH) sur le champ de bataille, et de doter le chirurgien d'une station d'opération par téléprésence (telepresence surgeon's workstation, TSW). Ainsi le chirurgien peut travailler dans de bonnes conditions, en sécurité, et avec précision.

3.3.2. Navigation d'interfaces

L'haptique peut aussi être utilisée pour ressentir une interface. Ainsi il est possible d'augmenter le réalisme d'une interface, tout en améliorant l'efficacité. Dennerlein *et al.* ont montré l'efficacité d'une souris à retour de force (FEELit) pour des tâches de précision [49]. L'aimantation de zones choisies de manière adéquate permet d'accélérer l'interaction.

Le retour tactile est aussi utilisé : Fukutomo et Sugimura utilisent un vibreur sur un PDA afin de le faire vibrer quand l'utilisateur clique sur un bouton [60]. Ceci permet à l'utilisateur de savoir s'il a bien cliqué dessus. Ils ont mesuré un accroissement de l'efficacité de 5% en milieu silencieux et 15% en milieu bruité. Nashel *et al.* [119] ont développé un système similaire où l'on ressent aussi le passage sur les boutons, et donc on sait si on est sur un bouton, sur le bord ou à côté. Plus tard, Brewster *et al.* ont mesuré l'efficacité d'un clavier virtuel sur PDA avec et sans retour tactile [27, 74]. Ils ont montré que le retour tactile permet de réduire significativement les erreurs de saisie. Ce genre de système est utilisé sur des consoles de jeux telles que la Wii de Nintendo [125]. Le contrôleur est une télécommande qui contient un vibreur. Les boutons et les touches de l'interface peuvent être ressentis grâce à des vibrations. Le principe d'inclure les composants créant l'effet tactile dans un stylet ou un contrôleur au lieu de le mettre dans l'écran a été étudié par plusieurs chercheurs. Lee *et al.* ont monté un solénoïde sur un stylo [94] afin de donner une sensation de clic à dureté variable. Un capteur de pression sur la pointe permet de connaître l'état du stylet et ainsi de calculer la force adéquate à appliquer. Ce système permet aux systèmes utilisant un stylet de faire ressentir l'interface.

3.3.3. Exploration

Dans un contexte plus général le guidage peut être utilisé pour apprendre à l'utilisateur toutes sortes de trajectoires. Bayart *et al.* ont développé une méthode d'apprentissage de trajectoire en quatre phases [14]. Dans la première phase l'utilisateur est complètement guidé par le système. Dans la seconde il essaye de reproduire la trajectoire, et le système n'intervient que lorsqu'il est perdu, c'est-à-dire qu'il dévie trop de la trajectoire. Dans la troisième phase l'utilisateur est replacé sur le chemin s'il s'en éloigne trop, sans recevoir d'indication sur la direction à prendre. Enfin dans la dernière phase, l'utilisateur n'est plus aidé. Ce système a été utilisé notamment pour faire un système permettant d'apprendre l'écriture de kanjis, et un système de guidage dans un labyrinthe.

La différence principale entre le rendu visuel et le rendu tactile est que le rendu visuel permet à la fois d'avoir une vue d'ensemble et une vue détaillée, alors que le rendu tactile ne permet d'avoir qu'un rendu à une seule échelle. Afin de contourner cette lacune, Ziat *et al.* se sont intéressés à

la possibilité de créer des interfaces tactiles zoomables. Le périphérique utilisé, appelé Tactos, est constitué d'une tablette graphique comme périphérique de pointage, et de deux cellules de 2×4 picots comme dispositif d'affichage tactile. Une étude préliminaire [179] montre qu'il est possible d'utiliser ce périphérique pour reconnaître des formes de taille entre $0{,}5mm$ et $2mm$. Par ailleurs les auteurs se sont rendus compte que les utilisateurs adaptent leurs mouvements pour percevoir les formes de petite taille. De plus ils développent plusieurs types de stratégies d'exploration. Plus tard, Ziat et Gapenne [178], examinent la reconnaissance d'ellipses de tailles diverses avec le même dispositif. Ils en ont déduit un seuil de confort basé sur le rapport entre la taille de la fenêtre et la taille de la forme, et que la stratégie d'exploration employée influe sur la performance selon la forme à reconnaître. Dans une autre étude [180], ils utilisent les mêmes formes mais cette fois-ci les différents groupes ne testent plus différentes tailles de figure, mais différentes tailles de fenêtre. Cette fois-ci ils constatent des ambiguïtés pour des rapports intermédiaires entre la taille de la fenêtre et la taille de la forme. Ces différentes études ont été complétées par une expérience où le but était de reconnaître une lettre englobée dans une forme géométrique [181]. Les résultats montrent que les utilisateurs arrivent mieux à identifier la forme que la lettre.

3.3.4. Les icônes

Icônes visuelles.

Les *icônes* sont des éléments essentiels des interfaces modernes. Les premières icônes visuelles sont apparues en même temps que les premières interfaces graphiques ou *GUI* (Graphical User Interface). Ces interfaces sont basées sur le modèle *WIMP* (Window, Icon, Menu, Pointing device : fenêtre, icône, menu, périphérique de pointage), et la première est issue du projet Xerox Star [36]. L'idée d'une icône est de synthétiser une information. Les icônes visuelles sont couramment utilisées pour représenter des applications ou des fichiers. Des études ont été menées pour créer des icônes utilisant d'autres modalités. Les motivations sont souvent les mêmes que toute application multimodale : le remplacement d'une modalité peut être souhaitable si l'environnement n'est pas propice à son utilisation (visuel dans l'obscurité, auditif en milieu bruité, etc.), ou si l'utilisateur possède des déficiences l'empêchant d'utiliser ces modalités.

Icônes sonores.

Les premières icônes non visuelles utilisaient la modalité sonore. Gaver a développé les *auditory icons* [63], qui sont des sons enregistrés. Par exemple un son de froissement de papier est joué pendant la destruction d'un fichier. Ce genre d'icône peut aussi être utilisé pour des notifications : réception d'un e-mail, téléchargement terminé, erreur système, etc.

Blattner *et al.* utilisent une autre approche : ils utilisent des mélodies pour représenter les informations de façon structurée [22]. Ces icônes appelées *earcons* utilisent des propriétés musicales pour définir plusieurs dimensions, chaque dimension codant une information. Brewster *et al.* ont examiné l'efficacité des earcons [28]. Ils utilisent le timbre, le rythme et le ton pour coder la famille (dessin,

texte, etc.), le type (fichier, dossier, application, ...), et un identifiant (pour différentier deux fichiers d'image par exemple).

Icônes tactiles.

L'idée des icônes est tout aussi transposable au sens auditif qu'au sens du toucher. Plusieurs systèmes d'icônes haptiques, que ce soit tactiles ou kinesthésiques, ont été développés.

Brewster *et al.* ont repris l'idée des earcons pour créer des icônes tactiles : les *tactons*[26, 25]. L'idée est de créer différentes sortes de vibrations sur des vibreurs tels que le Tactor VBW32 ou le Tactor C2 (figure 26), en combinant plusieurs paramètres : fréquence, amplitude, forme de signal, durée, rythme et position sur le corps. Une première étude a montré que des utilisateurs peuvent reconnaître efficacement des tactons construits à partir de trois « rugosités » (modulation d'amplitude) et trois rythmes [30], créant ainsi des tactons à 2 dimensions. Ces premiers résultats encourageants ont été complétés par d'autres études visant à trouver d'autres paramètres afin d'étendre les tactons à plus de 2 dimensions. Brown *et al.* [32] ont montré qu'il est possible d'utiliser des variations de nuances : crescendos (augmentation progressive de l'amplitude), decrescendos (diminution progressive de l'amplitude) et sforzandos (forte amplitude, puis passage brusque à une faible amplitude, puis crescendo). Trois paramètres ont été envisagés pour créer une troisième dimension : la fréquence, l'intensité et la localisation spatiale. La dernière a été sélectionnée pour être testée en combinaison avec les deux autres paramètres identifiés dans l'étude précédente (rugosité et rythme) [31]. Trois vibreurs ont été disposés sur le bras, créant ainsi trois valeurs pour chacune des trois dimensions. Le taux de reconnaissance est de 48% pour les icônes complètes, avec seulement 57% de reconnaissance du paramètre de rugosité. En réduisant le nombre de valeurs pour le paramètre de rugosité, le taux de reconnaissance monte à 80%, ce qui est acceptable.

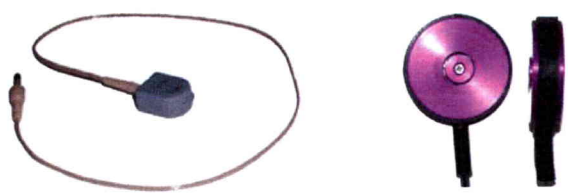


Figure 26 – Tactor VBW32 et C2

Evreinov *et al.* [56, 57] ont monté un moteur miniature dans un stylo pour permettre de ressentir des tactons. L'application proposée est un jeu où le but est de crocheter une serrure. L'avantage d'un tel système est qu'il est aussi bien accessible aux voyants qu'aux non-voyants, aux entendants et aux sourds.

D'autres systèmes de codage ont été créés pour représenter des indications pour des directions. Tan *et al.* ont formé une matrice de 3×3 vibreurs sur le dossier d'une chaise [162] (figure 27). Ils ont

créé deux jeux d'icônes, utilisant une illusion basée sur le principe de saltation. Il s'agit de disposer trois vibreurs à distance égale, et à donner trois impulsions sur le premier, puis trois sur le suivant et enfin trois sur le dernier. L'utilisateur a la sensation d'avoir ressenti neuf impulsions plus ou moins réparties équitablement entre le premier et le dernier vibreur. Cet effet permet de donner une plus grande résolution au niveau du ressenti, et ainsi d'indiquer des directions avec un nombre limité d'effecteurs. Les tests réalisés montrent que les utilisateurs arrivent à reconnaître entre 85% et 95% des directions avec le jeu d'icônes le plus efficace. Les auteurs suggèrent d'utiliser ce système en complément avec un GPS sur un siège d'automobile afin d'éviter au conducteur de quitter la route des yeux pour connaître son chemin.



Figure 27 – Le haptic back display

Des applications utilisant des icônes tactiles ont d'ores et déjà été imaginées. Le sens tactile est particulièrement intéressant pour transmettre des émotions. Rovers et van Essen ont créé un système de messagerie instantanée utilisant l'haptique pour transmettre des émotions [152]. Utiliser des icônes pour transmettre une émotion est devenu chose courante dans les systèmes de messagerie instantanée. Ces icônes, appelées émoticônes ou smileys, sont souvent visuelles. Certains systèmes ajoutent aussi du son. Elles permettent de décrire l'humeur du message textuel envoyé, et ainsi compenser le fait que l'interlocuteur ne voit pas les expressions faciales, ni le ton employé pour écrire le message. Ce système peut aussi être utilisé pour permettre aux correspondants de s'identifier secrètement en se transmettant un code basé sur des vibrations.

Benali-Khoudja *et al.* [17] ont créé des icônes tactiles pour représenter des émotions. Elles sont affichées sur le périphérique *VITAL* (Vibrotactile interface), qui possède une matrice de picots de 8×8. Le but est de créer un langage tactile composé d'icônes, telles que des lignes se déplaçant, des formes grossissant, etc. Les expériences conduites visaient à demander à des utilisateurs d'associer des émotions aux icônes. Dans une étude ultérieure [112], ils ont classé des icônes selon des descriptions données par des utilisateurs. Cette méthode va à l'encontre de la méthode des Tactons, suggérant de structurer l'information afin de définir l'espace des messages pouvant être transmis aux utilisateurs. Les auteurs évoquent la possibilité d'utiliser ces icônes pour créer des cartes géographiques tactiles.

Icônes kinesthésiques.

Les icônes utilisant le retour de force sont plus rares. Parmi elles on peut souligner les *Haptic Icons* d'Enriquez et MacLean [53]. Ces icônes utilisent un moteur à courant continu sur lequel est monté une poignée que l'utilisateur tient en main (figure 28). Quand le moteur tourne, l'utilisateur sent des forces dans un sens ou dans l'autre. Ce périphérique n'a donc qu'$1DOF$. Lors d'une expérience [102], des utilisateurs devaient classer des icônes construites sur le modèle des earcons et tactons, à partir de 3 formes de signal, 4 fréquences et 3 amplitudes. Les résultats montrent que les utilisateurs ont tendance à classer principalement les icônes par fréquence. Ils ont aussi tendance à séparer les icônes à forme sinusoïdale des icônes en carré et en dent de scie. Une autre expérience visait à examiner la possibilité d'associer une icône à un concept [54]. 3 fréquences ont été utilisées et étaient associées à des noms de plantes, ainsi que 3 formes de signal qui étaient associées à des noms de fruits. Après deux phases d'apprentissage, les utilisateurs ont réussi à faire l'association entre l'icône et le couple plante/fruit dans 76% des cas. Les désavantages de ces icônes sont d'une part qu'elles n'utilisent qu'un seul degré de liberté, ce qui enlève des paramètres possibles pour leur création et limite donc leur pouvoir expressif. D'autre part ces icônes utilisent un périphérique spécifique. Or comme elles utilisent un axe de rotation, les seuls périphériques courant permettant de les reproduire seraient les périphériques à $6DOF$.

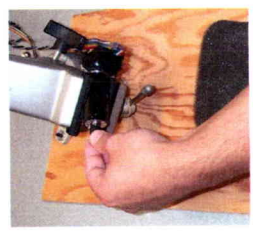


Figure 28 – Périphérique utilisé pour les haptic icons

Roberts et Franklin font une distinction entre icône et glyphe [150]. Les deux différences mentionnées sont que d'une part les icônes ne représentent qu'une unique représentation alors que les glyphes en représentent plusieurs. Or si on considère les earcons, tactons et haptic icons, on peut constater que plusieurs informations peuvent être codées dans la même icône. D'autre part ils avancent que la forme d'une icône est constante alors que celle des glyphes dépend des données qu'ils représentent. Or le pouvoir expressif des icônes ne réside pas dans une icône seule, mais dans l'ensemble des icônes possibles. Les auteurs introduisent le concept d'hlyphs, c'est-à-dire de glyphes haptiques. Un des exemples proposés montre comment schématiser les informations d'un graphique en créant des zones explorables avec un PHANToM. Leur taille et leur géométrie permet de coder entre autres les extremums, les points d'inflexion et les variations d'une courbe. Borst et Baiyya ont conçu de tels glyphes haptiques pour une matrice tactile de la taille de la paume de la main [24]. Ils proposent un exemple d'utilisation sur des cartes géographiques. Nous regrettons que l'aspect dynamique n'a été que peu exploré dans cette étude.

Icônes crossmodales.

Il est tentant de faire le parallèle entre les icônes vibrotactiles et les icônes auditives car les vibrations et les sons étant tous les deux de nature ondulatoire, il serait *a priori* possible d'utiliser les mêmes propriétés. Hoggan et Brewster [70] ont combiné les tactons et les earcons pour créer des *icônes crossmodales*. L'idée est d'avoir des jeux d'icônes tactiles et auditives qui peuvent encoder les mêmes informations. Il est aisé de faire le lien entre les paramètres : la fréquence, l'amplitude, la durée et le rythme sont utilisables en théorie pour les deux. La forme du signal tactile pourrait correspondre au timbre, et la position sur le corps pourrait correspondre à la position de l'instrument ou du haut parleur. Ce qui est moins évident c'est d'une part l'habileté des utilisateurs à distinguer les paramètres distinguables d'une modalité aussi facilement que ceux qui lui correspondent dans l'autre modalité. D'autre part les périphériques ne sont pas assez évolués pour permettre d'utiliser efficacement tous ces paramètres. Par exemple le timbre et le ton ont du mal à être rendues fidèlement sur un vibreur. Hoggan et Brewster ont expérimenté différentes positions sur le corps pour afficher des icônes crossmodales [71]. Il en résulte que la reconnaissance sur la cheville n'est pas bonne, et que le poignet n'est pas efficace en situation mobile. La meilleure solution retenue est la taille. Une étude plus récente visait à améliorer le paramètre de texture des tactons [73]. En effet, les études de Brown et Brewster [30] utilisaient la modulation d'amplitude pour créer de la rugosité. Hoggan et Brewster ont exploré trois paramètres différents : la modulation d'amplitude, la fréquence et la forme de signal. La forme de signal est le paramètre le mieux reconnu avec 94% de reconnaissance, contre 81% pour la fréquence et seulement 61% pour la modulation d'amplitude. La rugosité peut donc être utilisée comme dimension de tacton, en correspondance avec le timbre des earcons. Il est donc possible de construire des tactons et des earcons utilisant les 3 paramètres : rythme, rugosité et position spatiale. Hoggan et Brewster ont expérimenté la reconnaissance d'icônes crossmodales en situation stationnaire et mobile [72]. Deux cas ont été examinés : dans le premier les utilisateurs étaient entraînés avec des earcons et le test consistait à reconnaître les tactons correspondants. Dans ce cas le taux de reconnaissance était de 85% en situation stationnaire et 76% en situation mobile. Dans le deuxième c'est l'inverse : les utilisateurs étaient entraînés avec des tactons et devaient reconnaître des earcons. Les utilisateurs reconnaissaient les icônes avec un taux de réussite de 76% en situation stationnaire et 71% en situation mobile.

3.3.5. Conclusion

Les interactions haptiques ont déjà trouvé ces dernières années quelques applications. Elles sont utilisées intensivement dans des simulations, et particulièrement dans le milieu médical. Les systèmes présentés servent d'une part d'entraînement, ce qui a un intérêt évident en terme d'apprentissage avant de passer à de vrais patients. D'autre part il y a les systèmes de téléopération qui permet de faire des manipulations à distance. L'haptique est aussi utilisée pour augmenter le réalisme et améliorer l'efficacité d'interfaces. Différents systèmes utilisant des vibrations pour faire ressentir les composants d'une interface ont été présentés. Les études citées montrent un accroissement de l'efficacité en présence d'interaction haptique. La possibilité de ressentir un environnement en 3D, et notamment à travers un guidage permet de mieux comprendre cet environnement. Ceci a motivé la conception d'un système de guidage, que nous utiliserons dans une application multi-

modale dans le chapitre IV. Enfin à l'instar des modalités visuelles et auditives, l'haptique a aussi été utilisée pour coder de l'information. Nous avons aussi bien présenté des icônes tactiles que des icônes kinesthésiques. Nous pouvons observer deux approches pour la création d'icônes : soit elles fonctionnent par analogie avec quelque chose de connu par l'utilisateur (auditory icons, icônes de Benali-Khoudja), soit elles sont construites à partir de paramètres abstraits et le sens est donné par association (earcons, tactons, haptic icons). L'avantage de la seconde est qu'il est possible de transposer une association d'une technique d'interaction à l'autre et ainsi de créer des icônes crossmodales. Ceci permet de choisir la modalité la plus appropriée selon la situation.

3.4. Conclusion

Parmi les cinq sens du corps humain, nous avons décidé d'utiliser l'haptique pour afficher des informations. Comme nous avons pu en discuter, ce sens est ressenti par les mécanorécepteurs de la peau, les muscles, les tendons et les articulations. Dans les systèmes informatisés, ces sens sont stimulés par des périphériques haptiques. La diversité des sensations haptiques fait que chaque périphérique est spécialisé dans une sensation particulière. Il est courant de distinguer les périphériques tactiles qui stimulent principalement les mécanorécepteurs de la peau, et les périphériques à retour de force qui stimulent principalement les mécanorécepteurs des muscles, des tendons et des articulations. Parmi les périphériques tactiles nous avons évoqué quelques périphériques à matrices de picots, des vibreurs, des appareils à charge électrique et les appareils à pression. Parmi les périphériques à retour de force nous avons présenté en premier lieu des contrôleurs de jeux, car ce sont des périphériques couramment utilisés. Ensuite nous avons présenté les bras articulés, les pantographes, les systèmes pneumatiques et les exosquelettes. Au niveau logiciel, les applications utilisant ces périphériques sont d'ores et déjà nombreuses. D'une part il y a les systèmes de simulations et les systèmes d'exploration d'interface. D'autre part il y a les techniques d'interaction telles que celles que nous allons décrire dans les chapitres suivants, à savoir les systèmes d'exploration d'un environnement virtuel, et le codage d'information en utilisant l'haptique. Nous allons maintenant nous intéresser à la perception d'objets virtuels à travers des interfaces haptiques et comment cela peut aider des utilisateurs déficients visuels.

4. Les systèmes pour déficients visuels

Les interfaces modernes sont pensées pour être visuelles, et d'ailleurs on utilise souvent le terme de Graphical User Interface (GUI). Depuis l'invention de l'interface graphique [36], la souris est devenue le périphérique d'interaction principal et l'environnement de travail est devenu une métaphore du bureau physique. Alors que le texte se traduit facilement tactilement avec le Braille, tout ce qui est graphique est plus difficile à traduire. Typiquement les personnes non voyantes utilisent une synthèse vocale et une plage Braille (figure 29) pour les informations textuelles. Les graphiques sont soit ignorés soit remplacés par des textes alternatifs.

Nous allons à prime abord évoquer quelques études et recommandations sur les méthodes de présentation et d'exploration de systèmes à retour haptique. Nous verrons à la fois des exemples pour du retour de force et pour du retour tactile. Ensuite nous examinerons des études présentant



Figure 29 – Plage Braille

des méthodes d'exploration non visuelles utilisant l'haptique. Ceci inclut l'exploration d'objets, de graphiques, d'interfaces et de scènes 3D. Enfin nous aborderons le problème de l'utilisation de l'haptique pour les enfants déficients visuels, dans le domaine de l'enseignement.

4.1. Méthodes de présentation et d'exploration

Contrairement aux interfaces visuelles, l'exploration d'un environnement haptique n'est pas aisée. L'absence de vision prive l'utilisateur d'une vue d'ensemble, ce qui pose en particulier de sérieux problèmes d'orientation et de compréhension des objets de la scène.

Bras articulés.

En ce qui concerne le retour de force, Sjoström [158, 159] a étudié ce genre de problèmes en détails et a abouti à des règles de conception qui visent à favoriser la conception d'environnements haptiques accessibles. Il suggère par exemple de toujours fournir un point de référence dans la scène afin que l'utilisateur puisse se repérer par rapport à ce point. Il recommande aussi d'éviter de supprimer des éléments temporairement inutilisés dans la scène, mais de désactiver leurs fonctionnalités si c'est possible. Pour aider l'utilisateur à localiser un objet, il peut être utile de l'aimanter afin qu'il soit guidé dessus lorsqu'il passe à proximité. Ceci peut aussi être utile pour aider l'utilisateur à parcourir l'objet, et en particulier lors d'une exploration avec un périphérique disposant d'un seul point de contact comme le PHANToM. Dans le cas d'un objet qui peut ne pas exister, il est recommandé de créer un chemin conduisant à l'endoit où l'objet est censé se trouver. Ainsi l'utilisateur peut savoir si l'objet est présent ou non. De manière générale, il faut donner à l'utilisateur la possibilité de chercher efficacement les objets de la scène. Il précise aussi que les formes sont plus faciles à reconnaître lorsqu'elles sont parcourues de l'intérieur que de l'extérieur. En particulier, vu de l'extérieur, les angles sont perçus différemment. Les systèmes de widgets haptiques doivent attirer le curseur vers le centre du widget pour éviter que l'utilisateur glisse dessus. Enfin lors de la création d'une interface, il est important de savoir quel système sera manipulé pour explorer, car les forces à appliquer en dépendent.

Matrices de picots.

Les matrices de picots sont utilisées autant pour afficher du texte, avec l'alphabet Braille et ses dérivés pour la notation musicale [52], les mathématiques [8], etc., que pour afficher des images. Il existe plusieurs techniques pour transcrire une image visuelle en image tactile. La traduction tactile

d'images visuelles sur une matrice de picots est aisée : il suffit de faire un seuillage sur l'image, et de lever les picots correspondant aux pixels sombres et d'abaisser ceux correspondant aux pixels clairs. Cependant quelle que soit la représentation tactile, il est souvent nécessaire de simplifier et annoter l'image [55]. Certains concepts comme la perspective perturbent la reconnaissance au lieu d'apporter de l'information comme c'est le cas pour les images visuelles. De plus, comme dans le cas des périphériques à retour de force, les périphériques tactiles ont leurs limites en ce qui concerne la fidélité de la représentation.

4.2. Exploration non visuelle

Il existe de nombreuses tâches habituellement effectuées grâce à la vision. De manière générale il s'agit de l'exploration de l'interface ou de l'environnement virtuel et des objets qui le composent. Nous allons commencer par aborder des études sur l'exploration d'objets, puis nous allons examiner l'étude de différents types de graphiques. Ensuite nous étudierons quelques systèmes permettant d'explorer des interfaces sans la vision. Enfin nous nous intéresserons à l'exploration de scènes de manière non visuelle.

4.2.1. Objets

La perception par le toucher d'objets réels et d'objets virtuels sont deux choses distinctes. Dans le premier cas l'objet transmet directement ses propriétés tactiles et kinesthésiques : rugosité, température, forme, poids, etc. Dans le second cas ces propriétés sont transmises par des périphériques haptiques. La performance de perception peut être affectée par la fidélité du rendu des effets haptiques rendus. Le principal facteur limitant des bras articulés dans ce domaine est le nombre limité de points de contacts. C'est pour cette raison qu'il est important d'étudier les limites des périphériques au niveau de la perception des effets rendus.

Jansson [78] s'est intéressé à l'identification de formes géométriques 3D (cylindre, cône, cube et sphère) de différentes tailles ($5mm$, $25mm$ et $50mm$) avec un PHANToM Premium 1.5. Les tests ont été réalisés avec des utilisateurs voyants. La sphère est toujours reconnue correctement, et avec les temps d'exploration les plus faibles. Les formes les plus difficiles à identifier sont les cylindres et les cônes de $5mm$ avec environ 70% de reconnaissance, contre plus de 90% pour les autres formes. Une étude ultérieure [81] a été réalisée avec des utilisateurs déficients visuels, et des formes de taille allant de $6mm$ à $100mm$. 6 des 10 utilisateurs ont eu des résultats similaires à ceux des utilisateurs voyants, et les autres ont fait beaucoup d'erreurs et n'ont pas toujours fini les tests. Cependant les auteurs signalent que les utilisateurs ayant eu des difficulté ont considérablement amélioré leurs résultats lors d'expériences ultérieures après un apprentissage adéquat [80].

Deux autres études comparent l'identification des mêmes formes mais avec des tailles différentes. Ces études comparent l'exploration d'objets réels manipulés avec les doigts et d'objets virtuels manipulés avec un PHANToM, soit avec un stylet soit avec un dé à coudre. Dans la première étude [79] les tailles sont de $5mm$, $7mm$ et $9mm$. La perception des objets réels avec les doigts conduit à une reconnaissance parfaite et immédiate de toutes les formes. Les deux types d'explorations avec le

PHANToM entraînent significativement plus d'erreurs de reconnaissance (entre 67% et 80% de réponses correctes selon la taille), sans qu'il y ait de différence significative entre les deux méthodes d'exploration avec le PHANToM. On constate le même phénomène avec les temps d'exploration qui se situent entre $27s$ et $37s$ selon la taille de l'objet. Dans la seconde étude [80], Jansson *et al.* utilisent des tailles différentes : $10mm$, $50mm$ et $100mm$. Cette fois les utilisateurs n'ont pas parcouru d'objets réels, mais uniquement des objets virtuels avec un PHANToM. Les utilisateurs ont toujours reconnu la sphère, quelle que soit sa taille. De manière globale la grande majorité des erreurs ont été commises la première fois qu'une forme était présentée. Ainsi si on ne compte pas la première présentation de chaque forme, les utilisateurs obtiennent 95% de bonnes réponses sur toutes les formes avec toutes les tailles. Les plus mauvais résultats sont obtenus avec les cylindres et les cônes de $10mm$ avec une moyenne d'environ 70% de bonnes réponses, tous les autres résultats étant au dessus de 90%.

Plus tard [82], Jansson et Ivås se sont intéressés à l'impact d'un apprentissage sur la reconnaissance de ces objets en 3D. Six tailles ont été utilisées (entre $5mm$ et $15mm$), et les utilisateurs se sont livrés à 9 sessions de tests. Les résultats révèlent que les utilisateurs se divisent en deux groupes : une minorité d'utilisateurs (3/10) restent à environ 30% de réponses correctes, avec un temps d'exploration avoisinant les $25s$. Par contre une majorité d'utilisateurs (7/10) a doublé son pourcentage de réponses correctes : de 30% à 60%, et a réduit son temps d'exploration de $45s$ à moins de $30s$. Les auteurs ont identifié deux manières de tenir le stylet du PHANToM, cependant ils ne relèvent aucune différence de résultats par rapport à ce paramètre.

Par la suite, Jansson et Larsson [83] ont poussé ces expérimentations à l'exploration d'un visage avec un PHANToM. Le visage est défini par des yeux, des oreilles, des sourcils, un nez, une bouche et un menton, construits à partir de formes basiques (figure 30). Le but de l'utilisateur est d'identifier la partie dont la taille diffère par rapport à un visage témoin dont les parties sont définies comme étant de taille normale. 6 degrés de difficulté sont testés, correspondant au rapport entre la taille normale et la taille de l'objet différent. Les utilisateurs étaient divisés en deux groupes : des étudiants en psychologie, et des déficients visuels. On ne constate aucune différence significative d'identification entre les deux groupes d'utilisateurs. Cependant la performance est meilleure avec un faible niveau de complexité (80% de réponses correctes) qu'avec un niveau de difficulté élevé (40% de réponses correctes). Il y a une différence significative de reconnaissance selon la partie explorée, sans qu'il y ait toutefois de différences entre parties de petite taille et entre parties de grande taille. Le temps d'exploration est significativement plus court de $10s$ en moyenne pour les déficients visuels que pour les autres utilisateurs. La progression par rapport à la difficulté est aussi significative : $50s/60s$ pour les tâches faciles et $60s/80s$ pour les tâches plus difficiles. Cette étude montre bien la difficulté de reconnaissance d'objets avec un PHANToM, surtout lorsque les différences sont minimes.

4.2.2. Graphiques

Les graphiques sont des éléments importants dans une grande variété de documents : scientifiques, scolaires, financiers, etc. C'est pourquoi des équipes de recherche se sont penchés sur le



Figure 30 – Visage construit à partir de formes de base

problème de l'accessibilité des graphiques aux personnes déficientes visuelles. Nous allons aborder trois types de graphiques ayant des propriétés distinctes. Dans un premier temps nous étudierons les graphes, qui représentent des séries de données numériques sous plusieurs formes : courbes, histogrammes, camemberts, etc. Ensuite nous nous intéresserons aux cartes, dont les éléments principaux sont les frontières, les territoires, les zones de mer, etc. Les contours sont plus irréguliers que ceux des graphes. Enfin nous examinerons les schémas. Ceux-ci peuvent représenter des circuits électriques, des diagrammes UML, etc.

Graphes.

Les graphes représentent des ensembles de valeurs sous de multiples formes. Ils permettent à des yeux exercés d'extraire rapidement l'information contenu dans ces ensembles de valeurs. Étant donné leur nature visuelle, les utilisateurs déficients visuels sont souvent privés de ce merveilleux outil. Des recherches ont été conduites pour représenter des graphes à l'aide du retour sonore [87], ainsi qu'à l'aide du retour haptique. Nous allons détailler ces dernières, en dissociant celles utilisant le retour de force et celles utilisant le retour tactile.

Bras articulés.

Yu *et al.* [175] ont essayé d'utiliser des propriétés haptiques pour pouvoir ressentir l'inclinaison d'une courbe. Leur système affiche une courbe constituée de lignes sous forme de cylindres, à moitié enfoncé dans un plan vertical. L'utilisateur pouvait choisir d'afficher une grille ou non pour s'aider à réaliser les tâches qui lui étaient demandées. 15 utilisateurs dont 5 aveugles ont participé aux tests. L'expérience était divisée en deux phases. Dans la première, deux lignes parallèles étaient présentées : une glissante et une collante. Le but était de déterminer laquelle était glissante et laquelle était collante. Seuls 2 utilisateurs se sont trompés, ce qui donne un résultat de 93% de réponses correctes. La seconde tâche consistait à trouver les coordonnées des points maximum et minimum des courbes, ainsi que déterminer l'endroit où elles se coupent. Au vu des résultats, les auteurs ne peuvent pas conclure sur l'utilité de la grille pour accomplir les tâches. Le problème d'utilisation majeur de la grille est que lorsque la courbe coupe une ligne de la grille, celle-ci n'est pas ressentie, et donc ne peut pas être comptée. En examinant les trajectoires des utilisateurs, les auteurs remarquent qu'il y a des problèmes d'exploration aux angles.

Pour contourner ce problème, les auteurs proposent de graver la courbe sur le fond à la place de la faire ressortir [177]. Dans cette nouvelle expérimentation, deux groupes de graphes étaient considérés. Dans le premier groupe, les graphes avaient jusqu'à 3 courbes qui ne se croisent pas, et

dans le deuxième il y avait deux courbes qui se croisent. Les instructions des utilisateurs étaient de compter le nombre de courbes, d'identifier la texture de la courbe, de compter le nombre d'inflexions sur les courbes, et de compter le nombre d'intersections. Les auteurs on mesuré un taux de 85,56% de bonnes réponses pour les courbes du premier groupe et 94,07% pour les courbes du second, la différence étant significative. La tâche la plus facile était l'identification de la texture des courbes.

Cette représentation a été utilisée dans une étude ultérieure [176]. Le but cette fois-ci est de parcourir des histogrammes. Le rendu est multimodal car il utilise le retour de force et le retour sonore. Le retour de force est soit réalisé avec un PHANToM, tel que décrit ci-dessus, soit il est réalisé avec une souris Wingman Force Feedback. Le rendu sonore quant à lui utilise une synthèse vocale pour prononcer le numéro de la barre explorée ainsi que sa valeur lorsque l'utilisateur clique, et un rendu musical qui joue une note MIDI dont la hauteur correspond à la valeur de la barre (aigu pour haut et grave pour bas). Le but est de comparer le PHANToM et la souris à retour de force, ainsi que les modalités audio et haptique. Quatre questions étaient posées : décrire l'allure générale des données, trouver le maximum, trouver le minimum, et trouver les valeurs proches. Les tests avec la souris à retour de force ont été réalisés dans deux conditions : audio uniquement, et audio + retour de force. Les résultats montrent que la deuxième condition permet de répondre significativement plus facilement aux questions (82,81% et 88,59%). Les erreurs se situent principalement sur la quatrième question, et en particulier dans la condition auditive. Le temps de réponse n'est pas différent d'une condition à l'autre. Les tests avec le PHANToM ont été eux réalisés dans deux autres conditions : retour de force, et audio + retour de force. Les résultats ne permettent pas de dégager une différence significative entre les deux conditions en terme de réponses correctes (85,78% et 89,22%). La dernière question reste la plus problématique. Par contre la condition retour de force permet aux utilisateurs de répondre avec $25s$ de moins que la condition audio + retour de force.

Wall et Brewster se sont intéressés à la perception de certaines propriétés haptiques avec un PHANToM. Une première étude [166, 167] examine la reconnaissance de frottement, de texture et de dureté. Les utilisateurs devaient identifier la surface qui différait parmi trois présenter où une seule variait sur un des paramètres. La reconnaissance du frottement est significativement meilleure que celle des deux autres paramètres. Le taux de reconnaissance varie entre 50% et 90% selon la différence entre les deux textures montrées pour le frottement. Le taux pour la texture varie entre 37% et 85% et celui de la dureté varie entre 45% et 85%. Ces résultats ont poussé les auteurs à expérimenter une application d'exploration d'histogrammes à l'aide de la friction [168]. 12 utilisateurs aveugles devaient utiliser trois méthodes pour explorer des histogrammes. La première méthode consistait à représenter la valeur des barres grâce à la hauteur de la barre. La deuxième méthode utilise la valeur de frottement pour représenter la valeur. Enfin la troisième méthode utilise à la fois la hauteur et la valeur de frottement. Les résultats montrent qu'en dépit des bons résultats de l'expérience précédente, les utilisateurs ont significativement moins bien reconnu les graphiques avec le frottement qu'avec les deux autres méthodes. Les deux autres méthodes n'étaient pas statistiquement différentes. Le frottement n'est donc pas une méthode efficace pour représenter des histogrammes.

Un système de balises (*beacons*) a été proposé par Wall et Brewster afin de pouvoir explorer des histogrammes plus efficacement [169]. Ces balises sont des marques que l'utilisateur peut poser à l'aide du clavier. Quand il veut revenir à une marque il presse une autre touche du clavier, et le PHANToM guide l'utilisateur vers la marque. Le système d'affichage d'histogrammes est le même que dans les études précédentes de Yu *et al.*. 8 utilisateurs déficients visuels ont participé aux tests. Ils devaient repérer les minimums ou maximums parmi un sous-ensemble des barres sélectionnées par l'expérimentateur. L'expérience se déroulait dans deux conditions : avec et sans les aides haptiques. Les utilisateurs ont donné 76% de bonnes réponses, et nécessitaient $50{,}92s$ pour répondre, ce qui montre qu'ils ont éprouvé des difficultés à répondre aux questions. Les utilisateurs ont reconnu que l'aide était potentiellement utile, mais qu'elle était difficile à utiliser. L'utilisation du clavier en complément du PHANToM est jugée difficile. Certains utilisateurs suggèreraient d'utiliser ce système sur des graphes plus compliqués afin de mieux exploiter le potentiel de cette aide.

Matrices de picots.

Wall et Brewster ont étudié le rendu d'autres types de graphes : des camemberts [171]. Cette fois, ils utilisent une matrice tactile dans un système appelé Tac-tiles (figure 31). Les deux matrices de la souris VTPlayer sont utilisées comme afficheur. L'exploration se fait à l'aide d'une tablette graphique, sur laquelle sont empilés deux disques compact (de $6cm$ et $4cm$ de rayon), afin de réaliser un retour haptique du diagramme sous forme d'interface tangible. Le retour tactile consiste en une traduction du visuel au tactile des pixels autour du curseur. Les interactions sont complétées par deux types d'informations sonores : un clic déclenche une synthèse vocale qui prévient l'utilisateur qu'il est en dehors du diagramme, ou qui décrit les données de la portion courante du camembert. Une note MIDI, dont la hauteur est proportionnelle à la valeur de la section, est jouée lorsque l'utilisateur change de portion. Des utilisateurs déficients visuels ont participé aux tests : ils devaient trouver la plus petite et la plus grande section de chaque camembert présenté. 89% des réponses étaient correctes. Parmi les 9 erreurs, 4 sont dues à une erreur de perception de la synthèse vocale, 3 sont dues à des valeurs très proches, et 2 sont dues au fait qu'un utilisateur a inversé la correspondance des notes du retour sonore. Les utilisateurs ont tendance à utiliser les disques pour parcourir efficacement les camemberts, en utilisant le plus grand pour comparer les petites sections, et le plus petit pour comparer les grandes.



Figure 31 – Exploration de camemberts avec Tac-tiles

Wall et Brewster se sont intéressés à la représentation haptique de graphiques. Ils ont comparé la perception de lignes à différentes valeurs de pente [170]. L'expérience vise à comparer trois méthodes de représentation : sur papier en relief, avec une souris à retour de force (Logitech Wingman Force Feedback), et avec une souris à matrice de picots (VTPlayer). Les résultats montrent que la représentation sur papier en relief permet de détecter des pentes à partir de 2,42° par rapport à l'horizontal, alors que la représentation avec la souris à retour de force n'est reconnaissable qu'à partir de 3,25° de différence et la représentation avec la matrice de picots nécessite 4,7°. La supériorité du papier en relief s'explique par le fait qu'elle combine les informations tactiles et les informations kinesthésiques, et ce avec vingt doigts répartis sur deux mains. Le retard de la matrice de picots est dû au fait que la matrice de picots de la VTPlayer a une faible résolution, et qu'elle n'exploite pas tous les récepteurs sensitifs de la peau.

Cartes.

Les problèmes d'affichage d'informations essentiellement graphiques s'étend au delà du domaine des graphes. Les cartes géographiques constituent un autre problème de représentation majeur. Divers problèmes sont liés à l'exploration de cartes géographiques : localisation des territoires, compréhension des données, etc. L'exploration de cartes de manière non visuelle est très difficile. Des études montrent comment explorer des cartes avec un rendu sonore [48], mais nous allons observer comment on peut les explorer de manière tactile.

Jansson et Pedersen ont expérimenté l'exploration de cartes géographiques avec une souris VTPlayer [84]. Elle permet, grâce à un logiciel fourni, de ressentir une transcription du visuel au tactile de la carte des USA : les frontières font 2 pixels de large, et deux textures différentes différencient les zones de terre et les zones de mer. De plus une synthèse vocale prononce le nom de l'état lorsque l'utilisateur passe une frontière. L'utilisateur peut aussi s'aider d'une aide kinesthésique, correspondant à la position de sa main sur la VTPlayer. Un groupe d'utilisateurs voyants et un autre groupe d'utilisateurs déficients visuels avaient pour tâches de trouver des séquences d'états. Chaque groupe était divisé en deux : une partie avait accès au retour visuel, et l'autre non. Les résultats montrent que les performances avec et sans retour tactile ne sont pas distincts. Les auteurs soulignent aussi des problèmes liés à l'utilisation de la VTPlayer : le curseur n'avance pas de la même façon si la souris est tenue perpendiculairement au mouvement ou si elle est tenue de manière oblique. Ce problème vient du fait que le capteur optique se situe à l'arrière de la souris. Le système de synthèse vocale ne prononce le nom de l'état qu'à une certaine distance de la frontière, ce qui fait que l'utilisateur peut parcourir de longues distances sans avoir d'informations utiles. Ces zones sans informations troublent les utilisateurs.

Schémas.

Gallagher et Frasch ont développé TACIS (figure 32), un système utilisant l'haptique et le son pour explorer des schémas [61]. Le système utilise une base de données pour générer un graphique imprimé sur du papier en relief. La feuille est placée sur un touchpad de grande taille pour permettre au système de donner un retour sonore correspondant à l'exploration de l'utilisateur en plus du

retour tactile réalisé par le papier en relief. Les auteurs présentent trois exemples. Le premier est un piano : le système joue la note correspondant à la touche sur laquelle l'utilisateur a posé le doigt. Si l'utilisateur presse plus fort, le système prononce le nom de la note. Le deuxième exemple est une carte de la ville de Dublin. Les rues et la forme des bâtiments sont imprimées sur la feuille. Un son différent est joué selon que l'utilisateur a le doigt sur la rue ou sur un arrêt de bus par exemple. En pressant plus fort, l'utilisateur peut recevoir plus d'informations sur l'endroit qu'il explore. Sur une rue le système prononcera son nom, sur un arrêt de bus le système prononcera les horaires et les lignes circulant. Enfin le troisième exemple permet de traduire une image : les couleurs sont rendues sous forme de notes. Un utilisateur voyant peut ajouter des annotations qui seront prononcées par une synthèse vocale si l'utilisateur presse à cet endroit là. Les informations haptiques données par ce système sont données par le schéma sur papier en relief, ce qui fait qu'il en hérite les inconvénients (aspect statique, coût, etc.).



Figure 32 – Le système Tacis

Le système *TeDub* propose une autre approche [75]. Le but de ce système est de permettre à un utilisateur aveugle d'explorer un document essentiellement graphique : schéma électrique, diagramme UML, plans, etc. Le système est constitué de deux outils. Le premier, appelé *diagramInterpreter*, permet d'interpréter les diagrammes pour en extraire les informations à afficher. Il prend en entrée un fichier XML, un dessin vectoriel ou une image bitmap pour la transformer dans un format générique. Cet outil utilise des hypothèses pour extraire les informations. Ainsi il est possible d'extraire des informations dépendant du domaine lié au contenu du document. Par exemple les éléments des diagrammes UML possèdent une certaine géométrie, et les liens entre les objets sont normalisés. Par ailleurs un utilisateur voyant peut utiliser cet outil pour annoter le schéma avec des informations supplémentaires afin d'aider les utilisateurs mal voyants qui vont parcourir ces diagrammes. Le deuxième outil (*diagramNavigator*) est un navigateur permettant d'explorer les diagrammes génériques générés par le premier outil. L'utilisateur peut parcourir les diagrammes de manière hiérarchique avec un clavier. Des earcons sont utilisés pour donner des informations sur la position dans la hiérarchie (feuille, fin de liste, etc.). Il peut aussi revenir en arrière dans sa navigation, à la manière d'un navigateur web. L'élément courant peut être exploré à l'aide d'un joystick à retour de force. Quand l'utilisateur passe à proximité d'un élément voisin son nom est prononcé. Le son 3D est aussi utilisé pour suppléer le rendu haptique et textuel. Le retour de force du Joystick est utilisé

pour ressentir les objets de la scène, comme les murs sur un plan architectural. L'interface elle même est conçue de manière générique et avec des outils standards, de telle manière que les utilisateurs peuvent se servir de leur systèmes de synthèse vocale et d'affichage Braille usuels.

4.2.3. Interfaces

Ramstein *et al.* ont développé *PC-Access* [145], un système utilisant le retour sonore et le retour de force afin d'aider les utilisateurs non-voyants à explorer les interfaces graphiques. La navigation se fait soit avec une tablette graphique, soit avec un pantographe. L'avantage par rapport à une souris est que la position est absolue, et donc l'utilisateur peut la connaître par proprioception. Tous les objets et actions sont accompagnés d'un rendu sonore, ainsi l'utilisateur est toujours averti de ce qui se passe.



Figure 33 – Le Moose

O'Modhrain et Gillepsie [130] ont développé un autre système appelé le Moose (figure 33), permettant d'explorer une interface avec une souris et d'en ressentir les éléments grâce à un retour de force. Le périphérique est un carré de $33cm$ de côte sur lequel l'utilisateur déplace une souris. Dans un coin l'utilisateur dispose d'un palet à $2DOF$ pouvant se déplacer sur carré de $3cm$ de côté, et disposant d'un retour de force de $6N$ maximum grâce à des moteurs à courant continu. Les auteurs ont créé des icônes haptiques de manière empiriques afin de représenter les icônes sur le bureau, les cases à cocher, les boutons, etc. Ils mentionnent de potentiels travaux futurs dans ce domaine, cependant aucune étude de la sorte n'a été publiée par la suite.

4.2.4. Scènes 3D

Lecuyer *et al.* [93] ont créé un environnement de réalité appelé HOMERE (figure 34) destiné à des utilisateurs aveugles pour les aider à apprendre à explorer un environnement inconnu. Le système présenté permet d'explorer virtuellement la Cité des Sciences de La Vilette à Paris. Le système est consitué d'une partie de canne blanche attachée à un Virtuose 3D, permettant de ressentir une représentation haptique de l'environnement à explorer. Deux autres retours ont été ajoutés : un retour thermal avec un soleil virtuel jouant un rôle de boussole lorsque l'utilisateur est à l'extérieur des bâtiments dans l'environnement virtuel. Le retour sonore fournit plusieurs informations : un univers ambiant, les bruits de pas du guide que l'utilisateur suit passivement, et enfin les commentaires

du guide. L'utilisateur utilise la canne à la fois pour explorer son entourage, et pour être guidé sur un chemin prédéfini. Il s'est avéré que cette double facette active/passive sur la même modalité a perturbé certains utilisateurs.



Figure 34 – Le système Homere

Crossan et Brewster [44] reprennent l'idée d'interaction bi-manuelle utilisée dans certaines études ci-dessus. Le but est de parcourir un labyrinthe avec des informations haptiques. L'utilisateur parcourt le labyrinthe, dans lequel il est contraint, avec un PHANToM OMNI. Afin de l'aider aux intersections, l'utilisateur reçoit une transcription de la zone autour du curseur sur les matrices tactiles d'une souris VTPlayer. Après quelques préliminaires avec 4 utilisateurs aveugles [45], cette solution a été abandonnée. En effet selon les utilisateurs la charge cognitive est trop importante, et l'utilisation de cette technique demande trop d'efforts. Cette aide tactile a été remplacée par des tactons directionnels, que nous décrivons en détails dans le chapitre II. Deux jeux de tactons représentant quatre directions ont été développés : des tactons statiques (le motif ne change pas), et des tactons dynamiques (le motif se déplace au cours du temps). Durant l'exploration, le système affiche sur la souris VTPlayer un tacton représentant la direction à suivre pour aller vers la sortie. Dix utilisateurs avaient pour but de sortir de 8 labyrinthes avec des indications statiques, et de sortir de 8 autres avec des indications dynamiques. Chaque tentative devait se faire le plus rapidement possible, et en moins de $50s$, sans quoi la tentative était considérée comme un échec. Les résultats permettent de conclure que les utilisateurs arrivent à terminer plus de labyrinthes, et parcourent plus vite les labyrinthes avec des indications statiques qu'avec des indications dynamiques. On peut aussi souligner que les utilisateurs avouent préférer les tactons statiques aux tactons dynamiques.

4.3. Systèmes haptiques pour l'enseignement

D'ordinaire la vision est un élément crucial dans l'éducation. Les cours sont écrits sur un tableau, que les élèves recopient sur un cahier. Ils sont accompagnés de nombreux graphiques, schémas, diagrammes et cartes sur papier. Il a fallu trouver des substituts pour les enfants qui ne peuvent pas s'appuyer sur leur vision. Ainsi l'alphabet Braille (figure 35) a été inventé en 1929 par Louis Braille pour représenter les caractères alphanumériques, ainsi que certains signes de ponctuation. Ainsi les documents textuels sont facilement traduisible en une forme lisible par une personne non voyante.

À titre de comparaison, un voyant lit du texte à une vitesse moyenne de 300 mots par minutes alors qu'un expert en lecture braille n'atteint que 100 mots par minute en moyenne [97]. Le passage de la version imprimée à la version Braille induit donc un coût en terme d'efficacité. Des extensions ont été inventées pour représenter les expressions mathématiques, cependant il n'existe pas de standard unanimement admis et chaque constructeur de matériel tente d'imposer son alphabet.

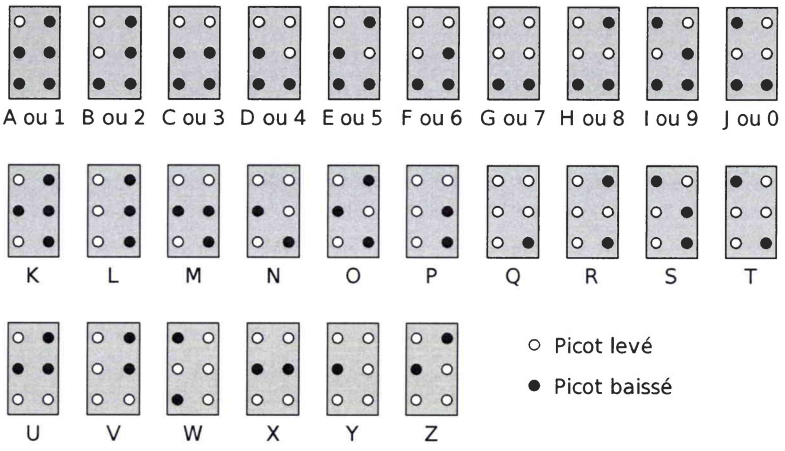


Figure 35 – L'alphabet Braille

Pour les schémas, les élèves ayant une vision résiduelle peuvent utiliser des schémas agrandis et contrastés. Par contre les non voyants doivent recourir à des schémas sur papier en relief. Cette méthode possède de nombreux désavantages. D'une part la préparation des supports prend du temps. D'autre part il n'est pas possible d'effacer, donc en cas d'erreur il faut recommencer le tracé. De plus il n'est pas toujours évident d'ajouter du texte car le texte Braille prend beaucoup de place en comparaison avec des lettres imprimées. Des systèmes informatisés ont donc été conçus afin de contourner ces problèmes.

4.3.1. MICOLE

Le projet européen MICOLE[1] [113] avait pour objectif d'étudier cette problématique. Concrètement ce projet a permis d'établir des recommandations sur la conception d'interfaces destinées aux enfants déficients visuels [104], et en particulier ce qui concerne la collaboration [156]. Les études que nous avons menées dans ce domaine pour les besoins de ce projet sont détaillées dans les chapitres II et III. Ce projet a aussi abouti une architecture multimodale qui permet de faciliter le développement d'applications multimodales et collaboratives pour enfants déficients visuels. Cette architecture sera décrite en détails dans le chapitre IV. Enfin plusieurs applications ont été réalisées pour ce projet, dont une application d'exploration de schémas de circuits électriques décrite dans le chapitre IV et quelques autres applications décrites ci-dessous.

[1] Multimodal Collaboration Environment for Inclusion of Visually Impaired Children

La géométrie est un domaine très difficile à enseigner à des enfants déficients visuels. D'ordinaire ils apprennent la géométrie à l'aide d'outils ayant des graduations en relief. Cependant cela n'aide pas à avoir une vue d'ensemble d'une construction géométrique, et d'identifier les formes dessinées. Zijp-Rouzier et Petit [182] ont créé un système appelé SALOME permettant de créer des constructions géométriques grâce à un bras articulé à retour de force. Les formes sont dessinées sur un plan horizontal grâce à des rainures. Un système de synthèse vocale a été intégrée afin de prononcer le type de forme exploré. Des encoches sont placées à des intervalles réguliers sur les segments et les courbes pour permettre à l'utilisateur de mesurer leur longueur. Le système permet de construire et de modifier des formes de base, ainsi que de dessiner des objets tels que des médiatrices. Dans un premier temps l'exploration se faisait autant avec le clavier qu'avec le périphérique haptique. Cependant l'utilisation conjointe de ces deux périphériques s'est avérée difficile. C'est pourquoi dans un second temps, les auteurs ont décidé de limiter l'utilisation du clavier. Pour cela ils ont créé des interactions permettant d'exécuter des commandes telles que la création de formes à l'aide de gestes avec le périphérique à retour de force.

Saarinen *et al.* [153] ont créé un environnement d'apprentissage du système solaire. Cette application contient six scènes plus une permettant de naviguer entre elles. Une scène présente les planètes du système solaire, et les orbites des planètes. Il est possible de toucher les planètes et de suivre leurs orbites grâce à un sillon au fond de la scène. Une autre scène présente la surface de la Terre : l'utilisateur peut sentir les reliefs, et le nom des continents est prononcé lorsqu'il passe dessus. Une autre scène permet d'étudier l'orbite de la Terre, dont les saisons en fonction de la position de la Terre. Il est aussi possible d'explorer les couches internes de la terre et son atmosphère grâce à deux autres scènes. Les différentes couches de la croûte terrestre donnent une sensation différente, reflétant la dureté de la couche, tandis que les couches de l'atmosphère provoquent des sons différents ainsi qu'un léger retour haptique. Cette application a été construite à l'aide d'un prototype d'architecture multimodale. Cette architecture a été améliorée dans le projet MICOLE, et cette application a été portée sur la nouvelle architecture. La nouvelle architecture sera présentée dans le chapitre IV.

Magnusson *et al.* ont développé AHEAD, un système de dessin multimodal [147]. Le dessin se fait soit avec un PHANToM soit avec une souris, sur une surface malléable virtuelle. En mode exploration, il est possible de ressentir le dessin avec le PHANToM. Il est possible d'activer un retour sonore permettant de se repérer sur la surface de dessin. Les figures dessinées peuvent être nommées et une synthèse vocale permet de prononcer le nom de la figure explorée. Il est aussi possible de guider le PHANToM vers la position de la souris, et à l'inverse de déplacer le curseur de la souris vers celui du PHANToM. Il est aussi possible de charger une image pour que l'utilisateur puisse la sentir en relief. Il est donc possible d'explorer, d'annoter et de modifier des schémas, des cartes et des graphiques grâce à cette application [146]. Cette application a été évaluée avec des enfants déficients visuels [105]. Pour commencer les enfants ont apprécié l'utilisation de l'application. Ensuite les résultats montrent que les utilisateurs déficients visuels utilisaient autant de temps pour dessiner les figures demandées avec le PHANToM que les utilisateurs voyants avec la souris. Enfin notons que le guidage du PHANToM par la souris a été utilisé avec succès par les enfants voyants pour faire explorer leur dessin à leurs camarades déficients visuels.

4.3.2. Autres systèmes

Petit *et al.* [134] ont créé le système MaskGen qui permet de transformer interactivement un graphique visuel en une représentation haptique lisible avec le périphérique STReSS^2. Ce périphérique permet de créer trois types de rendus : par points, des ondulations et des vibrations. L'utilisation de ces trois effets permet de représenter une vaste gamme de graphiques. Il existe aussi une synthèse vocale pour annoter des parties de schémas. Les auteurs donnent des exemples pour les trois catégories de graphiques que nous avons identifiées. Un exemple d'histogramme est donné, où les barres sont représentées avec un effet d'ondulation. Le contour des barres et les axes utilisent l'effet de vibration. Sur les cartes géographiques les frontières sont représentées avec les vibrations et les zones importantes avec les ondulations. Enfin les traits représentant une illustration simplifié d'une cathédrale gothique sont représentés par des vibrations. Les auteurs ont mené une évaluation avec des utilisateurs voyants et des utilisateurs déficients visuels. Les deux catégories d'utilisateurs ont pris autant de temps pour explorer les graphiques, mais les déficients visuels ont pris plus de temps pour répondre aux questions. On ne retient aucune différence en ce qui concerne les bonnes questions entre utilisateurs voyants et utilisateurs déficients visuels. La moyenne de réponses correctes varie entre 69,70% et 89,38% selon la tâche. Ce système est prometteur et a l'avantage de permettre l'exploration de tous types de graphiques. Cependant d'une part il est dépendant d'un périphérique particulier, et d'autre part il ne propose qu'une transcription d'informations visuelles en informations tactiles sans proposer de système d'aide ou de guidage mis à part la synthèse vocale qui permet de placer quelques annotations.

4.4. Conclusion

Remplacer des informations essentiellement visuelles par des représentations haptiques est une tâche difficile. Nous avons pu constater qu'a la fois dans le cas du retour de force que dans le cas du retour tactile il faut concevoir les techniques d'interaction avec précaution. Le problème principal provient du fait que la vision permet une vision d'ensemble alors que généralement l'haptique ne permet de n'avoir qu'un point de vue local. En particulier les bras articulés ne possèdent souvent qu'un seul point de contact. De nombreuses études ont été réalisées pour examiner l'exploration d'objets avec un bras articulé. Elles ont montré que les formes de taille supérieure à $10mm$ est aisée. Cependant l'exploration d'une scène contenant plusieurs objets se complique très vite.

Les graphiques sont des représentations essentielles dans la vie de tous les jours, que ce soit à l'école ou plus tard dans la vie active. Plusieurs équipes de recherches se sont penchées sur l'exploration de plusieurs types de graphiques grâce au retour tactile et au retour de force. Des systèmes permettent d'explorer avec succès des courbes, des histogrammes et des camemberts. Les cartes géographiques ont aussi été étudiées. Cependant dans le système testé, le retour tactile n'est pas plus efficace que le retour sonore proposé. De plus le retour tactile est mal exploité et l'exploration se fait à la souris, ce qui n'est pas pratique pour un déficient visuel [84]. Le rendu de schéma est aussi un problème difficile. La complexité de l'information présentée peut être élevée, les schémas peuvent contenir du texte, et ils possèdent une certaine structure. Les systèmes que nous

avons décrits permettent d'explorer de tels schémas de manière fidèle, cependant le retour haptique proposé reste limité. Un des systèmes utilise une représentation classique sur papier en relief, et un autre n'a que le retour de force d'un joystick comme représentation haptique.

L'utilisation d'un ordinateur de nos jours repose sur la capacité à naviguer dans son interface graphique. Plusieurs systèmes utilisant un retour sonore et un retour haptique ont été conçus pour explorer les interfaces graphiques habituelles. Ils permettent de ressentir les composants des interfaces. Cependant ces systèmes ne sont toujours pas utilisés couramment. Les aveugles utilisent majoritairement une synthèse vocale, une tablette braille et un clavier pour naviguer. Enfin nous avons vu des systèmes de guidage dans une scène 3D. Ces systèmes permettent à des utilisateurs aveugles de s'orienter dans une environnement virtuel. Dans le premier système présenté l'utilisateur est complètement passif, ce qui a été un reproche de la part des utilisateurs. Dans le second système l'utilisateur reçoit des indications grâce à des icônes que nous avons développées et qui sont présentées dans le chapitre II.

Enfin nous avons abordé des études sur des systèmes multimodaux utilisant l'haptique pour aider les enfants déficients visuels en milieu scolaire. Plusieurs systèmes ont été conçus par les partenaires du projet européen MICOLE, dont un système pour faire des exercices de géométrie, un système permettant d'explorer le système solaire et ses planètes et un système de dessin. Un autre système d'exploration de schémas destiné à des enfants déficients visuels a été présenté. Ce système a l'avantage de pouvoir représenter une grande variété de schémas. Cependant il ne propose pas de technique de guidage pour favoriser l'aspect collaboratif tel que dans les études précédentes et celles que nous présenterons dans le chapitre IV.

5. Conclusion

Nous avons étudié dans un premier temps les interfaces en général et la manière dont elles sont conçues. Il existe deux types de modèles d'interfaces : les modèles à couches qui font transiter les informations entre le système et l'utilisateur par analyse lexicale, syntaxique et sémantique, et les modèles à agents qui utilisent les langages objets pour répartir les tâches entre des entités. Il existe aussi des modèles d'interface hybrides qui utilisent à la fois des caractéristiques des modèles à couches et des caractéristiques des modèles à agents. La complexité des interfaces fait qu'il est souvent nécessaire de réaliser plusieurs tâches à la fois, ou que certaines tâches se divisent en plusieurs sous-tâches. Ceci est une première raison pour offrir plusieurs manières d'interagir avec un système informatisé. La deuxième raison est qu'il est possible qu'une des manières d'agir soit impossible, que ce soit à cause du contexte d'utilisation, par souci d'efficacité, ou par incapacité de la part de l'utilisateur. Cette problématique se cache sous le nom de multimodalité. Il est possible d'offrir de nouvelles manières d'agir avec un système soit en ajoutant des périphériques, soit en créant de nouveaux langages d'interaction. L'organisation des techniques d'interaction offertes par une application est structurée grâce des espaces de conception et de caractérisation. Certains espaces permettent d'adopter le point de vue de l'utilisateur afin d'étudier la manière dont l'utilisateur

agit sur le système. D'autres espaces adoptent le point de vue du système, ce qui permet de savoir comment les modalités sont organisées. Enfin certains espaces adoptent les deux points de vue pour montrer à la fois les deux aspects. Grâce à l'étude de ces espaces, nous pouvons réfléchir sur la manière dont nous organiserons nos techniques d'interaction au sein d'une application.

Les personnes non voyantes sont habituées à utiliser le toucher pour compenser leur absence de vision. La canne blanche permet de ressentir les obstacles lors des déplacements. Ils lisent du texte grâce à l'alphabet braille qui est un alphabet tactile. Le problème majeur de l'haptique est la quasi-absence de point de vue global. Ceci est très gênant lorsqu'il s'agir de représenter des données sous forme graphique. Le travail exposé dans les chapitres suivant consiste à développer des techniques d'interaction permettant de guider un utilisateur déficient visuel et de lui permettre d'acquérir facilement l'information dont il a besoin. Le premier pas fût d'étudier l'haptique d'un point de vue biologique. Les sensations tactiles proviennent de stimulation de mécanorécepteurs situés dans la peau, alors que les sensations kinesthésiques sont dues à la stimulation de mécanorécepteurs dans les muscles, les tendons et les articulations. La sensation de température est parfois assimilée à une sensation tactile, et est ressentie par des thermorécepteurs dans la peau. Cependant nous ne l'étudierons pas par la suite. La transmission d'information à l'utilisateur par ces sens se fait grâce à des périphériques.

Nous avons évoqué quelques périphériques que nous avons classé par technologie, tout en séparant les périphériques offrant principalement un retour tactile et ceux qui donnent essentiellement un retour de force. La distinction n'est pas toujours nette car un périphérique à retour de force peut vibrer et ainsi stimuler des mécanorécepteurs de la peau. Parmi les périphériques tactiles nous avons retenu les matrices de picots, telles que la souris VTPlayer. Ces périphériques sont souvent placés sur le bout des doigts et permettent de représenter des informations sous forme de grille de points. Les périphériques tactiles les plus courants sont les vibreurs. Ils sont utilisés quotidiennement dans les téléphones portables pour remplacer la sonnerie quand le retour sonore n'est pas souhaité. Ils sont aussi utilisé dans certains contrôleurs de jeu pour augmenter le réalisme. Cependant les effets tactiles proposés par ces périphériques sont très limités. À l'instar des matrices de picots, certains périphériques tactiles à vibreurs permettent d'afficher des informations sous forme de grilles de points. Ces périphériques sont intéressant sur de nombreux points, dont la miniaturisation qui fait défaut aux matrices de picots. Malheureusement ces périphériques sont peu ou pas exploités commercialement, et il est donc difficile de se les procurer. Il existe des périphériques qui stimulent les mécanorécepteurs directement en créant des décharges électriques. Ils se placent sur la langue ou sur le front, et affichent les informations sous forme de grilles de points. Par contre la matrice utilisée est plus grande, et les points sont plus espacés. De plus la sensation peut être désagréable. Nous n'avons donc pas envisagé pas cette solution. Enfin certains périphériques compriment des zones de la peau pour simuler la sensation de textures. Cet effet est intéressant, et mériterait d'être étudié. Cependant la zone stimulée par ces périphériques est trop grande et ne convient donc pas à nos besoins.

Les périphériques à retour de force sont souvent aussi des périphériques de pointage, ce qui permet de manipuler des objets et d'en ressentir les propriétés. Ils ont déjà fait leur apparition parmi le grand public sous forme de contrôleurs de jeux. Les joysticks et les volants à retour de force permettent d'augmenter le réalisme des jeux. Dans le milieu de la recherche on trouve une grande variété de gras articulés à retour de force. Ces périphériques possèdent généralement entre 3 et 7 degrés de liberté en sortie : 3 pour les translations, 3 pour les rotations et parfois 1 pour le pincement. Les pantographes offrent une autre solution pour donner un retour de force. De base un pantographe offre 2 degrés de liberté. Cependant il est possible d'augmenter leur nombre en combinant plusieurs pantographes. Ainsi des périphériques utilisant des pantographes possèdent jusqu'à 7 degrés de liberté en sortie. Les systèmes pneumatiques ont l'avantage de pouvoir exercer des forces plus importantes, ce qui augmente le niveau de réalisme. Cependant en contrepartie ces périphériques sont volumineux. Enfin les exosquelettes possèdent beaucoup de degrés de liberté. Ils contraignent les articulations, ce qui permet entre autres de manipuler des objets virtuels.

L'haptique est déjà utilisée dans de nombreux domaines. Nous avons évoqué quelques exemples d'utilisations courantes. Les simulations utilisent le retour haptique pour augmenter le réalisme des environnements virtuels. Ce retour permet aussi de créer des environnement de simulation pour l'apprentissage de techniques utilisant la sensation du toucher comme dans les systèmes de formation de vétérinaires et d'ostéopathes que nous avons évoquer. Il est aussi possible d'effectuer de la téléopération, où le retour haptique permet de contrôler la manipulation. Le retour haptique est aussi utilisé dans certaines interfaces, toujours pour augmenter le réalisme et pour donner des indications à l'utilisateur sur la manière dont le système interprète ses actions. Nous avons abordé des études utilisant le retour haptique pour explorer des environnements virtuels. La première propose un système de guidage pour l'exploration d'une scène 3D, et l'autre étudie une technique de zoom afin d'explorer des formes tactiles. Enfin nous avons abordé un sujet essentiel à notre étude : les icônes, et en particulier les icônes haptiques. Parmi les systèmes d'icônes que nous avons abordés, nous retenons l'idée de structuration d'icônes tactiles afin de coder plusieurs informations grâce à des paramètres indépendants, et l'idée d'afficher des informations sous forme de grilles de points.

Nous nous intéressons en particulier à l'utilisation de l'haptique dans l'enseignement pour les enfants déficients visuels. Nous avons donc abordé dans le chapitre des méthodes de présentation et d'exploration d'environnements virtuels avec un retour de force ou un retour tactile. Nous avons étudié l'exploration de différents types d'objets, de graphiques, d'interfaces et de scènes 3D. Ensuite nous avons abordé le cœur de notre problématique, à savoir l'utilisation de systèmes haptiques dans l'enseignement. Comme nous avons pu le constater l'haptique peut être utilisé dans de nombreux domaines, comme la géométrie, l'astronomie, le dessin, ainsi que pour la présentation de documents de différentes formes.

Dans les chapitres qui suivent, nous allons détailler nos contributions. Pour commencer, nous étudierons dans le chapitre II un nouveau système d'icônes tactiles utilisant une matrice de picots.

Chapitre II

Icônes tactiles

Le talent, ça se travaille, mais ça ne se perd pas.

Un philosophe belge

Résumé

Nous définissons une technique nouvelle d'interaction permettant de coder des informations sous formes d'icônes à l'aide de matrices de picots. Nous déclinons ces icônes en dex catégories : les icônes statiques et les icônes dynamiques. Les premières sont des motifs affichés de manière continue, alors que les secondes sont des animations créées en faisant évoluer le motif affiché sur la matrice au cours du temps. Après une série de définitions, nous abordons une série d'expériences visant à trouver des paramètres adéquats pour la conception de nos icônes. L'étude suivante consiste à utiliser la structure définie précédemment afin de coder plusieurs informations sur chaque icône. Nous utilisons ensuite ces icônes pour créer une méthode de guidage sur des formes géométriques. Nous présentons une expérience comparant cette méthode à une méthode plus tradictionnelle consistant à lever les picots correspondant aux pixels sombres autour du curseur. Enfin nous définissons une analogie entre nos icônes et la notation musicale. Ce travail préparatoire sera utilisé à l'avenir afin de créer des icônes crossmodales.

1. Introduction

Les icônes sont des moyens de représenter une idée ou une information sur un espace limité. Une icône n'a pas forcément un lien syntaxique avec l'information qu'elle véhicule : elle peut être construite avec un codage comme un alphabet, ce qui nécessite un apprentissage du codage en question. Elle peut aussi être construite par une image suggérant l'information, et dans ce cas l'information devrait être compréhensible sans apprentissage. Nous commencerons ce chapitre par une série de définitions permettant d'une part d'expliquer la technique d'interaction que nous avons conçue, et d'autre part de la structurer. Nous présentons ensuite une série d'expériences visant à explorer certains paramètres afin de trouver des ensembles d'icônes utilisables pour coder de l'information. L'étude suivante consiste en une expérience ayant pour but de créer des icônes permettant de véhiculer plusieurs informations à l'instar des Tactons vibrotactiles [31]. Nous utiliserons ensuite ces icônes dans une application d'exploration de formes géométriques. Nous finirons par un travail de

réflexion sur une analogie entre nos icônes et la notation musicale, pouvant aboutir à l'avenir sur des icônes crossmodales telles que celles développées par Hoggan et Brewster à partir d'icônes sonores et vibrotactiles [70].

2. Le concept d'icônes

Les matrices de picots peuvent être utilisées de plusieurs manières, en fonction du type d'information qu'on désire afficher, et la manière de l'encoder. Le Braille est devenu une méthode naturelle de présentation d'information avec des picots. Les lettres Braille utilisent 2×3 picots, 2×4 picots pour la version étendue. Cette version a été développée pour les systèmes informatiques afin de gérer plus simplement les majuscules et les chiffres par exemple. En effet grâce à 8 picots, on peut représenter exactement la même quantité d'information qu'avec un octet. Plusieurs standards sont définis pour le braille mathématique comme MathML, un standard mathématique français pour le web, qui peut être transformé en braille par des systèmes comme BraMaNet [1], Lambda [51] ou MaWEn [8]. Des conventions ont aussi été définies pour la notation musicale [52].

2.1. Définitions

Nous allons commencer par définir la notion de *Tacton à matrices de picots*, une extension des *Tactons vibrotactiles* définis par Brown *et al.* [30]. Ces derniers donnent la définition de Tacton ci-dessous :

Définition 1 Un *Tacton* est un message tactile abstrait et structuré, pouvant être utilisé pour transmettre de l'information de manière non-visuelle.[1]

Cette définition donne un bon aperçu de l'idée générale des icônes que nous allons créer. Cependant elle est trop générique et ne permet pas de caractériser assez finement nos icônes. Nous avons établi les définitions ci-dessous afin d'affiner la notion de Tacton à notre cas. Dans le reste de cette étude, sauf mention contraire, nous désignerons par Tacton un Tacton utilisant une matrice de picots de taille $n \times p$, chaque picot pouvant se trouver dans deux états : *haut* et *bas*. Il est cependant possible d'étendre ces définitions à des matrices de picots où chaque picot peut avoir une hauteur comprise entre une valeur min et une valeur max. Cette extension n'est pas étudiée ici, mais pourra faire l'objet d'une étude future.

Définition 2 Un *motif* m est défini comme la matrice représentant l'état de chaque picot $m_{i,j}$:

$$m_{i,j} = \begin{cases} 0 & \text{si l'état du picot } (i,j) \text{ est } bas \quad 1 \leq i \leq n \\ 1 & \text{si l'état du picot } (i,j) \text{ est } haut \quad 1 \leq j \leq p \end{cases}$$

L'ensemble des motifs de taille $n \times p$ est noté $M_{n,p}$. Cet ensemble est fini et a pour cardinalité $card(M_{n,p}) = 2^{n \times p}$.

Définition 3 Un *Tacton statique* est défini par un motif. Ce motif est affiché jusqu'à ce que le Tacton change.

[1] Dans le texte : « Tactons are structured, abstract, tactile messages which can be used to communicate information non-visually »

L'ensemble des Tactons statiques de taille $n \times p$, $Ts_{n,p}$, est isomorphe à l'ensemble des motifs de taille $n \times p$, c'est-à-dire $M_{n,p}$. L'ensemble des Tactons statiques est noté $Ts = \bigcup_{n,p \in \mathbb{N}^*} Ts_{n,p}$.

Définition 4 Une *image* est une étape d'une animation. On note une image $i = (m, d)$. Le motif m correspond à ce qui est affiché sur la matrice pendant l'étape de l'animation correspondant à l'image, et le nombre rationnel d correspond à la *durée* de l'image. Cette dernière n'a pas d'unité car elle correspond à un coefficient appliqué à l'*unité temporelle* de l'animation (voir définition 5 ci-dessous). 1 correspond à la durée de base, donc une image ayant une durée de 2 est affichée deux fois plus longtemps. Par contre une image ayant un coefficient de $1/4$ est affichée quatre fois moins longtemps.

Le motif peut être extrait grâce à la première projection : $\pi_1(i) = m$ et la durée pouvant être extraite à l'aide de la seconde projection : $\pi_2(i) = d$.

L'ensemble des images de taille $n \times p$, $I_{n,p}$, est le produit cartésien de l'ensemble des motifs de taille $n \times p$ et l'ensemble des nombres rationnels strictement positifs : $I_{n,p} = M_{n,p} \times \mathbb{Q}_+^*$.

Définition 5 Un *Tacton dynamique* est une animation. On note un Tacton dynamique $t_d = (s, \tau)$. Le k-uplet d'images s correspond à la séquence de motifs à afficher, et l'entier naturel non nul τ représente l'*unité temporelle* (en millisecondes) de l'animation, c'est-à-dire le temps d'affichage d'une image de durée 1. Nous utilisons un nombre entier pour simplifier les raisonnements. Cependant si nous avons besoin de plus de précision nous pouvons diviser les durées des images. Néamoins il est peu probable que des résolutions inférieures à la milliseconde soient nécessaires.

La séquence d'images peut être extraite par la première projection : $\pi_1(t_d) = s$, et la $j^{\text{ème}}$ image peut être extraite par composition de cette projection avec la $j^{\text{ème}}$ projection sur le k-uplet obtenu : $\pi_j(\pi_1(t_d)) = (\pi_j \circ \pi_1)(t_d)$. L'unité temporelle peut être extraite à l'aide de la seconde projection $\pi_2(t_d) = \tau$. La durée d'un Tacton dynamique t_d peut être calculée par $\tau \times \Sigma_{j=1}^{k} d_j$, avec k la longueur de t_d, $\tau = \pi_2(t_d)$ l'unité temporelle de t_d et $d_j = (\pi_2 \circ \pi_j \circ \pi_1)(t_d)$.

L'ensemble des Tactons dynamiques de taille $n \times p$, $Td_{n,p}$, est le produit cartésien entre les k-uplets d'images et l'ensemble des entiers naturels non nuls : $Td_{n,p} = \bigcup_{k=1}^{\infty} I_{n,p}^k \times \mathbb{N}^*$. k représente la *longueur* du Tacton, c'est-à-dire le nombre total d'images. L'ensemble des Tactons dynamiques est noté $Td = \bigcup_{n,p \in \mathbb{N}^*} Td_{n,p}$.

2.2. Paramètres et dimensions des Tactons

Les notions définies dans la section précédente permettent à la fois d'expliquer la nature des Tactons, mais aussi de permettre de définir une structure telle que décrite dans la définition 1. Il faut cependant préciser quelques autres concepts que nous allons réutiliser plus tard.

Définition 6 Les *paramètres* correspondent aux propriétés physiques des Tactons tels que définis dans la section précédente : motif, image, durée d'une image, unité temporelle et séquence d'images.

Définition 7 Les *dimensions* d'un Tacton correspondent à des propriétés logiques basées sur ses paramètres. La forme, la taille, la vitesse de clignotement ou la vitesse d'animation sont des exemples de dimensions. Ce sont des ensembles finis, donc lorsque l'on définit une dimension, on doit aussi définir ses valeurs. Par exemple $taille = \{grand, petit\}$ est une dimension.

Définition 8 *L'information* est le message à transmettre à l'utilisateur.

Les *ensembles de Tactons* sont construits par produit cartésien de ses dimensions. La cardinalité d'un ensemble de Tactons est donc le produit des cardinalités de ses dimensions. Une collection de dimensions n'engendre qu'un seul ensemble de Tactons. Par contre un ensemble de Tactons peut être défini par plusieurs collections de dimensions. Le choix des dimensions dépend des informations que l'on souhaite transmettre grâce à l'ensemble de Tactons en question. Ainsi chaque morceau d'information utilise une dimension. Par exemple si on veut représenter des objets par forme, couleur et taille : on peut utiliser la forme du Tacton comme étant la forme de l'objet, la taille du Tacton comme étant la taille de l'objet, et la vitesse de clignotement du Tacton comme étant la couleur. Chaque valeur de chaque dimension représente une valeur de chaque information. Donc si les objets à représenter peuvent avoir deux formes, il faut définir deux formes pour les Tactons, et ainsi de suite pour la taille et la couleur.

Dans la section suivante, nous allons explorer les paramètres des Tactons afin d'obtenir des ensemble de Tactons satisfaisants. Nous allons commencer par expérimenter des Tactons à une seule dimension.

3. Les Tactons mono-dimensionnels

Les premières expériences visent à examiner la reconnaissance de Tactons simples. Il est donc naturel de se pencher sur des Tactons mono-dimensionnels. La dimension utilisée ne repose que sur le paramètre de motif. En effet, pour le moment nous ne testons pas la reconnaissance de rythmes, de durées d'images, etc. Nous avons pensé que ce genre d'informations pouvait être particulièrement utile pour représenter des directions. Nous avons donc utilisé des Tactons statiques et dynamiques pour représenter 8 directions [141]. Les 4 premières directions : haut, bas, gauche et droite sont appelés *Tactons radiaux*, et les quatre coins sont appelés *Tactons diagonaux* (figure 1).

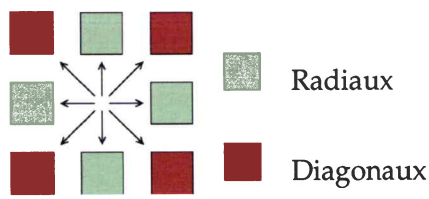


Figure 1 – Tactons radiaux et diagonaux

Trois types de Tactons ont été testés. Les premiers sont des Tactons statiques, et les deux autres sont des Tactons dynamiques. Les Tactons statiques indiquent la direction par la position des picots levés sur la matrice. Les *Tactons clignotants* sont une alternance entre un motif ayant un ou plusieurs picots levés et le motif avec aucun picot levé. Ils indiquent la direction de manière analogue aux Tactons statiques grâce au motif à picots levés. Les *Tactons en vagues* représentent l'évolution d'un motif dans le temps. La direction représentée est celle dans laquelle le motif se déplace.

3.1. Méthodologie

Plusieurs expériences ont été menées pour identifier des ensembles de Tactons discriminables. Le protocole utilisé dans ces expériences est similaire, c'est pourquoi nous allons dans un premier temps présenter la partie commune. Les Tactons étaient affichés sur une matrice de 4×4 picots d'une souris VTPlayer (figure 2). Les Tactons créés seront donc pris dans $Ts_{4,4}$ pour les Tactons statiques, et $Td_{4,4}$ pour les Tactons dynamiques. La cellule utilisée était toujours celle sous l'index de la main dominante. La main tenant la VTPlayer était masquée afin d'empêcher les utilisateurs d'avoir une aide visuelle. L'unité temporelle n'est pas un paramètre variable dans ces expériences. Tous les Tactons dynamiques ont donc une unité temporelle de $\tau = 100ms$, et chaque image a une durée de $d = 1$. Nous avons choisi ne ne pas faire varier ces paramètres dans ces expériences afin de nous concentrer sur le paramètre de motif. Nous ferons varier ces paramètres dans les sections suivantes.



Figure 2 – VTPlayer et ses matrices de picots

Chaque expérience était divisée en plusieurs blocs pour tester séparément chaque ensemble de Tactons. L'ordre des blocs était balancé entre les utilisateurs afin d'éviter un effet d'apprentissage. Chaque bloc de test commençait par une session d'entraînement où l'utilisateur pouvait explorer les 8 Tactons de l'ensemble testé. Une fois qu'il se sentait apte à reconnaître les directions il pressait une touche pour commencer les tests. L'interface est semblable à celle de l'entraînement (figure 3). L'utilisateur devait reconnaître la direction représentée par 100 Tactons. Pour chacun lorsque l'utilisateur reconnaît le Tacton, il clique dans la zone correspondante sur l'écran et retourne dans la zone centrale pour lancer le Tacton suivant. À la fin des sessions, les utilisateurs sont informés du nombre d'erreurs qu'ils ont effectuées.

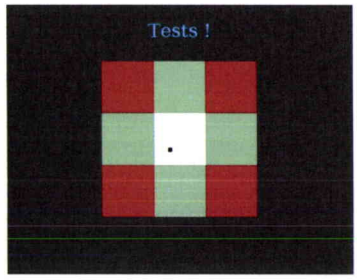


Figure 3 – Capture d'écran du logiciel de tests

Dans chaque expérience, chaque utilisateur a testé tous les ensembles de Tactons utilisés dans l'expérience. Les erreurs mesurées correspondent au nombre de réponses erronées, et le temps correspond au temps de réponse moyen par Tacton. À la fin de chaque expérience les utilisateurs remplissaient un questionnaire (voir annexe A) pour fournir des appréciations et des critiques vis-à-vis des Tactons et de la façon dont ils étaient présentés, ainsi que pour fournir des suggestions. Les résultats ont été analysés à l'aide de statistiques non paramétriques car une partie des échantillons obtenus n'est pas répartie selon une loi normale. Des tests de la somme des rangs de Kruskal-Wallis ont été réalisés pour trouver les différences significatives entre les échantillons. Dans le cas d'échantillons significatifs, nous avons réalisé une comparaison par paires à l'aide d'un test de la somme des rangs de Wilcoxon avec un ajustement des p-values de Holm.

3.2. Expérience 1 : comparaison de Tactons statiques, clignotants et vagues

Dans la première expérience nous allons explorer les possibilités offertes par les trois sortes de Tactons : statiques, clignotants et vagues. Ces ensembles de Tactons ont été construits itérativement suite à des tests préliminaires informels. Les motifs sont générés à partir de combinaisons de simples lignes. Différents nombres de picots levés ont été utilisés lors de ces tests afin de voir si la quantité de picots levés influe sur la reconnaissance. Neuf utilisateurs de 24 à 48 ans (médiane : $26 ans$) ont participé à ces premiers tests, afin d'identifier 4 ensembles de Tactons. Tous les utilisateurs étaient voyants, sans problème de perception tactile connu, et recrutés au département Informatique de l'Université Paul Verlaine – Metz.

Les 4 ensembles de Tactons ont été sélectionnés afin d'examiner certains facteurs soupçonnés de pouvoir influer sur la discrimination des Tactons. Les deux premiers ensembles de Tactons utilisent peu de picots levés (2 pour chaque Tacton) pour indiquer les directions. La différence entre les deux ensembles est que le premier contient des icônes clignotantes (figure 4(a)) alors que le second est composé de Tactons statiques (figure 4(b)).

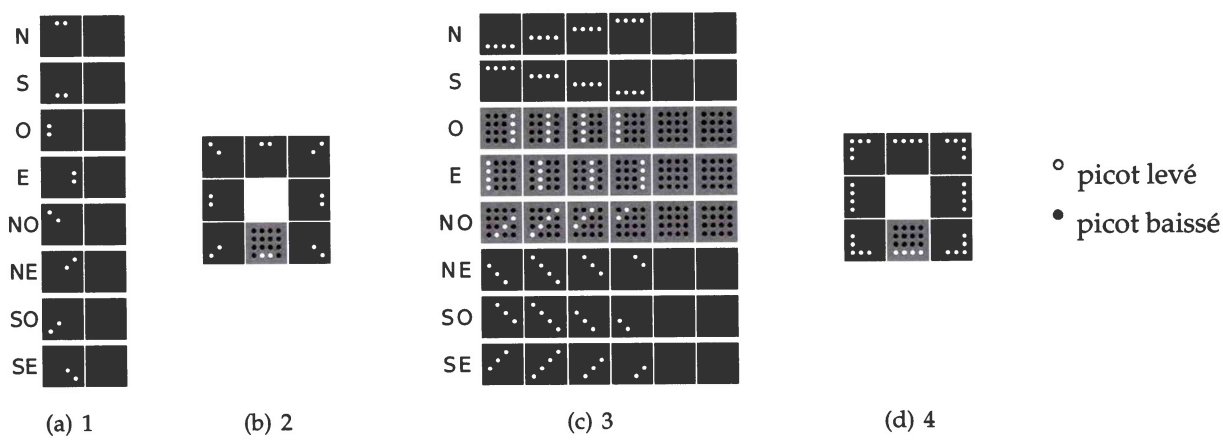


Figure 4 – Ensembles de Tactons utilisés dans l'expérience 1

L'ensemble 3 (figure 4(c)) a été créé sur le modèle des vagues : une ligne se déplace dans la direction à indiquer. Les Tactons radiaux utilisent une ligne de 4 picots de longueur et de 1 seul d'épaisseur. 4 images sont donc nécessaires pour parcourir la matrice. Pour les diagonales nous avons choisi de maintenir le nombre d'images. Nous avons donc opté pour les 4 diagonales parallèles les plus grandes afin de représenter la ligne qui se déplace. Les 4 images retenues utilisent 3, 4, 3 et 2 picots levés. Une pause a été placée à la fin de chaque Tacton en vague, car les tests pilotes informels ont montré que sans elle il est difficile voire impossible de comprendre le sens de déplacement de la ligne. Nous avons dessiné deux images à motif vide et de durée 1 sur la figure 4(c) au lieu d'une image de durée 2 afin de simplifier la représentation visuelle du schéma. Cependant au niveau des Tactons, ces deux représentations sont équivalentes.

Enfin l'ensemble 4 (figure 4(d)) est un ensemble de Tactons statiques utilisant plus de picots que l'ensemble 2. Les Tactons radiaux indiquent la direction par une ligne de 4 picots sur 1, tandis que les diagonaux l'indiquent avec un angle de 3 picots de côté et de 1 d'épaisseur. Nous avons essayé de maintenir le nombre de picots levés au sein du même ensemble de Tactons afin de maintenir une surface tactile équivalente au sein des ensembles de Tactons pour ce premier stade d'expérimentation.

Nous voulions vérifier grâce à cette expérience que les Tactons statiques avec peu de picots sont plus durs à discriminer que les Tactons à beaucoup de picots. Nous cherchions aussi à savoir si les Tactons statiques sont plus faciles à reconnaître que les Tactons dynamiques, et en particulier les vagues. Enfin nous voulions aussi savoir si les Tactons clignotants améliorent ou perturbent la reconnaissance d'un Tacton. Nous posons pour cela plusieurs hypothèses :

- Les Tactons avec plus de picots levés seront plus faciles à reconnaître que les autres. Ceci vient du fait qu'en augmentant la surface tactile nous pensons faciliter la tâche de l'utilisateur.
- Les Tactons statiques seront plus faciles à reconnaître que les Tactons en vagues. En effet l'utilisateur contrôle l'exploration dans le cas des Tactons statiques, contrairement au cas des Tactons en vagues.
- Les Tactons clignotants sont plus difficiles à reconnaître que les Tactons statiques. La raison de cette hypothèse est que l'information ne sera présente que la moitié du temps, ce qui peut perturber la reconnaissance.

Résultats.

La figure 5 montre les taux d'erreur et les temps de réponses pour les 4 ensembles de Tactons testés. Les résultats montrent que les utilisateurs arrivent à identifier plus efficacement les Tactons de l'ensemble 4 avec une médiane de 0% d'erreurs, contre 17% pour l'ensemble 1, 8% pour l'ensemble 2, et 3% pour l'ensemble 3. Le test de Kruskal-Wallis montre une différence significative entre les quatre ensembles testés ($\chi^2 = 19{,}93, p < 0{,}001$). La comparaison des paires montrent que l'ensemble 4 provoque significativement moins d'erreurs que les autres ensembles ($p < 0{,}002$ pour les ensembles 2 et 3, et $p = 0{,}04$ pour l'ensemble 3). Nous ne constatons aucune différence significative entre les autres ensembles. L'ensemble 4 a aussi été reconnu significativement plus rapidement que les autres

ensembles, avec un temps de réponse médian de $1{,}66s$, comparé à $2{,}64s$ pour l'ensemble 1, $2{,}69s$ pour l'ensemble 2 et $2{,}74s$ pour l'ensemble 3. Nous pouvons observer une différence significative grâce au test de Kruskal-Wallis ($\chi^2 = 16{,}32$, $p < 0{,}001$). Encore une fois nous observons de meilleurs résultats pour l'ensemble 4 que pour les autres ensembles ($p = 0{,}002$ pour l'ensemble 1, et $p = 0{,}01$ pour les ensembles 2 et 3). Aucune autre différence significative n'est à signaler.

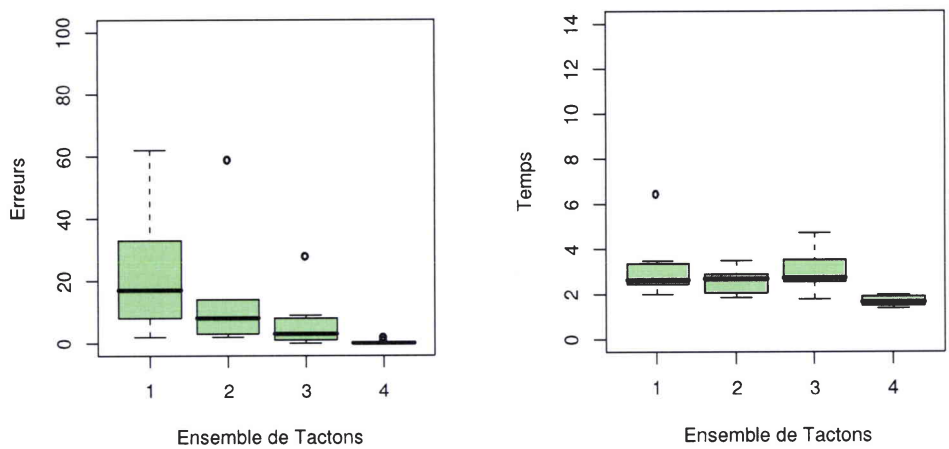


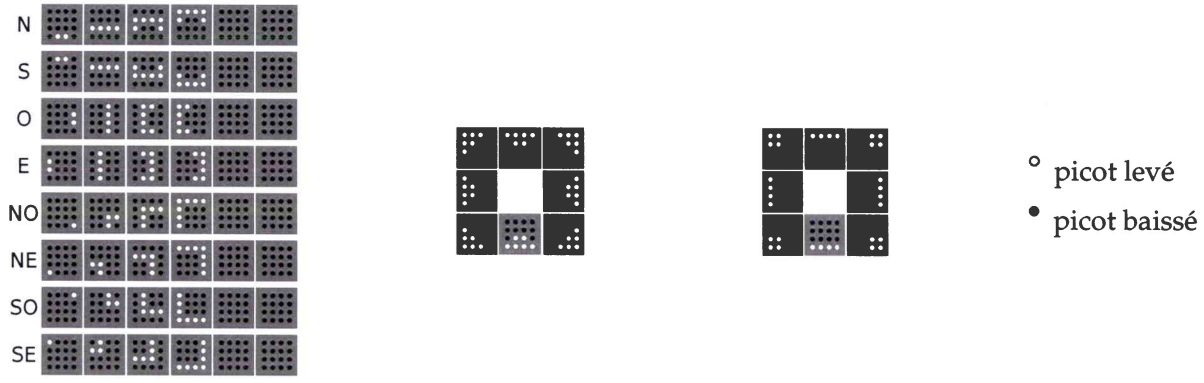
Figure 5 – Expérience 1 : erreurs et temps de réponse

Discussion.

D'après ces résultats nous ne pouvons pas affirmer que les Tactons avec plus de picots levés sont plus faciles à reconnaître que les Tactons à moins de picots levés car l'ensemble 3 ne se démarque pas des ensembles 1 et 2. La première hypothèse qui avance que les Tactons ayant plus de picots levés sont plus facilement identifiables que les autres Tactons est donc rejetée. Cependant l'ensemble 4 est plus facile à reconnaître que les autres dans cette expérience : les utilisateurs ont fait significativement moins d'erreurs, et ont eu besoin de moins de temps pour répondre. Les résultats de l'ensemble 2 ne sont pas meilleurs que ceux de l'ensemble 3. Nous devons donc rejeter la seconde hypothèse qui stipule que les Tactons statiques sont plus faciles à reconnaître que les Tactons en vagues. Les Tactons clignotants ont provoqué beaucoup d'erreurs. Ceci peut être dû au motif utilisé, qui utilise peu de picots. Nous devons trouver des motifs plus facilement distinguables avant d'explorer plus en détails les Tactons clignotants, ou au contraire de les abandonner. En effet il serait nécessaire d'examiner la vitesse de clignotement, ainsi que le rythme des images pour en déduire des valeurs de paramètres utilisables. Cependant ceci requiert une étude à part entière (voir section 4), donc pour cette expérience nous nous sommes concentrés sur les Tactons statiques et en vagues dans un premier temps. Pour le moment nous devons rejeter la troisième hypothèse qui prévoyait que les Tactons statiques seraient plus efficaces que les Tactons clignotants car l'ensemble 2 n'est pas significativement plus efficace que l'ensemble 1.

3.3. Expérience 2 : plus de picots, et formes grossissantes

Dans le questionnaire de l'expérience précédente nous avons demandé aux utilisateurs de proposer d'autres ensembles de Tactons, selon ce qu'ils ont aimé ou souhaité par rapport aux tests. Après de nouveaux tests pilotes informels, nous avons sélectionné parmi les propositions deux nouveaux ensembles de Tactons statiques (ensembles 6, figure 6(b) et 7, figure 6(c)) et un ensemble de Tactons en vagues (ensemble 5, figure 6(a)). L'expérience 2 permet de comparer les résultats de ces 3 nouveaux ensembles de Tactons avec le meilleur ensemble obtenu lors de l'expérience précédente, c'est-à-dire le 4 (figure 4(d)). Ainsi nous cherchons à trouver de meilleurs Tactons statiques et de meilleurs Tactons à vagues. Onze utilisateurs, âgés de 23 à 27 ans ont participé à cette expérience. Ils ont été recrutés parmi des étudiants en Master d'informatique de l'Université Paul Verlaine – Metz. Aucun d'entre eux n'a participé à l'expérience précédente. Ils sont tous droitiers et aucun ne souffre d'un problème connu de perception tactile.



(a) 5 (b) 6 (c) 7

Figure 6 – Ensembles de Tactons utilisés dans l'expérience 2

Les ensembles statiques que nous avons sélectionnés sont ceux qui utilisent le plus de picots levés puisque le meilleur ensemble de l'expérience précédente était celui qui utilise le plus de picots. De plus les ensembles préférés des utilisateurs selon les questionnaires sont ceux qui utilisent le plus de picots. Le but est de trouver des ensembles de Tactons aussi faciles et rapides à reconnaître qu'agréables pour les utilisateurs. L'ensemble de Tactons dynamique créé utilise un nouveau principe : des vagues qui grandissent, c'est à dire que le nombre de picots levés augmente d'image en image. Nous espérons que cette façon de représenter l'information aide à reconnaître la direction.

Pour cette expérience nous posons les hypothèses suivantes :

- Les Tactons avec plus de picots levés sont plus faciles à reconnaître que les Tactons avec moins de picots levés. En outre nous prévoyons de meilleurs résultats avec l'ensemble 6 qu'avec les

ensembles 4 et 7. Cette hypothèse est analogue à celle posée dans la première expérience, à ceci près qu'elle porte sur les nouveaux Tactons.

- Les vagues qui grandissent de l'ensemble 5 vont améliorer la reconnaissance, et nous obtiendrons donc de meilleurs résultats que ceux de l'ensemble 3 de l'expérience précédente. Cette hypothèse vient du fait que nous pensons que le fait d'utiliser un plus grand motif pour indiquer la direction pointée réduira les ambiguïtés.

Résultats.

Les taux d'erreur et les temps de réponse moyens sont représentés sur les graphiques de la figure 7. Encore une fois l'ensemble 4 est l'ensemble reconnu le plus facilement avec un taux d'erreur médian de 2%. L'ensemble 7 obtient un taux d'erreur de 10%, alors que les ensembles 5 et 6 obtiennent respectivement 23% et 14%. Le test de Kruskal-Wallis montre qu'il y a une différence significative de taux d'erreurs entre certains ensembles de Tactons de cette expérience ($\chi^2 = 10{,}22$, $p = 0{,}01$). La comparaison des paires révèle que l'ensemble 4 provoque significativement moins d'erreurs que l'ensemble 5 ($p = 0{,}04$). L'ensemble 7 est le plus rapidement reconnu avec une médiane de seulement $1{,}91s$ par Tacton, suivi de l'ensemble 4 qui nécessitait $2{,}33s$, puis l'ensemble 6 avec une médiane de $2{,}49s$, et plus loin l'ensemble 5 avec une médiane de $3{,}91s$. Encore une fois le test de Kruskal-Wallis permet de trouver une différence significative entre les temps de réponses pour les ensembles de Tactons testés dans cette expérience ($\chi^2 = 21{,}39$, $p < 0{,}001$). La comparaison des paires suggère que l'ensemble 7 est reconnu significativement plus rapidement que les autres ensembles : $p = 0{,}03$ pour l'ensemble 4, $p < 0{,}001$ pour l'ensemble 5 et $p = 0{,}03$ pour l'ensemble 6. À l'inverse l'ensemble 5 est significativement plus lent que l'ensemble 4 ($p = 0{,}03$), l'ensemble 6 ($p = 0{,}01$), ainsi que l'ensemble 7 déjà mentionné.

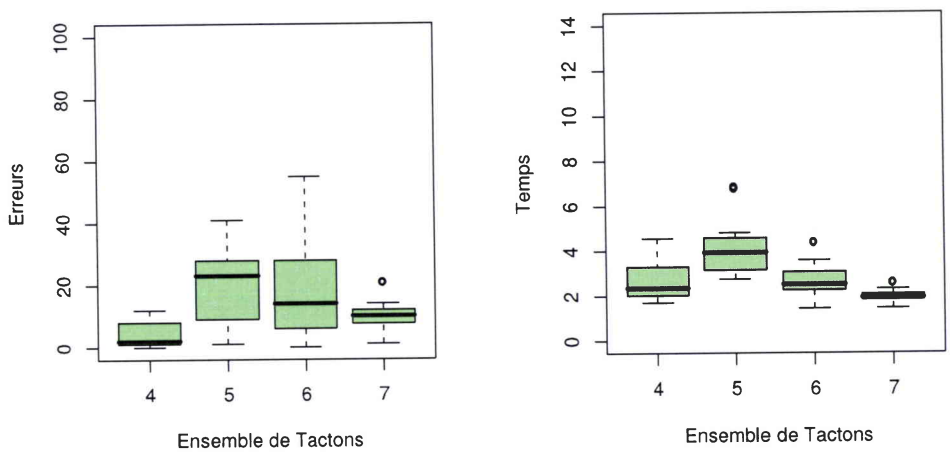


Figure 7 – Expérience 2 : erreurs et temps de réponse

Discussion.

Les Tactons de l'ensemble 4 sont toujours les Tactons les plus satisfaisants : ils ont toujours un faible taux d'erreur, et se trouvent être significativement meilleurs que les Tactons de l'ensemble 5 en terme de taux d'erreurs et de temps d'exploration. Cependant l'ensemble 7 est reconnu significativement plus rapidement que tous les autres ensembles. Cet ensemble doit donc être considéré pour les études suivantes. L'ensemble 5 est clairement difficile à reconnaître. D'après les questionnaires, le changement de la forme affichée au cours des images a perturbé les utilisateurs, ce qui fait qu'ils ont eu du mal à comprendre les étapes des animations. Les Tactons dynamiques requièrent potentiellement plus de temps pour être reconnus vu que si l'information n'est pas comprise après une vague, l'utilisateur doit attendre pour en sentir une autre. L'ensemble 5 utilise 6 images de $100ms$, ce qui fait que chaque vague dure $600ms$, ce qui signifie que les utilisateurs avaient besoin d'environ 7 vagues en moyenne pour reconnaître un Tacton. Le temps de reconnaissance de ces Tactons peut varier selon la vitesse de lecture. Il serait intéressant d'évaluer ce paramètre dans des expériences ultérieures. Les faibles résultats de l'ensemble 5 nous forcent à rejeter la seconde hypothèse. Une nouvelle expérience sera conduite afin d'essayer de trouver des Tactons dynamiques plus satisfaisants (section 3.5). L'ensemble 6 a un taux d'erreur élevé, ainsi qu'un fort temps de réponse moyen. Il est possible que la similitude des motifs des Tactons de cet ensemble était trop importante. De plus du fait de leur épaisseur, il y avait peu d'écart entre deux directions antagonistes (haut et bas par exemple). Les Tactons des ensembles 4 et 7 utilisent les picots du bord de la matrice, alors que ceux de l'ensemble 6 utilisent tous les picots du milieu. Le faible taux d'erreur des ensembles 4 et 7 comparé à celui de l'ensemble 6 nous force à rejeter la première hypothèse selon laquelle l'augmentation du nombre de picots levés améliore la reconnaissance.

3.4. Expérience 3 : tests in situ

Les tests de la troisième expérience visent à examiner l'utilisation des Tactons dans des conditions d'utilisation courantes. Ces tests ont eu lieu pendant la fête de la science à l'Université Paul Verlaine – Metz. L'environnement était bruité et les utilisateurs étaient moins concentrés que dans les autres expériences car des personnes discutaient et se déplaçaient à proximité. Le panel d'utilisateurs était plus large que dans les autres expériences : la tranche d'âges est plus vaste (des enfants aux personnes âgées), et la plupart d'entre eux n'étaient pas informaticiens. Aucun d'entre eux n'a participé aux expériences précédentes, et aucun n'a déjà utilisé une matrice de picots. À cause du peu de temps qu'un utilisateur pouvait accorder à ces tests, les blocs de ces expérimentations ne contenaient que 50 Tactons, contre 100 dans les autres expériences. De plus chaque utilisateur n'a testé qu'un seul des ensemble de Tactons proposés. Nous avons sélectionné les 3 ensembles qui ont donné les plus faibles taux d'erreur jusqu'ici. 15 utilisateurs ont testé l'ensemble 3, 23 ont testé l'ensemble 4 et 15 ont testé l'ensemble 7.

Les hypothèses de cette expérience sont :

- Les utilisateurs vont faire plus d'erreurs et seront plus lents en condition réelle qu'en laboratoire. Cette hypothèse se justifie par le fait que les utilisateurs seront perturbés en conditions réelles et n'auront pas toujours l'attention nécessaire pour reconnaître efficacement les Tactons.

- Les utilisateurs obtiendront des résultats similaires entre les ensembles en situation réelle qu'en laboratoire, c'est-à-dire que les ensembles 4 et 7 auront tous les deux un faible taux d'erreur et seront reconnus rapidement, et que l'ensemble 3 sera moins efficace que les ensembles 4 et 7 avec cependant un taux d'erreur correct. Avec cette hypothèse nous supposons que tous les ensembles sont autant pénalisés en conditions réelles.

Résultats.

Les résultats sont présentés par les boîtes à moustaches de la figure 8. Deux utilisateurs ont un taux d'erreur au delà du seuil de chance (7 chances sur 8, soit 87,5%) avec l'ensemble 3 : 92% et 88%. Il s'avère que ces utilisateurs ont confondu le sens des Tactons, en inversant la direction : haut/bas, droite/gauche, etc. L'environnement était bruité, donc ces utilisateurs ont probablement mal compris les instructions. Leurs résultats doivent être écartés pour ne pas fausser les statistiques.

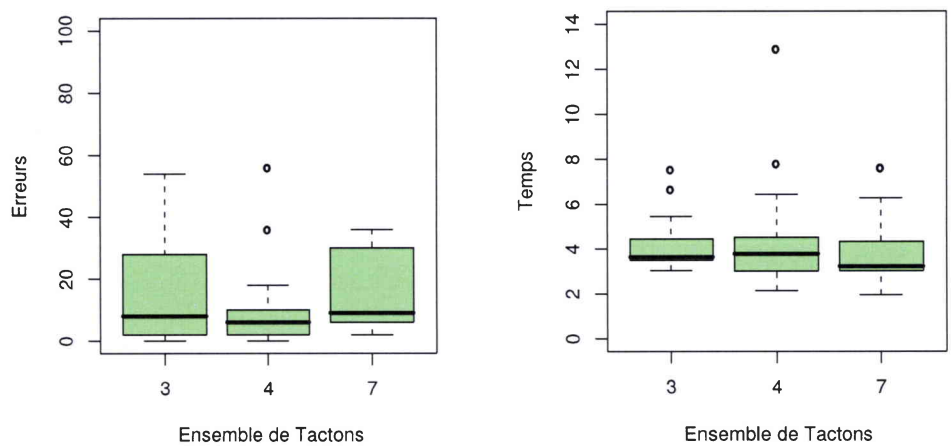


Figure 8 – Expérience 3 : erreurs et temps de réponse

L'ensemble 4 a encore une fois le taux d'erreur le plus faible, avec une médiane de 6% d'erreurs, contre 8% pour l'ensemble 3 et 10% pour l'ensemble 7. L'analyse de Kruskal-Wallis ne permet pas de trouver de différence significative entre les ensembles ($\chi^2 = 3{,}32, p = 0{,}1$). Un test de Mann-Whitney montre que les utilisateurs font significativement moins d'erreurs avec l'ensemble 4 en laboratoire qu'en situation réelle ($W = 206, p < 0{,}001$). On ne constate pas de telle différence pour l'ensemble 3 ($p = 0{,}1$) et l'ensemble 7 ($p = 0{,}5$). Le temps d'exploration des Tactons est équivalent pour les trois ensembles de Tactons : $3{,}71s$ pour l'ensemble 3, $3{,}77s$ pour l'ensemble 4 et $3{,}22s$ pour l'ensemble 7. Le test de Kruskal-Wallis confirme qu'il n'y a pas de différences significatives entre les temps de reconnaissance ($\chi^2 = 1{,}44, p = 0{,}4$). Par contre cette fois-ci le test de Mann-Whitney montre que les trois ensembles de Tactons testés dans cette expérience sont reconnus plus rapidement en laboratoire qu'en conditions réelles ($W = 104{,}5, p = 0{,}02$ pour l'ensemble 3 ; $W = 234, p < 0{,}001$ pour l'ensemble 4 ; et $W = 162, p < 0{,}001$ pour l'ensemble 7).

Discussion.

En comparant les résultats en laboratoire et en conditions réelles, on trouve une différence significative de taux d'erreurs que pour l'ensemble 4, et une différence significative en terme de temps d'exploration pour tous les ensembles de Tactons. La première hypothèse peut donc être acceptée pour l'ensemble 4. Dans cette expérience nous ne pouvons pas trouver de différence significative entre les ensembles, que ce soit en taux d'erreur ou en temps moyen de reconnaissance. En particulier l'ensemble 3 n'est pas plus mauvais que les ensembles 4 et 7. Nous ne pouvons donc pas accepter la seconde hypothèse. Les Tactons de l'ensemble 3 ont nécessité en moyenne 7 vagues pour être reconnus. Pour rappel, les utilisateurs de l'expérience 1 n'avaient besoin que de 5 vagues pour le même ensemble et les utilisateurs de l'expérience 2 avaient besoin de 7 vagues pour l'ensemble 5. Ceci renforce l'idée d'essayer d'autres vitesses à l'avenir afin que les utilisateurs puissent reconnaître les Tactons avec moins de vagues et donc réduire le temps d'exploration. De plus, les Tactons radiaux de l'ensemble 3 ont nécessité en moyenne 6 images alors que les diagonales en requéraient 7. Nous avons donc décidé de travailler plus en détails sur les Tactons en vagues, et en particulier les diagonales (section 3.5). En examinant les taux d'erreurs moyens entre les Tactons radiaux et les Tactons diagonaux, on s'aperçoit que les utilisateurs ont fait deux fois plus d'erreurs sur les diagonaux que sur les radiaux avec l'ensemble 7. Nous pouvons expliquer ce phénomène par le fait que les 4 Tactons diagonaux de cet ensemble utilisent la même forme et de la même Taille. La seule différence est la position, qui est difficile à juger à cause d'un manque de point de repère. Nous introduirons un point de référence dans l'expérience 5 (section 3.6) pour juger de l'efficacité de cette idée.

3.5. Expérience 4 : amélioration des vagues

Les résultats des Tactons en vagues précédents ne nous ont pas donné satisfaction, en particulier l'ensemble 5. Nous avons donc décidé d'essayer de nouveaux ensembles de Tactons à vagues. Pour cela nous avons conçu de nouvelles animations pour les diagonales. Afin de pouvoir comparer plus facilement les résultats de l'expérience 1, les mêmes utilisateurs ont testé ces nouveaux ensembles de Tactons dans les mêmes conditions.

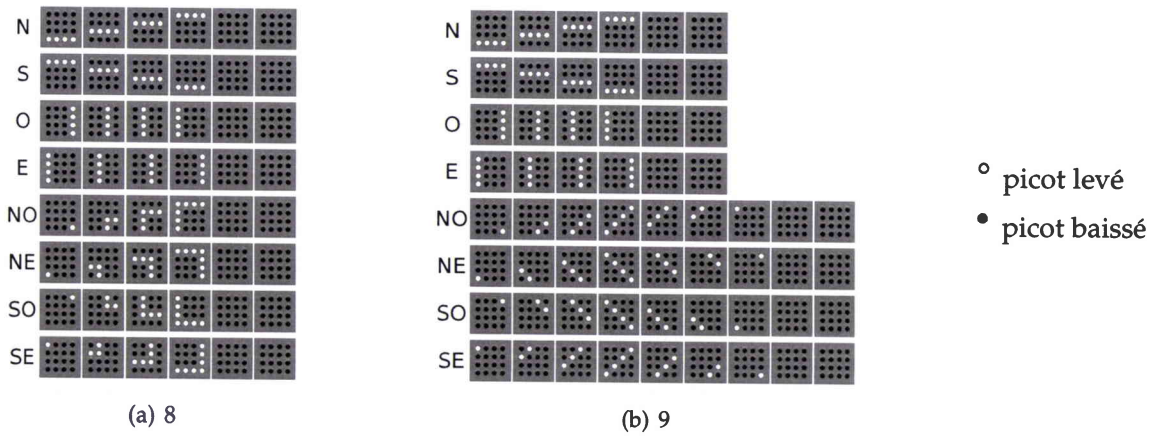


Figure 9 – Ensembles de Tactons utilisés dans l'expérience 4

La figure 9 présente les deux nouveaux ensembles de Tactons testés dans cette expérience. L'ensemble 8 (figure 9(a)) est une combinaison des ensembles 3 et 5 : il utilise les Tactons radiaux de l'ensemble 3 et les Tactons diagonaux de l'ensemble 5. L'idée est de créer un effet tactile différent entre les Tactons radiaux et diagonaux, contrairement à ce qui a été fait jusqu'ici. L'ensemble 9 est une modification de l'ensemble 3. Nous avons donc décidé d'utiliser toute la surface de la matrice pour déplacer la ligne en diagonale, en utilisant plus d'images. Cela signifie que l'animation des lignes qui se déplacent en diagonale est plus longue que celle des Tactons radiaux. Cette différence peut aider potentiellement les utilisateurs à différencier les Tactons radiaux des Tactons diagonaux.

Pour cette expérience, nous avançons les hypothèses ci-dessous :

- Les taux d'erreurs pour ces deux ensemble seront plus faibles que ceux de l'ensemble 3 dans l'expérience 1. En effet, nous pensons que la différence entre les Tactons radiaux et les Tactons diagonaux va aider à mieux identifier les directions.
- Les temps de réponse de ces deux nouveaux ensembles de Tactons sera inférieur à ceux de l'ensemble 3 dans l'expérience 1. En effet, comme nous pensons diminuer les ambiguïtés nous pensons que les utilisateurs hésiteront moins et répondront donc plus vite.

Résultats.

La figure 10 rassemble les taux d'erreurs et les temps moyens de réponse pour cette quatrième expérience, ainsi que les résultats obtenus avec l'ensemble 3 dans l'expérience 1 à des fins de comparaison. Les résultats donnent un taux d'erreur médian de 0% pour les ensembles 8 et 9, comparé à 3% pour l'ensemble 3 dans l'expérience 1. Le test de Kruskal-Wallis ne montre pas de différences significatives entre les 3 ensembles de Tactons ($\chi^2 = 4{,}67$, $p = 0{,}09$). Pour les temps de réponse, les utilisateurs ont fait une médiane de $2{,}74s$ pour l'ensemble 3 sur l'expérience 1. Dans cette expérience, ils ont exploré les Tactons de l'ensemble 8 en $2{,}52s$ et l'ensemble 9 en $2{,}19s$. Les utilisateurs avaient besoin de 5 vagues en moyenne pour identifier les directions pointées par les Tactons de l'ensemble 3, alors que seulement 4 vagues sont nécessaires en moyenne pour l'ensemble 8 et environ 3 pour l'ensemble 9[2]. Cependant le test de Kruskal-Wallis ne montre pas de différences significatives entre les temps de réponse des utilisateurs ($\chi^2 = 4{,}98$, $p = 0.08$).

Discussion.

La plupart des utilisateurs se sont sentis plus à l'aise avec ces nouveaux ensembles de Tactons. De plus les taux d'erreurs mesurés sont très faibles. Ces deux raisons nous permettent d'affirmer que nous avons obtenu des Tactons dynamiques satisfaisants. Cependant la différence avec les résultats de l'ensemble 3 n'est pas significative. Nous devons donc rejeter notre première hypothèse. De plus nous notons que quelques utilisateurs sont mal à l'aise avec ce genre de Tactons. Un d'entre eux ne supporte pas la sensation, et avoue avoir des difficultés à trouver le sens de déplacement des vagues. Il serait nécessaire de faire des tests plus approfondis en faisant varier d'autres paramètres pour permettre à ces utilisateurs d'apprécier ces Tactons. Une piste intéressante serait de modifier la vitesse de l'animation. Les temps de réponse de ces deux nouveaux ensembles sont statistiquement équivalents

[2]En prenant en compte le fait que certains Tactons ont 9 images et d'autres 6

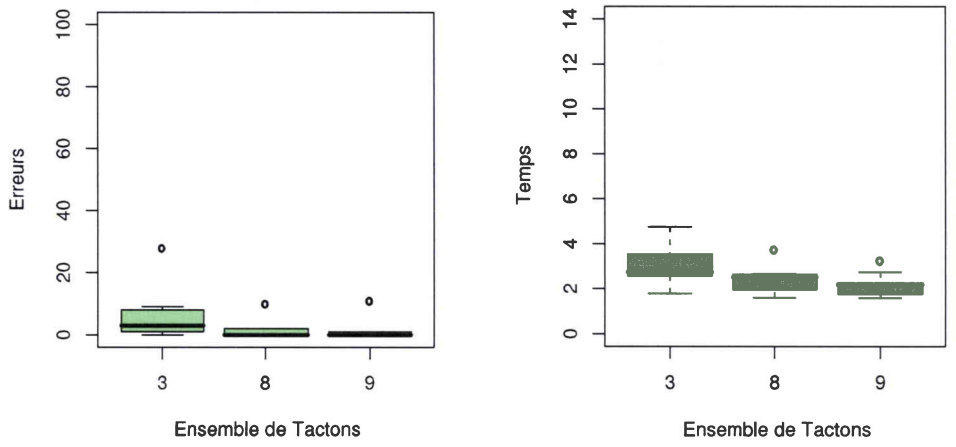


Figure 10 – Expérience 4 : erreurs et temps de réponse

à ceux de l'ensemble 3 de l'expérience 1. Nous n'avons donc pas amélioré le temps d'exploration et nous devons donc rejeter la seconde hypothèse.

3.6. Expérience 5 : Tactons mixtes

Un des problèmes majeurs pour la reconnaissance des directions selon les utilisateurs vient du manque de point de repères. En effet la direction est pointée par la position des picots levés sur la matrice. Or il n'est pas toujours évident de deviner cette position. De plus nous avons laissé de côté les Tactons clignotants. Maintenant que nous avons trouvé des Tactons statiques et en vagues satisfaisants, nous allons essayer d'utiliser un point de repère dans les Tactons clignotants pour améliorer la reconnaissance. Les mêmes utilisateurs que dans les expériences 1 et 4 se sont livrés à des tests sur 4 nouveaux ensembles de Tactons utilisant ce principe.

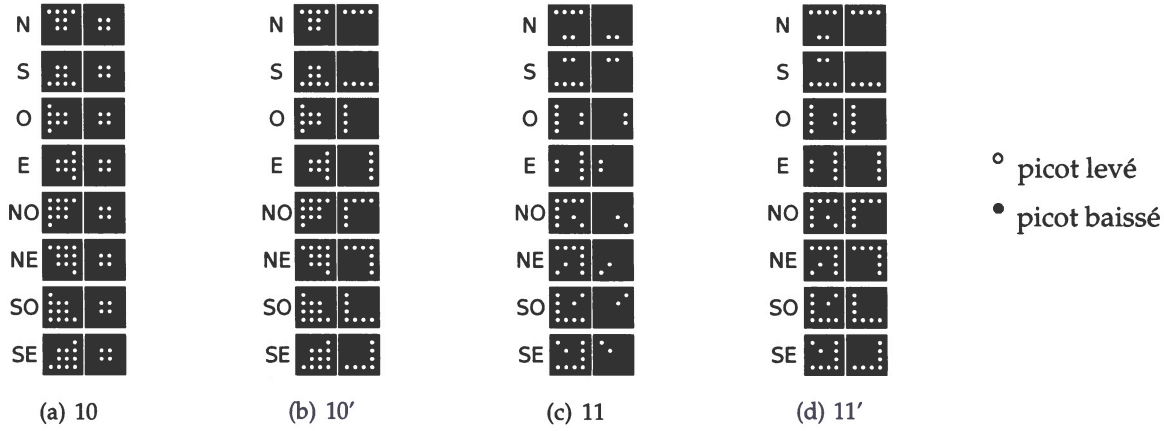


Figure 11 – Ensembles de Tactons utilisés dans l'expérience 5

Au sens de nos définitions ces Tactons sont des Tactons dynamiques. Nous les appelons mixtes dans le sens où ils ont une partie statique et une partie dynamique. La direction est marquée par les mêmes motifs que l'ensemble 4 des expériences précédentes, si ce n'est que les diagonales sont des coins de 4 picots de côté au lieu de 3. Deux types de points de référence ont été construits : soit un carré central avec les 4 picots du milieu de la matrice levés (ensembles 10, figure 11(a) et 10′, figure 11(b)) soit une direction opposée marquée par un motif des ensembles 1 et 2 (ensembles 11, figure 11(c) et 11′, figure 11(d)). Pour ces deux motifs de point de repère, deux techniques ont été testées. Soit le point de repère était fixe et la direction clignotait (ensembles 10 et 11), soit c'était la direction qui était fixe et le repère qui clignotait (ensembles 10′ et 11′).

Les hypothèses à vérifier dans cette expérience sont les suivantes :

- Les ensembles de Tactons avec le point de repère qui clignote auront des résultats significativement différents de ceux avec la direction qui clignote, en particulier en ce qui concerne le taux d'erreurs. Nous ne pouvons pas prévoir laquelle sera meilleure avant les tests. Cependant il est fort possible qu'une technique soit plus efficace que l'autre selon que l'attention soit plutôt portée sur les picots clignotants ou sur les picots fixes.
- Il y aura une différence significative entre les ensembles qui utilisent le motif central comme point de repère et les ensembles qui utilisent un motif directionnel antagoniste. Tout comme le mode de clignotement, nous ne pouvons pas prévoir lequel sera différent, mais suite à quelques tests informels nous pensons qu'une méthode sera plus efficace que l'autre.

Résultats.

Les taux d'erreur et les temps moyens de réponse sont représentés sur les graphiques de la figure 12. Les utilisateurs ont réalisé une médiane de 6% d'erreurs avec l'ensemble 10, 3% avec l'ensemble 10′, 3% avec l'ensemble 11, et 0% avec l'ensemble de Tactons 11′. L'analyse de Kruskal-Wallis permet de trouver une différence significative entre ces résultats ($\chi^2 = 15{,}53$, $p = 0{,}001$). En examinant ces résultats de plus près grâce à une comparaison des paires, on révèle que l'ensemble 11′ provoque significativement moins d'erreurs que les ensembles 10 ($p = 0{,}006$), 10′ ($p = 0{,}02$) et 11 ($p = 0{,}01$). Cette comparaison ne révèle pas d'autre paire significativement différente. Les temps d'exploration sont eux très similaires : on obtient une médiane de 2,56s pour l'ensemble 10, 2,39s pour l'ensemble 10′, 2,63s pour l'ensemble 11, et 1,99s pour l'ensemble 11′. D'ailleurs l'analyse de Kruskal-Wallis ne permet pas de déduire de différence significative entre ces temps de réponse ($\chi^2 = 3{,}58$, $p = 0{,}3$).

Discussion.

L'ensemble 11′ est le mieux reconnu de tous les ensembles. D'après les questionnaires il est aussi celui que les utilisateurs ont préféré : 5 utilisateurs sur 9 le préfèrent, contre 2 pour l'ensemble 11 et 2 pour l'ensemble 10′. En plus d'être celui qui provoque le plus d'erreurs en moyenne, l'ensemble 10 est le moins apprécié des utilisateurs. 7 utilisateurs sur 9 préfèrent les Tactons où le repère est clignotant à ceux où la direction clignote. Cependant les statistiques ne permettent pas d'affirmer que les Tactons à repère clignotant sont meilleurs que les Tactons à direction clignotante, notamment

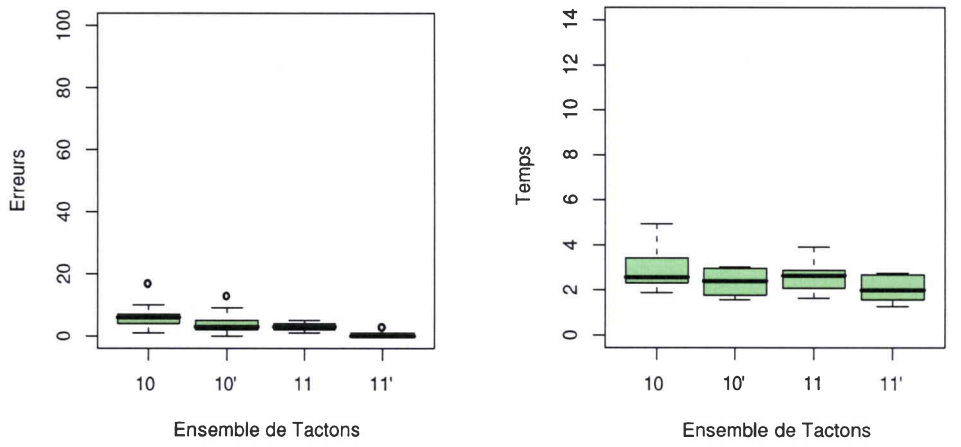


Figure 12 – Expérience 5 : erreurs et temps de réponse

à cause des résultats moyens de l'ensemble 10′. Ainsi nous rejetons la première hypothèse. De plus l'ensemble 11 ne se dégage pas assez des ensembles 10 et 10′ pour pouvoir conclure que les Tactons à repère antagoniste (ensembles 11 et 11′) sont meilleurs que les Tactons à repère central (ensembles 10 et 10′). Ainsi même si 7 utilisateurs sur 9 préfèrent ces Tactons, la seconde hypothèse est donc rejetée. Nous retiendrons que l'ensemble 11′ a obtenu des résultats très satisfaisants, à la hauteur de ceux des Tactons 4 pour les statiques et des Tactons 8 et 9 pour les vagues.

3.7. Conclusion

Les expériences réalisées ci-dessus ont permis de trouver des ensembles de Tactons satisfaisants, utilisant les différentes techniques proposées : Tactons statiques, Tactons en vagues et Tactons clignotants. La première expérience avait pour but de comparer des ensembles de Tactons utilisant ces trois techniques. Il est apparu que les utilisateurs ont préféré les Tactons statiques, et ont eu de bons résultats avec les Tactons statiques avec beaucoup de picots levés. Les Tactons en vagues ont des résultats moyens, mais encourageants. Un utilisateur n'aime cependant pas cette technique, et trouve ces Tactons désagréables. D'autres ensembles en vagues ont été testés dans une expérience ultérieure afin d'améliorer les résultats. Les Tactons clignotants quant à eux ont eu des résultats décevants. Nous avons donc décidé de les mettre de côté. Dans la deuxième expérience nous avons tenté d'utiliser des Tactons avec plus de picots, ce qui s'est avéré être un échec. En effet les formes étaient trop similaires et trop proches sur la matrice. De plus nous avons introduit des Tactons en vagues dont le motif change au cours du temps. Ceci a aussi été un échec, car les utilisateurs ont eu du mal à interpréter le changement de forme. L'ensemble de Tactons 4 déjà utilisé dans l'expérience précédente a confirmé ses bons résultats. Enfin un autre ensemble statique, similaire au précédent, a aussi obtenu de bons résultats. Dans la troisième expérience nous avons cherché à examiner l'utilisation de cette technique d'interaction avec un public plus varié et dans un environnement ressemblant à une utilisation de tous les jours. Les trois meilleurs ensembles découverts jusque là ont été examinés. Si on compare les résultats en laboratoire et les résultats en situation réelle : pour tous les ensembles le temps d'exploration est plus important en situation réelle, et l'ensemble

4 est plus dur à reconnaître. Une quatrième expérience a été menée pour trouver de meilleurs Tactons en vagues. Les résultats sont satisfaisants, même si les statistiques ne permettent pas de les distinguer des Tactons dynamiques précédents. L'utilisateur mal à l'aise avec les ensembles de Tactons en vagues précédents a toujours du mal avec ceux là. Des efforts devront être faits pour améliorer la sensation de ces Tactons. Enfin une cinquième expérience a eu pour but de reprendre contact avec les Tactons clignotants. Un point de référence a été introduit afin d'aider à résoudre les ambiguïtés. Deux modes ont été testés : soit la direction clignote et le point de référence est fixe, soit l'inverse. Cette fois-ci les résultats sont satisfaisants, tant par les taux d'erreurs, que par le temps d'exploration et la satisfaction des utilisateurs. Ainsi nous avons obtenu des taux d'erreurs et des temps d'exploration satisfaisants pour chaque technique.

Dans la section suivante nous allons explorer d'autres dimensions, en faisant varier des paramètres tels que l'unité temporelle, qui était fixée dans la série d'expériences précédente. Ainsi cette fois nous allons coder plusieurs informations sur chaque Tacton.

4. Les Tactons multi-dimensionnels

La section précédente présente des expériences utilisant des Tactons n'ayant qu'une seule dimension. En l'occurrence, ces Tactons utilisent cette dimension pour représenter une direction. Dans cette section nous allons nous intéresser à des Tactons utilisant plusieurs dimensions, ce qui nous permet de transmettre potentiellement plus d'informations à l'utilisateur. Cette étude a été réalisée en collaboration avec l'Université de Glasgow dans le cadre du projet Micole [136].

Nous allons utiliser des Tactons clignotants dans cette expérience. À l'instar des expériences précédentes, la forme des Tactons sera une dimension de ces Tactons. Les deux autres dimensions considérées sont la vitesse et la taille, qui vont varier respectivement l'unité temporelle et le nombre de picots utilisés pour chaque Tacton. Augmenter la quantité d'information dans le signal permet de fournir potentiellement plus d'informations à l'utilisateur. Mais en contrepartie cela augmente la complexité du signal, et donc cela risque d'augmenter le taux d'erreurs. L'expérience décrite dans cette section a pour but de déterminer si les bénéfices tirés de l'augmentation de la quantité d'information compense la perte induite par l'augmentation de la difficulté de la tâche.

Ces Tactons ont été créés afin d'aider un utilisateur à naviguer dans des informations non visuelles ou des données complexes. Il est possible que la tâche entraîne un stress pour l'utilisateur dans la mesure où il est difficile de remplacer une exploration visuelle par une exploration non-visuelle. Ceci est particulièrement vrai dans le cas d'exploration de grandes quantités de données ou quand les relations spatiales entre les objets ont un sens. La stimulation tactile pourrait être une solution pour aider par exemples les utilisateurs à localiser des objets déjà parcourus, ou afficher des informations sur un objet tout en laissant le canal auditif libre pour des informations sonores ou de la parole. C'est pourquoi nous cherchons à augmenter la quantité d'information transmise à l'utilisateur.

Le concept de transmission d'information a été utilisé pour mesurer la performance des utilisateurs à interpréter les Tactons. Miller décrit comment la transmission d'information peut être utilisée pour décrire la qualité d'un canal de communication en examinant la quantité potentielle de messages pouvant être lus à travers le canal et la corrélation entre les entrées et les réponses des utilisateurs [115]. L'avantage d'utiliser la transmission d'information comme métrique est que cela fournit une méthode de comparaison des canaux de communication sans unité.

4.1. Conception

Le développement de Tactons commence par le choix de la structure des informations à afficher. En effet pour chaque dimension souhaitée, il convient d'utiliser des paramètres de Tactons adéquats. En particulier il faut prendre soin de choisir des paramètres qui n'interfèrent pas afin que les utilisateurs puissent percevoir l'information que l'on souhaite véhiculer.

Le choix des paramètres pour créer des Tactons est varié. On peut par exemple utiliser la forme du motif, la taille du motif, le mouvement du motif, l'unité temporelle d'un Tacton dynamique, ou encore utiliser différents rythmes en variant les durées d'images. Cependant, tous ne sont pas susceptibles d'être utiles. Les expériences précédentes décrites dans la section 3 ont montré par exemple que certains utilisateurs peuvent être mal à l'aise avec des motifs qui se déplacent, ou qu'il est difficile d'interpréter les Tactons qui ont peu de picots levés. Les résultats de ces expériences seront utilisés ici afin d'utiliser la forme du motif comme paramètre pour représenter la dimension de direction. Les deux autres dimensions utilisées sont la taille qui joue sur la quantité de picots levés, ainsi que la vitesse qui joue sur l'unité temporelle des Tactons dynamiques.

Les motifs intéressants ont déjà été trouvés dans les expériences précédentes. Des tests pilotes ont été réalisés afin de trouver de bonnes valeurs de taille et de vitesse, avec une analyse de la transmission d'information.

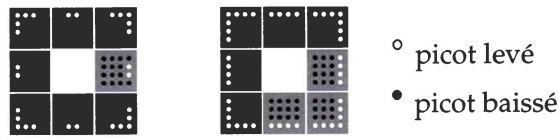
Direction.

Les formes de motifs utilisées dans cette expérience sont similaires à ceux des ensembles ayant obtenu de bons résultats dans la section 3. Nous représentons toujours 8 direction basées sur les points cardinaux : Nord, Sud, Est, Ouest, Nord-Est, Nord-Ouest, Sud-Est et Sud-Ouest. Les ensembles de Tactons présentés ici utilisent les mêmes formes que les Tactons 4 des expériences précédentes : des lignes pour les Tactons radiaux et des angles pour les Tactons diagonaux. Le fait d'utiliser des formes similaires permet de comparer ces résultats avec ceux obtenus avec les Tactons mono-dimensionnels. Ainsi nous pourrons évaluer l'influence sur la performance de l'ajout d'information présentée à l'utilisateur. La dimension de direction dans cette étude est donc définie par $Direction = \{N, S, E, O, NE, NO, SE, SO\}$, et le paramètre associé est le motif des Tactons.

Taille.

Étant donné les formes sélectionnées, la taille des matrices de la VTPlayer (4×4) et les résultats précédents, il n'est possible de réaliser que deux tailles. La plus petite taille correspond à un

rétrécissement de l'ensemble 4 de la section 3 (figure 13(a)) : les diagonales sont laissées intactes et les Tactons radiaux sont rétrécis et n'utilisent que la moitié de la matrice. La plus grande correspond à une élongation de ces mêmes Tactons (figure 13(b)) : cette fois-ci les Tactons radiaux sont les mêmes et les Tactons diagonaux utilisent toute la longueur de la matrice.



(a) Petits (b) Grands

Figure 13 – Formes de Tactons utilisés pour les Tactons multi-dimensionnels

Une étude pilote a été réalisée avec quatre utilisateurs pour décider si la taille était paramètre crédible pour la suite de nos études. Les participants ont pu différencier les grands et les petits Tactons avec peu d'entraînement avec un taux de reconnaissance de plus de 90% pour 96 essais chacun. Ces résultats montrent qu'il est justifié d'essayer d'utiliser ce paramètre pour coder de l'information. La dimension créé ici est la taille et comprend deux valeurs : $Taille = \{Petit, Grand\}$. Elle est encodée avec un paramètre taille qui fait varier le nombre de picots levés.

Vitesse.

Lors du choix de la troisième dimension, en plus de considérer la vitesse de clignotement nous avons considéré plusieurs rythmes différents. Nous ne cherchons pas à faire différencier les rythmes, ceci pourra être l'objet d'une étude future. Nous cherchons à savoir si un rythme particulier est plus propice à la différenciation de différentes vitesses de clignotement. En notant d_l la durée de l'image dans laquelle les picots sont levés et d_b la durée de l'image dans laquelle les picots sont baissés, nous pouvons choisir des valeurs telles que :

- $d_l = d_b$
- d_l varie et d_b est constant
- d_l est constant et d_b varie
- $d_l + d_b$ est constant

Nous avons réalisé une étude pilote pour découvrir quel rythme était le plus propice pour permettre aux utilisateurs de différencier des valeurs de vitesse. Quatre participants ont participé à ces tests, en essayant chaque condition avec 7 unités temporelles entre $40ms$ et $500ms$ par image. Dans chaque série, chaque paire de stimulus possible était répétée quatre fois. Les utilisateurs devaient dire si les deux était les mêmes ou si ils étaient différents. Le seul paramètre variant dans les Tactons présentés était la vitesse et un seul Tacton était affiché à la fois. L'analyse montre des résultats similaires pour les quatre rythmes. Le premier ($d_l = d_b$) est donc choisi comme rythme dans les études futures vu qu'aucune préférence n'a été suggérée par les utilisateurs. De plus comme ce rythme est celui utilisé dans la section 3, cela permet d'effectuer d'éventuelles comparaisons.

Le choix des valeurs de vitesses à conserver est plus délicat. Nous devons faire un compromis entre la longueur des messages et la différence entre les valeurs proposées. Un message trop long va ralentir l'interprétation des messages par l'utilisateur. Par contre des valeurs trop proches peuvent conduire à des erreurs d'interprétation. Les résultats montrent qu'il n'y a pas de confusion parmi 5 des 7 valeurs choisies. Cependant comme les utilisateurs vont juger la durée d'une image de manière relative, nous allons utiliser les résultats de Brown *et al.* [31] qui suggèrent que différencier trois valeurs d'un paramètre de leurs Tactons vibrotactiles qui ne varient que relativement est difficile. Nous n'allons donc pas utiliser plus de 3 vitesses, et vérifier si nous obtenons des résultats similaires. La dimension créée est la vitesse et prend donc trois valeurs : $Vitesse = \{Lent, Moyen, Rapide\}$. Le paramètre utilisé pour agir sur cette dimension est l'unité temporelle τ des Tactons dynamiques (définition 5). En effet le rythme est fixé à une durée de 1 pour toutes les images. Les valeurs d'unités temporelles pour les vitesses *Lent*, *Moyen* et *Rapide* sont respectivement $40ms$, $200ms$ et $500ms$.

4.2. Méthodologie

Cette expérience a été conduite avec 20 utilisateurs voyants, qui ont été recrutés sur la liste de diffusion de l'Université de Glasgow. Les participants sont âgés de 18 à 29 ans. Les quatre gauchers présents ont été répartis équitablement dans les deux groupes d'utilisateurs. Cette étude se fait aussi avec des utilisateurs voyants afin d'avoir un aperçu général de la performance d'un utilisateur avec cette technique d'interaction.

Dans cette expérience, les utilisateurs ont ressenti le stimulus avec l'index de leur main non-dominante. Cette fois la main non-dominante a été utilisée car nous destinons cette technique d'interaction à un contexte bi-manuel. Le PHANToM sera utilisé comme périphérique de pointage dans la main dominante, et la souris VTPlayer sera utilisée comme afficheur à picots dans la seconde main. Le but est d'examiner la performance des utilisateurs à capter l'information contenue dans des Tactons à trois dimensions. Les dimensions utilisées sont celles découvertes ci-dessus : la direction, la taille et la vitesse. Les participants portaient un casque diffusant un son neutre afin d'éviter toute information sonore du périphérique. Cette précaution n'était pas nécessaire dans l'expérience précédente car d'une part les Tactons statiques ne provoquent pas de bruit, et d'autre part les Tactons dynamiques utilisaient tous la même vitesse et donc le bruit produit par la matrice de la souris VTPlayer était identique pour chaque Tacton.

Deux conditions ont été testées dans cette expérience : avec deux valeurs de vitesse (S_2) et avec trois valeurs de vitesse (S_3). Les utilisateurs ont été répartis aléatoirement et équitablement entre les deux groupes. Cependant en ajoutant une vitesse supplémentaire, il y a plus de Tactons dans l'ensemble du groupe S_3 que dans le groupe S_2. Nous avons donc un choix à faire pour que les tâches effectuées par les deux groupes soient équivalentes. Il y a trois possibilités :

- un nombre d'essais équivalent dans les deux conditions.
- un nombre équivalent de répétition de chaque Tacton dans chaque conditions.
- tester un sous-ensemble des Tactons dans S_3.

Si nous choisissons un nombre de répétitions équivalent pour chaque Tacton dans chaque condition, le test de la condition S_3 sera plus long que celui de la condition S_2. Il en résulte qu'il y a un risque de différence de fatigue et d'apprentissage entre les deux groupes d'utilisateurs. Par contre si nous restreignons les tests de S_3 à un sous-ensemble des valeurs possibles, cela va compliquer l'analyse des résultats et en particulier la comparaison avec les autres ensembles de Tactons. Nous choisissons donc de maintenir un nombre équivalent d'essais dans chaque condition. L'ensemble de Tactons utilisé dans S_2 contient 32 Tactons (8 directions $\times$ 2 tailles $\times$ 2 vitesses), et celui utilisé dans S_3 contient 48 Tactons (8 directions $\times$ 2 tailles $\times$ 3 vitesses). Les utilisateurs ont donc exploré 96 Tactons, ce qui fait que chaque Tacton de S_2 a été exploré 3 fois et chaque Tacton de S_3 a été exploré 2 fois.

Les utilisateurs devaient placer leur main non-dominante sur la souris VTPlayer et presser une touche du clavier avec leur main dominante pour ressentir le Tacton. Une fois la touche relâchée le Tacton était stoppé et ils devaient donner leur réponse verbalement à l'expérimentateur. Pour la dimension direction, les réponses possibles étaient *nord*, *sud*, *est*, *ouest*, *nord − est*, *nord − ouest*, *sud − est* et *sud − ouest*. Les réponses possibles pour la taille était *petit* et *grand*, et celles pour la vitesse étaient *rapide*, *moyen* ou *lent*. La réponse est considérée comme fausse si au moins une des trois parties de la réponse est fausse. Le temps mesuré est le temps que le Tacton est ressenti, c'est-à-dire la durée pendant laquelle la touche du clavier était enfoncée. Le temps d'exposition du Tacton a été limitée à $10s$.

Afin de mesurer le profit potentiel de l'utilisation des multiples paramètres du signal, nous utiliserons l'information transmise par Tacton comme métrique pour comparer les performances de S_2 et S_3 avec le meilleur ensemble de Tactons dans l'expérience précédente (ensemble 4). Les résultats de l'expérience précédente seront notés T_4.

Dans cette expérience, nous formulons les hypothèses que :
- Les participants de S_2 feront significativement moins d'erreurs d'identification. Ceci est dû au fait que les utilisateurs peuvent trouver plus facile de différencier deux vitesses que trois.
- Les participants de S_2 auront besoin de significativement moins de temps pour examiner les Tactons que ceux de S_3. Ceci pour la même raison que pour la première hypothèse.
- Il y aura significativement plus d'information transférée dans S_3 que dans S_2. En effet, même si les utilisateurs risquent de faire plus d'erreurs dans S_3, il y a potentiellement plus d'informations à transférer dans S_3 que dans S_2.
- Dans les deux cas il y aura significativement plus d'informations transmises aux utilisateurs que dans le meilleur cas des Tactons mono-dimensionnelles décrit dans la section 3.

4.3. Résultats

Une fois encore nous avons utilisé des statistiques non-paramétriques pour analyser les résultats. Le test de Mann-Whitney a été utilisé pour tester les différences significatives entre les groupes. La comparaison entre les groupes pour chaque dimension a été réalisée par des tests pairés de Wilcoxon.

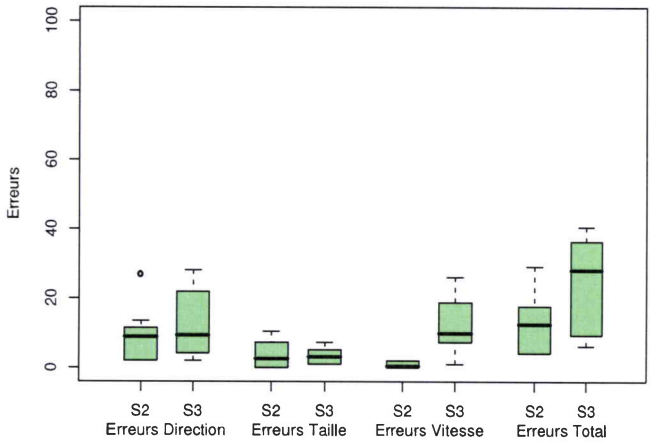


Figure 14 – Erreurs de reconnaissance des dimensions pour chaque condition

S_2 contre S_3.

Les résultats sont représentés sur la figure 14. Le taux d'erreur médian dans la condition S_2 est de 12,5%, et 28,12% dans la condition S_3. Cette différence est significative ($W = 135{,}0$, $p < 0{,}03$). En examinant de plus près au niveau des erreurs par dimension, cette différence a l'air d'être due à l'écart de reconnaissance de la vitesse. Les utilisateurs de S_3 ont réalisé une médiane de 9,9% d'erreurs, contre 0,52% pour ceux de S_2. Cette différence est significative selon le test de Mann-Whitney ($W = 61{,}5$, $p < 0{,}01$). Aucune différence n'a été constatée pour les dimensions de direction ($W = 97{,}5$, $p = 0{,}60$) et de taille ($W = 101{,}5$, $p = 0{,}82$). Le temps médian d'exploration des Tactons est de $2{,}09s$ dans S_2 et $2{,}31s$ dans S_3. Il n'y a pas de différences significatives entre ces temps d'exploration ($W = 107{,}0$, $p = 0{,}91$).

Les analyses de transmission d'information sont résumées par les boîtes à moustache de la figure 15. Il apparaît qu'une médiane de 5,0 bits d'information sont transmis dans S_3, comparé à 4,6 bits dans S_2. Le test de Mann-Whitney permet de montrer une différence significative entre les deux conditions ($W = 75{,}0$, $p < 0{,}03$). En comparant avec la médiane de 2,88 bits d'information transmise par les meilleurs Tactons du meilleur ensemble de l'expérience précédente, on trouve une différence significative aussi bien par rapport à S_2 ($W = 153{,}0$, $p < 0{,}01$) que par rapport à S_3 ($W = 153{,}0$, $p < 0{,}01$).

Interactions significatives entre les dimensions.

La taille a un effet significatif sur la performance des utilisateurs, que ce soit dans S_2 ou dans S_3. Les grands motifs étaient significativement plus faciles à identifier que les plus petits. En effet il y avait significativement plus d'erreurs de reconnaissance de petits Tactons dans S_2 ($W = 0{,}0$, $p < 0{,}01$) et S_3 ($W = 0{,}0$, $p < 0{,}01$). Le pourcentage médian de petits Tactons mal interprétés dans S_3 est de 28,13%, pour 15,63% avec les grands Tactons. Parallèlement, dans S_2 on obtient 16,67% pour les petits Tactons et 5,21% pour les grands. En analysant les résultats au niveau des dimensions, on remarque que cet écart de résultat est dû à une différence significative sur la dimension de direction

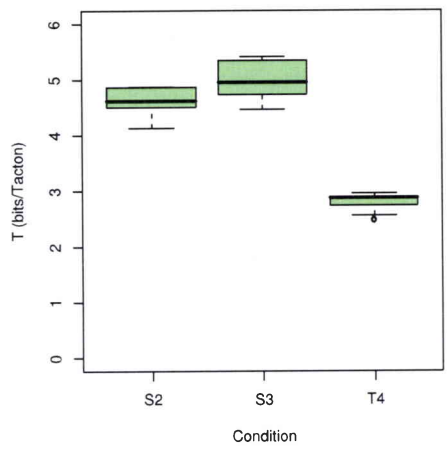


Figure 15 – Transmission d'information pour les conditions S_2, S_3 et T_4

dans S_2 ($W = 0{,}0$, $p < 0{,}01$) et S_3 ($W = 0{,}0$, $p < 0{,}01$). Dans S_3, les taux d'erreur médians sont respectivement de $2{,}08\%$ et $16{,}67\%$ pour les tailles *grand* et *petit*. De même, dans S_2, les taux d'erreur médians sont de $2{,}08\%$ pour les grands tactons et $14{,}58\%$ pour les petits.

Les utilisateurs ont aussi requis significativement plus de temps pour explorer les petits Tactons que les grands, que ce soit dans S_2 ($W = 6{,}0$, $p < 0{,}04$) ou dans S_3 ($W = 6{,}0$, $p < 0{,}04$). Pour S_3 les participants ont ressenti les Tactons pendant respectivement $1{,}94s$ et $2{,}27s$ avant de répondre pour les grands Tactons et les petits. De manière analogue dans S_2 les utilisateurs ont ressenti les Tactons pendant $1{,}97s$ et $2{,}62s$ pour reconnaître les grands et les petits Tactons.

4.4. Discussion et conclusion

En préambule nous pouvons constater que la méthode utilisée pour analyser les résultats affecte le point de vue sur les résultats. Si nous regardons les taux d'erreur, il y a significativement moins d'erreurs lorsque l'on n'utilise que deux vitesses par rapport à quand on utilise trois vitesses. Ceci est prévisiblement dû à une plus grande difficulté à distinguer les valeurs de la dimension vitesse. Ce résultat est intéressant et dans un sens confirme les résultats de Brown *et al.* [31]. Les remarques des participants sur les unités temporelles des Tactons confirment les résultats de Brown qui révèlent que l'ajout d'une troisième valeur pour une comparaison relative peut augmenter significativement la complexité de la tâche pour l'utilisateur. Ces données suggèrent qu'il est préférable de n'utiliser que deux vitesses.

Si on examine les données ainsi, on ignore les gains obtenus par l'ajout d'un niveau supplémentaire pour cette dimension. En effet cette valeur supplémentaire permet de transférer plus d'informations à l'utilisateur grâce à un seul Tacton. Ce gain est révélé par les résultats de transmission d'informations, qui montrent que la quantité d'informations transmises à l'utilisateur est significativement plus importante, malgré ce surplus d'erreurs. Il est cependant important de prendre des précautions lors de la création d'interface car le coût de ce surplus d'erreur peut être élevé.

Les interactions significatives entre la direction et la taille suggèrent que ces dimensions ne sont peut être pas indépendantes sur une petite matrice tactile. Ceci est peut être dû au fait que ces deux dimensions utilisent le même paramètre : le motif tactile. La performance des utilisateurs montre que les petits Tactons sont significativement plus difficiles à distinguer que les grands. Ceci confirme la difficulté à identifier les Tactons à peu de picots levés identifiée dans la section 3. Lorsque nous choisissons les dimensions pour des Tactons multi-dimensionnels, celles-ci ne devraient pas interférer. Cependant réduire la taille d'un motif affecte la perception des formes par les utilisateurs dans les deux conditions S_2 et S_3.

Nous avons montré par l'exemple qu'il est possible de coder plusieurs informations à l'aide de nos Tactons. Dans la section suivante nous allons utiliser des Tactons multi-dimensionnels dans un système de guidage permettant d'explorer des formes géométriques.

5. Exploration de formes géométriques

Les Tactons que nous avons conçus peuvent servir dans de nombreux contextes. Nous avons choisi de représenter des directions car cela permet de créer des systèmes de guidage, et ainsi aider un utilisateur à explorer des environnements de manière non visuelle. Une partie de nos Tactons ont été utilisés par Crossan et Brewster pour guider un utilisateur dans un labyrinthe [45]. Ils ont utilisé des Tactons mono-dimensionnels, avec 4 directions. L'exploration était réalisée à l'aide d'un PHANToM. Ils ont comparé l'utilisation de Tactons statiques et de Tactons dynamiques en vagues. Les utilisateurs étaient plus efficaces avec les Tactons statiques. Nous avons décidé d'étudier en collaboration avec cette équipe l'exploration de formes géométriques à l'aide d'une souris VTPlayer.

5.1. Modes d'exploration

L'exploration de formes géométriques de manière non visuelle n'est pas une tâche facile. La méthode que nous utilisons utilise le retour tactile. Il a été montré par plusieurs études que la simple traduction des pixels en picots n'est pas efficace [55, 84]. Ceci est dû à plusieurs problèmes, dont le fait que le sens tactile ne permet en général de n'avoir qu'un aperçu local de la forme à explorer. D'autre part les déficients visuels utilisent leurs dix doigts pour explorer les schémas sur papier en relief, alors que sur les systèmes interactifs ils ne peuvent utiliser généralement qu'un ou deux doigts.

Méthodes existantes.

Nous avons vu dans la section 3.3.3 du chapitre I une approche proposée par Ziat *et al.* [179, 178, 180, 181]. Ils utilisent une technique de zoom afin de pouvoir explorer les formes à plusieurs échelles. La méthode utilisée effectue aussi une simple transcription des pixels, mais contourne le problème du point de vue local. Cependant selon l'échelle les détails des objets peuvent disparaître, et rien ne garantit que l'utilisateur explorera les détails une fois qu'il aura changé le niveau de zoom.

Petit *et al.* utilisent une autre approche consistant à utiliser des textures [134]. En variant les effets haptiques, il est possible de créer une simili-coloration qui aide à créer une plus grande variété de détails. Malheureusement ce genre de textures n'est pas réalisable avec autant de précision avec des matrices à picots de taille réduite. Nous allons donc proposer une autre approche, utilisant les matrices de picots de la souris VTPlayer.

Notre approche.

Notre approche consiste à guider l'utilisateur sur la forme en utilisant les Tactons directionnels testés dans les sections précédentes. À titre d'études préliminaires, des étudiants de Master ont étudié ce problème et implémenté des systèmes d'exploration de formes géométriques utilisant les Tactons. Ces études n'ont pas été menées suffisamment minutieusement pour obtenir des statistiques précises. D'autres études sont restées au stade expérimental [127], mais elles montrent cependant que la tâche est possible. De plus ces systèmes ne donnaient que des informations de direction.

Dans cette nouvelle étude nous avons choisi de décomposer la forme en segments. À l'instar des prototypes précédents nous aurons besoin d'informations pour coder les directions, et nous allons ajouter des informations de distance. Nous avons donc décidé d'utiliser les Tactons clignotants à deux dimensions conçus dans la section précédente. Huit motifs issus des expériences précédentes (figure 16) sont utilisés pour afficher la direction. Le Tacton indiquant la direction est toujours affiché sur l'index. Le système adapte l'affichage en fonction de la main utilisée pour tenir la souris VTPlayer. Le changement de main peut être dû à deux facteurs : soit l'utilisateur est gaucher, soit l'exploration se fait à l'aide d'un autre dispositif de pointage.

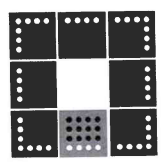


Figure 16 – Motifs utilisés pour l'exploration de formes géométriques

Trois vitesses de clignotement[3] sont utilisées pour représenter les longueurs. Nous avions deux choix pour la représentation des distances :

- **Représentation absolue :** les changements de vitesse sont répartis sur des longueurs fixées en pixels. Cette méthode permet de créer une métaphore de la règle graduée, et permet donc à des utilisateurs de mesurer les formes explorées en reconnaissant les vitesses ressenties. Cependant le nombre de valeurs utilisées est restreint, ce qui d'une part complique la mesure des formes avec précision, et d'autre part implique que l'ordre de grandeur des graduations doit être adapté à la taille des formes.
- **Représentation relative :** chaque segment est divisé en trois parties égales. Chaque portion du segment parcouru possède sa vitesse de clignotement. La mesure d'un segment se fait donc en

[3]Les valeurs d'unités temporelles de $50ms$ $100ms$ et $200ms$ ont été retenues suite à des tests pilotes.

interprétant la vitesse et le temps de déplacement par l'utilisateur. L'utilisateur doit donc pouvoir estimer sa vitesse de déplacement ou se déplacer toujours à la même vitesse afin de pouvoir comparer la taille des segments. Cette tâche est d'autant plus ardue qu'il n'est pas certain que la perception de la vitesse soit la même selon la direction du mouvement de la main.

Nous avons décidé d'utiliser la première méthode pour représenter les distances car suite à des tests informels, elle nous a semblé plus adéquate. Pour le choix de représentation, nous avons décidé d'utiliser la vitesse rapide pour les petites distances et la vitesse lente pour les grandes distances. Ce choix est motivé par une analogie avec un sonar. Nous avons adapté l'échelle d'espacement des valeurs à la taille des formes afin d'optimiser les bénéfices apportés par cette technique.

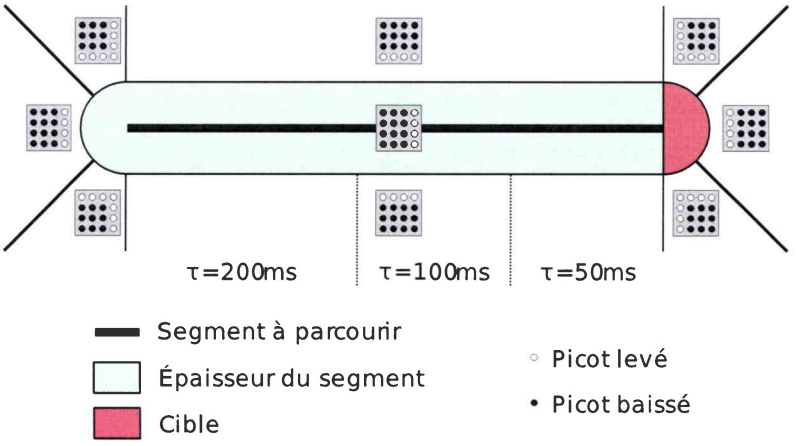


Figure 17 – Guidage sur un segment

Pour chaque segment nous guidons l'utilisateur de point en point de la manière indiquée sur la figure 17 :

- lorsque l'utilisateur se trouve sur le segment[4], la direction indiquée pointe vers la cible, c'est-à-dire l'extrémité suivante du segment en cours. La distance représentée par la vitesse de clignotement indique la distance entre le curseur et la cible. Suite aux tests pilotes nous avons fixé les deux seuils pour les vitesses à 50 et 100 pixels, sachant que la longueur des segments des formes retenues (figure 18) varie entre 75 et 212 pixels.
- lorsque l'utilisateur est sur la cible[5], le segment suivant est déclenché. La direction affichée pointe donc vers la cible suivante. La distance représentée est la distance jusqu'à la cible suivante.
- lorsque l'utilisateur est en dehors de la forme, tous les picots de la matrice sous le majeur sont levés pour signaler à l'utilisateur qu'il est en dehors de la forme. La direction pointée par le Tacton est perpendiculaire au segment si le curseur est sur les côtés, ou désigne l'extrémité du segment la plus proche dans le cas contraire (figure 17). Cette fois la distance correspond à la plus petite distance entre le curseur et le segment.

[4] L'épaisseur a été fixée à 20 pixels par des tests pilotes.
[5] Demi-cercle au bout du segment dont le rayon est égal à l'épaisseur du segment (figure 17).

Ce mode d'exploration impose de parcourir la scène par composante connexe. Dans le cas où il y aurait plusieurs formes géométriques, il serait possible de passer d'une forme à l'autre à l'aide d'une commande telle que la pression d'une touche. Le problème qui se poserait alors serait la capacité de repérer les objets les uns par rapport aux autres, notamment pour les intersections. À l'inverse il est plus facile que dans le cas de la transcription à partir du visuel de séparer les composantes d'une scène. Cependant dans cette étude nous ne nous intéressons qu'à l'exploration d'une seule forme.

Notons que notre technique requiert une version vectorielle de la scène à explorer. Ce n'est pas gênant dans notre cas car les formes que nous souhaitons explorer sont simples. Dans une étude future, nous pourrons chercher un système permettant d'extraire une trajectoire de guidage à partir d'un schéma en bitmap. L'utilisation de nos Tactons restreint les formes utilisables car nous n'utilisons que 8 directions. Avec ces directions nous excluons les angles intermédiaires et les courbes. À défaut ces dernières peuvent être polygonalisées. Nous pourrons étudier de nouveaux Tactons et ainsi explorer des formes plus complexes dans des travaux futurs. Pour le moment nous étudions la faisabilité d'un tel système avec des Tactons multi-dimensionels.

5.2. Méthodologie

Huit utilisateurs voyants occultés ont participé à cette expérience. Ils sont âgés de 26 à 28 ans, et aucun ne souffre de problèmes connus de perception tactile. Tous sont habitués à utiliser un système informatique, mais aucun n'utilise de tablette graphique couramment. Un des participants est gaucher, les commandes étaient donc inversées pour qu'il explore avec sa main dominante, et que les indications de direction se trouvent sur son index.

Conditions.

Nous avons d'une part voulu examiner la différence entre l'exploration à l'aide de Tactons et l'exploration en utilisant la transcription du visuel au tactile. D'autre part nous avons souhaité tester deux modes d'exploration :

- soit avec la souris VTPlayer comme dans l'étude de Jansson sur l'exploration de cartes géographiques [84]. L'exploration et le retour tactile se fait donc sur le même périphérique et donc sur la même main.
- soit avec une tablette graphique tel que dans l'étude de Wall et Brewster sur l'exploration de camemberts [171]. L'utilisateur utilise donc la tablette graphique avec sa main dominante pour explorer et reçoit le retour tactile sur la souris VTPlayer qui se trouve dans sa main non-dominante. L'intérêt de la tablette graphique est de pouvoir remédier au déplacement relatif de la souris qui est problématique pour une interaction non-visuelle.

En combinant ces 2 facteurs nous obtenons 4 conditions :

- **Pt :** exploration avec la tablette graphique, représentation à l'aide des Tactons.
- **Pv :** exploration avec la tablette graphique, représentation traduite du visuel.
- **Mt :** exploration avec la souris VTPlayer, représentation à l'aide des Tactons.
- **Mv :** exploration avec la souris VTPlayer, représentation traduite du visuel.

Paramètres.

Nous avons conduit des tests pilotes pour déterminer les paramètres de l'application de test. Comme nous l'avons déjà évoqué, lors de cette phase nous avons réglé l'épaisseur des segments (20 pixels), les vitesses de clignotement ($50ms$, $100ms$ et $200ms$) et les distances de clignotement (50 et 100 pixels). Nous avons aussi déterminé la taille des formes (entre 75 et 212 pixels par segment), sachant que la tablette graphique était utilisée en absolu, et que la résolution d'écran était de 1920×1200 pixels. Par contre nous avons remarqué que l'exploration en mode traduction visuelle vers tactile était trop rapide. En effet si le curseur se déplace trop vite d'une part il est difficile de suivre le contour de la forme. D'autre part si le curseur se déplace trop vite au dessus d'une ligne, il possible que rien ne se soit affiché. Ceci est dû aux limites de la fréquence de rafraîchissement de la matrice de picots. Nous avons donc ralenti le curseur d'un facteur 3.

Nous avons mesuré le temps d'exploration nécessaire aux utilisateurs pour parcourir la plupart des formes. En effet comme nous souhaitons faire explorer le même nombre de formes à tous les utilisateurs, nous devons arrêter l'exploration si elle dure trop longtemps. Nous nous sommes arrêtés sur un compromis de 3 minutes par forme, ce qui laisse *a priori* suffisamment de temps aux utilisateurs pour explorer correctement les formes.

Nous avons aussi fixé le nombre de formes à explorer à 20 afin de limiter la fatigue des utilisateurs. Ceci fixé, il reste à décider combien de conditions seraient effectuées par utilisateur. Si chaque utilisateur n'avait fait qu'une condition, il aurait fallu beaucoup d'utilisateurs afin de pouvoir comparer les résultats par condition. Au contraire si chaque utilisateur avait effectué les 4 conditions alors il aurait dû y avoir 4 apprentissages, ce qui aurait pris beaucoup de temps. De plus ceci n'aurait donné que 5 formes par condition par utilisateur, ce qui aurait fait peu de données. Nous avons donc choisi le compromis de 2 conditions par utilisateurs, et donc 10 formes par condition par utilisateur.

Groupes.

2 conditions parmi 4 donnent 12 combinaisons, ce qui fait que nous aurions besoin d'un nombre d'utilisateurs multiple de 12, ce qui est très difficile en pratique. Nous avons donc choisi de limiter les combinaisons en faisant en sorte que les utilisateurs utilisent toujours le même mode d'exploration (tablette graphique ou souris VTPlayer). Nous avons donc divisé les utilisateurs en 4 groupes, expérimentant les conditions suivantes :

- Pt puis Pv
- Pv puis Pt
- Mt puis Mv
- Mv puis Mt

Formes.

Nous avons conçu 25 formes géométriques (figure 18). Nous avons conçu plus de formes que nécessaire car d'une part il faut une forme supplémentaire pour chaque session d'apprentissage, et d'autre part il faut plus de formes pour éviter que les utilisateurs ne répondent par élimination. Les

dimensions des formes sont du même ordre de grandeur. Les formes 1, 6 et 7 sont les seules ne possédant pas de lignes diagonales, et la forme 10 est la seule n'ayant ni de lignes horizontales ni de lignes verticales. Pour chaque forme, plusieurs orientations sont proposées afin de détecter les éventuelles confusions d'orientation.

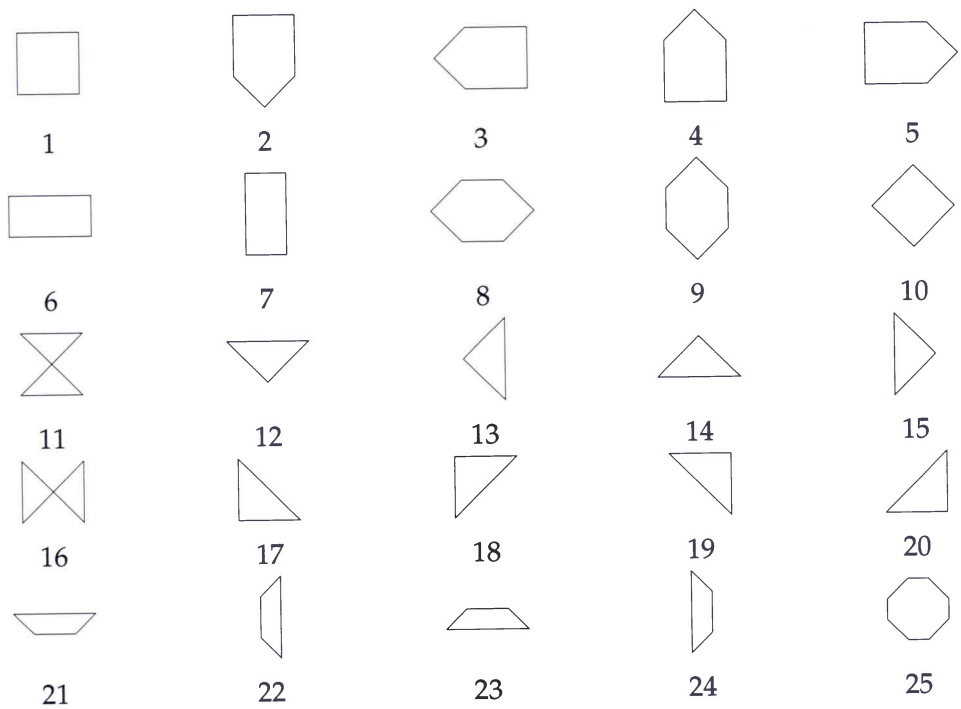


Figure 18 – Formes proposées

Déroulement.

Chaque utilisateur doit effectuer deux blocs de 10 formes à explorer. Les formes explorées par chaque utilisateur ont été tirées aléatoirement, mais d'une part de manière à ce que chaque forme ne puisse pas être explorée plusieurs fois par le même utilisateur. D'autre part le nombre d'utilisateurs explorant chaque forme a été uniformisé afin de pouvoir faire des statistiques sur les résultats par forme (6 ou 7 utilisateurs par forme). Le participant utilise le même périphérique pour l'exploration dans les deux blocs : soit la tablette graphique, soit la souris VTPlayer selon le groupe auquel il appartient. Avant chaque bloc, l'expérimentateur explique la technique d'interaction à l'utilisateur. Celui-ci peut explorer librement une forme à titre d'apprentissage, et ce aussi longtemps qu'il le souhaite. Il a pour instruction de se familiariser avec la technique d'interaction et de trouver une méthode d'exploration qui lui convient. Une fois l'utilisateur prêt, il peut commencer le bloc de 10 formes. Pour chaque forme, le logiciel enregistre la trajectoire, le temps d'exploration, la configuration affichée sur la VTPlayer à chaque instant, et la vitesse de clignotement dans le cas de l'exploration par Tacton. Les trajectoires des erreurs sont reproduites en annexe B. L'application s'arrête automatiquement au bout de 3 minutes si l'utilisateur n'a pas répondu avant. L'expérimentateur note la réponse de l'utilisateur,

ainsi qu'un indice de confiance entre 1 (incertain) et 7 (certain). À la fin des tests, l'expérimentateur recueille les remarques de l'utilisateur et discute des erreurs commises par l'utilisateur. Celui n'est pas informé de ses résultats après chaque forme afin de ne pas influer sur les réponses suivantes.

Hypothèses.

Ces expériences visent à comparer à la fois les performances du guidage à l'aide de Tactons à ceux de la transcription des pixels, et les performances de l'exploration avec la tablette graphique à ceux de l'exploration à l'aide de la souris VTPlayer. Nous allons donc vérifier les hypothèses suivantes :

1. Les utilisateurs utilisant la tablette graphique auront de meilleurs résultats que ceux utilisant la souris VTPlayer pour explorer les formes. En effet nous pensons que les utilisateurs répondront plus rapidement, avec moins d'erreurs, et seront plus confiants en leur réponse. Cette hypothèse est motivée d'une part par le fait que la tablette fonctionne en absolu, ce qui est plus pratique pour une exploration non-visuelle. D'autre part la souris VTPlayer n'est pas précise car le capteur de position est placé à l'arrière. Il arrive donc que les utilisateurs se déplacent en biais.
2. Le guidage à l'aide de Tactons est plus efficace que la transcription des pixels. En outre nous prévoyons que les utilisateurs répondront plus rapidement, feront moins d'erreurs et seront plus confiants en leur réponse. Les raisons principales de cette hypothèse sont que le guidage par Tactons impose d'explorer la forme en entier et il est possible à tout moment de la retrouver, ce qui n'est pas le cas de l'autre mode.
3. Il n'y aura pas de différence de résultats entre les formes. En effet nous avons volontairement choisi des formes basiques afin de mesurer l'impact de la technique d'exploration et la technique de présentation.
4. Il n'y aura pas de différence de résultats entre les utilisateurs. En effet les utilisateurs ont été sélectionnés selon le même profil d'âge et d'utilisation des outils informatiques afin de limiter l'effet utilisateur.
5. Les utilisateurs seront plus à l'aise lorsqu'ils répondront juste. En effet nous pensons qu'ils répondront plus rapidement et donneront un indice de confiance plus élevé quand leur réponse sera correcte. À l'inverse les erreurs seront commises lorsque les utilisateurs auront des doutes.

5.3. Résultats

Résultats par condition.

La figure 19 montre les résultats par condition. Les trois graphiques représentent la quantité d'erreurs, les indices de confiance et le temps d'exploration par condition. Premièrement on remarque que les utilisateurs ayant utilisé la tablette graphique (Pt et Pv) ont fait plus d'erreurs et étaient moins confiants que ceux utilisant la souris VTPlayer pour explorer (Mt et Mv). Un test de la somme des rangs de Wilcoxon montre une différence de confiance significative entre les deux modes d'exploration ($W = 2576, p = 0{,}02$). Par contre on n'observe pas de différence entre l'exploration par guidage avec des Tactons et l'exploration par transcription du visuel au tactile ($W = 3453, p = 0{,}3$). Deuxièmement on remarque que les utilisateurs sont plus confiants dans les conditions où ils ont

fait peu d'erreurs, et inversement. Troisièmement on remarque que les utilisateurs utilisaient plus de temps pour répondre avec la représentatin transcrite à partir des pixels qu'avec le guidage par Tacton. Un test de Kruskal Wallis sur les 4 conditions montre une différence significative entre les conditions ($\chi^2 = 16{,}5, p < 0{,}001$). La comparaison par paires à l'aide de tests de Wilcoxon révèle que le condition Pv est significativement plus lente que les conditions Mt ($p = 0{,}008$) et Pt ($p = 0{,}009$). La différence est toujours significative si on joint les résultats de Pt à ceux de Mt, et ceux de Pv à ceux de Mv ($W = 2066, p < 0{,}001$). Par contre nous ne trouvons pas de différences significatives entre la réunion des conditions Pt et Pv et la réunion des conditions Mt et Mv ($W = 3370, p = 0{,}5$).

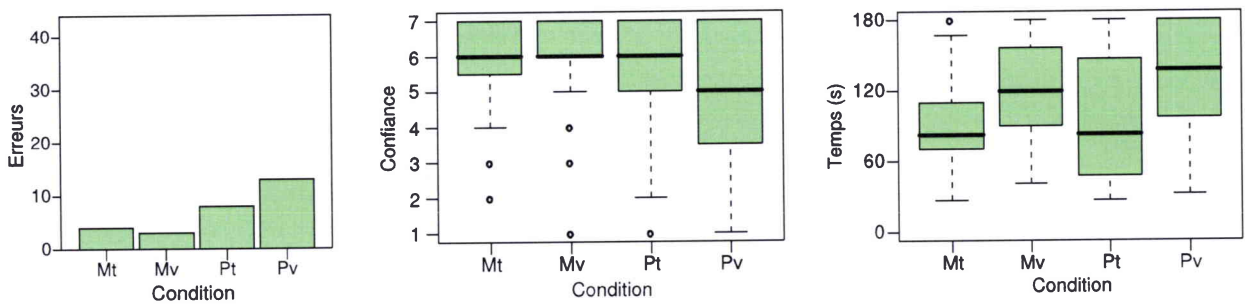


Figure 19 – Erreurs, confiance et temps d'exploration par condition

En examinant de plus près les erreurs par condition et par utilisateur, on remarque que les erreurs sont concentrées sur certains utilisateurs (figure 20). 7 des 13 erreurs en condition Pv ont été réalisées par l'utilisateur 4 et 5 des 8 erreurs en condition Pt ont été commises par l'utilisateur 3. Par contre 5 utilisateurs on fait peu d'erreurs : l'utilisateur 8 n'en a fait aucune, les utilisateurs 1, 5 et 6 en ont fait deux, et l'utilisateur 7 en a fait trois.

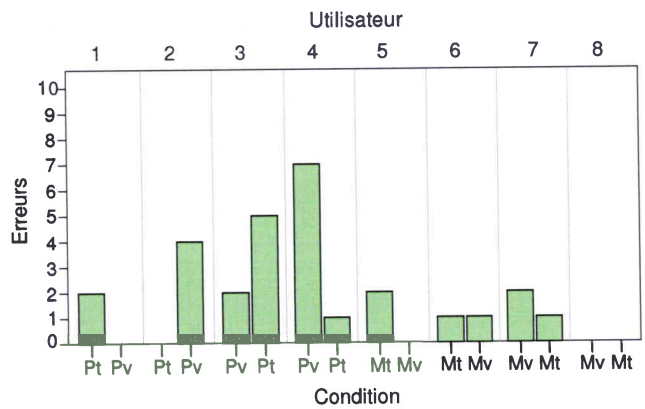


Figure 20 – Erreurs par utilisateur par condition

Résultats par forme.

Chaque forme a été parcourue par 6 ou 7 utilisateurs. La forme ayant posé le plus de difficultés est la forme 16 : 4 utilisateurs sur 7 se sont trompés, l'indice de confiance médian est de $4/7$ et le temps d'exploration médian est de $180s$. Les 4 utilisateurs ayant donné une mauvaise réponse sont ceux qui ont donné les indices de confiance les plus faibles (1, 1, 3 et 4), et ont tous utilisé le temps maximal de 3 minutes pour explorer la forme. Les tests de Kruskal-Wallis ne montrent cependant pas de différences significatives entre les formes, que ce soit pour les indices de confiance ($\chi^2 = 22{,}9$, $p = 0{,}5$) ou pour les temps d'exploration ($\chi^2 = 15{,}5$, $p = 0{,}9$). La taille des formes est légèrement différente de l'une à l'autre. En effet selon la forme, le nombre de pixels utilisé varie entre 478 et 720. Cependant si nous comparons les rapports obtenus en divisant le nombre de pixels par le temps d'exploration, nous n'obtenons toujours pas de différences significatives ($\chi^2 = 27{,}2$, $p = 0{,}2$).

Résultats par utilisateur.

Les résultats des utilisateurs sont très hétérogènes. Comme nous avons pu le voir précédemment, certains utilisateurs ont fait très peu d'erreurs, et d'autres ont eu des difficultés (figure 20). L'analyse des indices de confiance par utilisateurs reportée sur la figure 21 est intéressante. Elle révèle que certains utilisateurs sont très confiants en leur réponse : les utilisateurs 1, 6 et 8 ont majoritairement donné des indices de 6 ou 7 alors que l'utilisateur 3 a donné majoritairement des indices entre 3 et 7. Cet utilisateur est l'un de ceux qui ont donné le plus de mauvaises réponses. Un test de Kruskal-Wallis montre des différences significatives entre les utilisateurs ($\chi^2 = 30{,}6$, $p < 0{,}001$). La comparaison par paires à l'aide de tests de Wilcoxon montre que les indices de confiance des utilisateurs 1, 6 et 8 sont significativement plus élevés que ceux des utilisateurs 2 ($p = 0{,}006$, $p = 0{,}02$ et $p = 0{,}03$) et 5 ($p = 0{,}007$, $p = 0{,}02$ et $p = 0{,}04$).

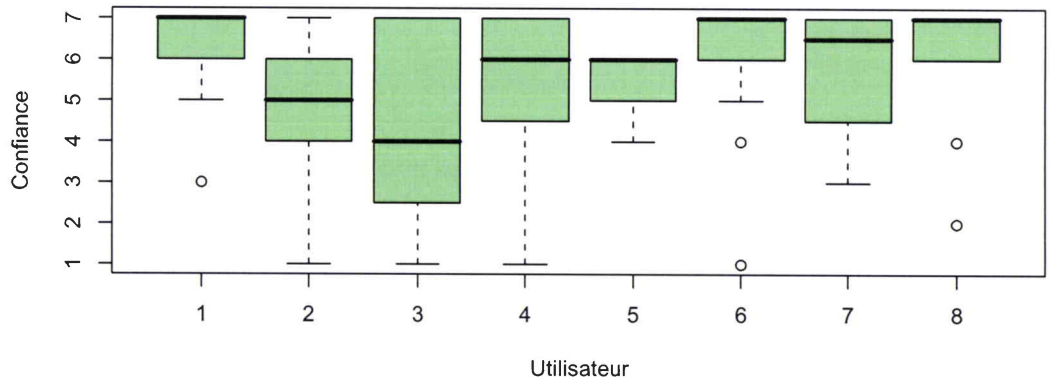


Figure 21 – Indice de confiance par utilisateur

L'analyse des temps de réponses présentés sur la figure 22 est tout aussi intéressante. L'utilisateur 1 est très rapide, avec un temps médian de réponse de $45s$, alors que les utilisateurs 2, 5, 7 et 8 n'ont jamais répondu en si peu de temps. À l'inverse l'utilisateur 3 a pris beaucoup de temps : sa médiane est de $180s$, c'est-à-dire le temps maximal d'exploration. Il s'agit d'un des deux utilisateurs ayant

fait le plus d'erreurs. Nous pouvons remarquer aussi que l'utilisateur 8 fait partie des utilisateurs les plus certains de leurs réponses, n'a fait aucune erreur, et pourtant a un temps d'exploration médian légèrement supérieur à celui de l'ensemble des utilisateurs ($107s$ contre $102s$). Un test de Kruskal-Wallis montre qu'il y a des différences significatives entre les temps d'exploration des utilisateurs ($\chi^2 = 59{,}1$, $p < 0{,}001$). Les comparaisons par paires montrent que l'utilisateur 1 a exploré les formes significativement plus rapidement que tous les utilisateurs ($p < 0{,}001$ pour chaque utilisateur), sauf l'utilisateur 6 ($p = 0{,}2$). L'utilisateur 6 est significativement plus rapide que les utilisateurs 2 ($p = 0{,}01$), 3 ($p < 0{,}001$), 7 ($p = 0{,}01$) et 8 ($p = 0{,}03$). Enfin l'utilisateur 3 est significativement plus lent que les utilisateurs 1 ($p < 0{,}001$), 2 ($p = 0{,}01$), 5 ($p = 0{,}002$), 6 ($p < 0{,}001$) et 8 ($p = 0{,}01$).

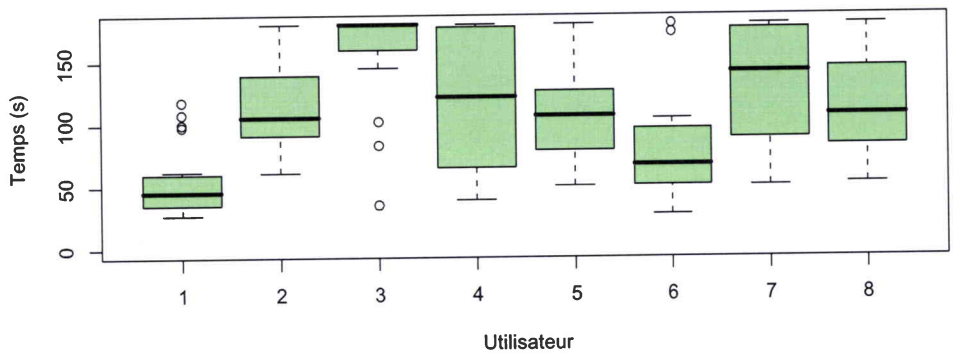


Figure 22 – Temps d'exploration par utilisateur

Réponses justes/réponses fausses.

L'analyse ci-dessous consiste à comparer les indices de confiances et les temps d'exploration quand les utilisateurs ont répondu correctement par rapport à quand ils ont répondu faux. Le graphique de gauche de la figure 23 montre les indices de confiance selon la réponse des utilisateurs. Lorsque la réponse est juste, les utilisateurs ont donné un indice de confiance médian de 5,99, alors qu'il n'est que de 3,71 lorsque la réponse est fausse. La médiane sur l'ensemble des réponses est de 5,59. Nous avons pu trouver une différence significative entre les indices de confiance des réponses justes et des réponses fausses grâce à un test de Wilcoxon ($W = 3110$, $p < 0{,}001$).

Sur le graphique de droite de la figure 23, nous pouvons observer les temps de réponse selon l'exactitude de la réponse donnée. Le temps de réponse médian pour les réponses correctes est de $101s$, alors qu'il est de $148s$ pour les réponses erronées. Sur l'ensemble des explorations, les utilisateurs ont pris un temps médian de $109s$. Le test de Wilcoxon révèle une différence significative entre les temps d'exploration lorsque les réponses sont correctes et lorsqu'elles sont erronées ($W = 849{,}5$, $p < 0{,}001$).

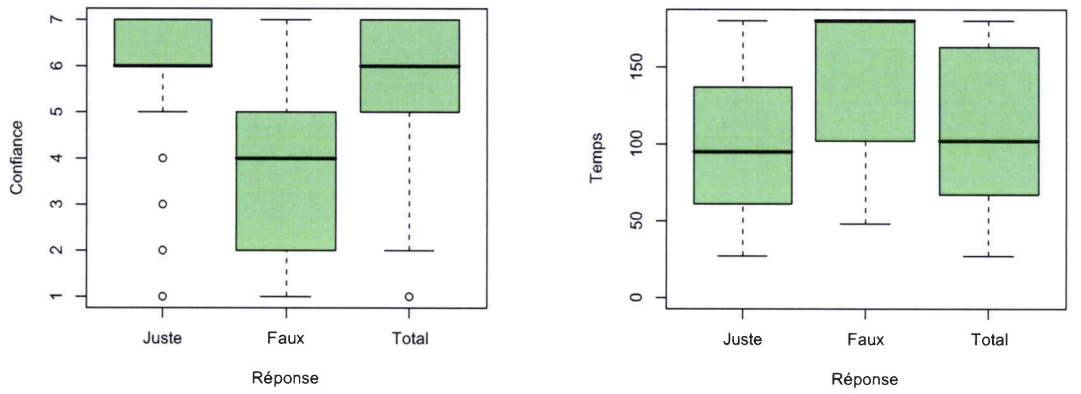


Figure 23 – Confiance et temps d'exploration par réponse

5.4. Discussion

Tablette graphique contre souris VTPlayer.

L'étude comparative par condition nous a donné des informations tant sur la comparaison entre les deux modes d'exploration que sur les deux modes de présentation. Concernant les modes d'exploration, les utilisateurs on réalisé 26% d'erreurs avec la tablette graphique, et 9% avec la souris VTPlayer. Nous n'avons pas relevé de différences significatives en ce qui concerne le temps de réponse, mais par contre la confiance en la réponse donnée est significativement plus importante avec l'exploration par la souris VTPlayer. En s'appuyant sur les erreurs et la confiance en la réponse, nous rejetons la première hypothèse selon laquelle l'exploration avec la tablette graphique serait plus efficace. Au contraire les résultats sont meilleurs en explorant les formes avec la souris VTPlayer. Ceci est surprenant vis-à-vis des résultats de Jansson et Pedersen [84], et tend à montrer que l'exploration à l'aide de la souris VTPlayer n'est pas responsable des mauvais résultats des utilisateurs dans leurs expériences.

Nous avions songé à utiliser la tablette graphique afin d'avoir un déplacement absolu pour aider l'utilisateur en navigation non visuel. Cependant il s'avère que cette technique a posé plusieurs problèmes. Le premier est que lorsque les utilisateurs levaient la main (pour explorer plus loin par exemple), ils avaient des difficultés à se repositionner au même endroit. Ce problème est particulièrement handicapant en condition Pv car il n'y a pas de système de guidage permettant de retrouver simplement la forme. Le deuxième problème vient de l'exploration bi-manuelle. En effet certains utilisateurs étaient perturbés en condition Pv par l'exploration avec la main dominante et le ressenti sur l'autre main. Cela les a perturbés pour comprendre la technique d'interaction.

Tactons contre transcription du visuel.

La comparaison des modes de présentation nous a aussi donné quelques éléments. Les taux d'erreurs sont proches : les utilisateurs ont réalisé 15% d'erreurs avec le guidage par Tacton, contre 20% avec la transcription du visuel au tactile. Les études comparatives statistiques entre les indices

de confiance ne permettent pas de dégager de différences significatives. Cependant l'étude des temps d'exploration montre que les utilisateurs étaient significativement plus rapides en condition Tacton qu'en condition transcription visuelle. En effet les utilisateurs ont exploré les formes avec une médiane de 94*s* en condition Tacton, et 124*s* en condition transcription visuelle. Nous devons cependant relativiser ces résultats suite aux remarques de certains utilisateurs. En effet ils nous ont signalé que les formes semblaient plus grandes en condition transcription visuelle. Or ces remarques sont justifiées par le fait que nous avons ralenti le curseur en condition transcription visuelle afin de rendre le système utilisable. Nous aurions dû réduire la taille des formes du même facteur afin que les formes paraissent de la même taille dans les deux conditions. Nous ne jugeons donc pas ces résultats suffisants pour accepter l'hypothèse 2 selon laquelle la condition Tacton serait plus efficace que l'autre méthode pour explorer les formes.

Nous pouvons ajouter que les préférences des utilisateurs sur les deux méthodes sont partagées. Cependant parmi les utilisateurs ayant préféré la méthode traditionnelle de transcription des pixels, plusieurs ont souligné que le problème majeur était l'adaptation. En effet il y a beaucoup d'informations à assimiler d'un coup. Certains de ces utilisateurs ont d'ailleurs avoué qu'ils préfèreraient probablement la méthode utilisant les Tactons une fois qu'ils se seraient habitués à la technique d'interaction. Il serait donc avisé d'introduire les informations petit à petit à l'avenir.

Résultats par forme.

Les résultats par forme explorée ne permettent pas de trouver de différence significative. En effet les statistiques que nous avons effectuées ne permettent pas de trouver de différences significatives entre les indices de confiance donnés par les utilisateurs ou entre les temps d'exploration. La forme n'est donc pas une source de problèmes déterminante, et nous pouvons donc accepter la troisième hypothèse prévoyant que les résultats par forme serait similaires.

Il est à noter que l'exploration du carré (forme 1) et des deux rectangles (formes 6 et 7) ont révélé que les utilisateurs se servaient peu des informations de distance indiquées par la vitesse de clignotement. L'utilisateur 2 a été surpris lorsqu'il devait donner sa réponse car il ne s'attendait pas à ce qu'il y ait à la fois un carré et des rectangles. Il a répondu juste, mais avec un indice de confiance de 2/7. Il a fait plus attention sur la forme de même type suivante, et a répondu correctement avec un indice de confiance de 6/7 cette fois. L'utilisateur 5 quant à lui a répondu 1 au lieu de 7, avec un indice de confiance de 5/7. Il est possible de remédier à ce problème en introduisant les informations progressivement, tel que nous l'avons suggéré dans le paragraphe précédent.

Résultats par utilisateur.

L'étude des résultats entre utilisateurs révèle que certains n'ont pas eu difficultés à explorer les formes, alors que d'autres ont trouvé la tâche difficile. En effet un utilisateur n'a pas fait d'erreurs et trois autres n'en ont fait que 2, alors qu'un utilisateur en a fait 8 et un autre en a fait 7. Ces différences se traduisent aussi au niveau de la confiance en leur réponse. Nous avons trouvé des différences significatives entre les indices de confiance donnés par les utilisateurs. Les

trois utilisateurs ayant réalisé le moins d'erreurs sont significativement plus confiants que deux autres utilisateur. Par contre un de ces deux utilisateurs fait partie des utilisateurs ayant fait le moins d'erreurs avec seulement 2 erreurs. Il se trouve que cet utilisateur n'a jamais donné d'indice de confiance de 7. Après un entretien avec lui, il semble qu'il souhaitait se donner une marge d'erreur.

Nous avons aussi trouvé des différences significatives entre les temps de réponse par utilisateur. En effet un des utilisateurs était significativement plus rapide que les autres, et un autre était significativement plus lent que les autres. Nous avons aussi remarqué que l'utilisateur n'ayant fait aucune erreur et ayant des indices de confiance élevés a pris autant de temps que l'ensemble des utilisateurs pour explorer les formes. Après un entretien avec lui, nous avons appris que cet utilisateur préférait prendre son temps pour être certain de ses réponses, ce qui a été bénéfique en terme de reconnaissance et de certitude. Un autre utilisateur était significativement plus rapide qu'une partie des autres utilisateurs. Il a avoué qu'il aurait pu répondre encore plus rapidement dans plusieurs cas mais préférait assurer ses réponses. Cet utilisateur a en effet fait peu d'erreurs et a donné des indices de confiance élevés.

Au vu de ces résultats, il y a visiblement des différences entre les résultats des différents utilisateurs. Nous devons donc rejeter la quatrième hypothèse. Nous prévoyons d'effectuer ce genre d'expérience sur un plus grand nombre d'utilisateurs afin de limiter l'effet utilisateur. Bien que l'apprentissage ait été réalisé avec soin, certains utilisateurs ont eu du mal à trouver une technique d'exploration efficace. Par exemple l'utilisateur 4, qui a fait beaucoup d'erreurs en condition Pv, a alterné plusieurs techniques d'exploration sans s'attacher à l'une d'entre elle. La figure 24 montre d'une part à gauche l'exploration de la forme 21 par l'utilisateur 4, qui a répondu 12, et d'autre part à droite l'exploration de cette même forme par l'utilisateur 2 en même condition, et qui a donné la bonne réponse. La comparaison de ces deux trajectoires montre bien que l'utilisateur 2 a trouvé une stratégie efficace pour suivre le contour de la forme, alors que ce n'est pas le cas de l'utilisateur 4. Il serait donc intéressant d'examiner l'impact de l'apprentissage sur les résultats des utilisateurs.

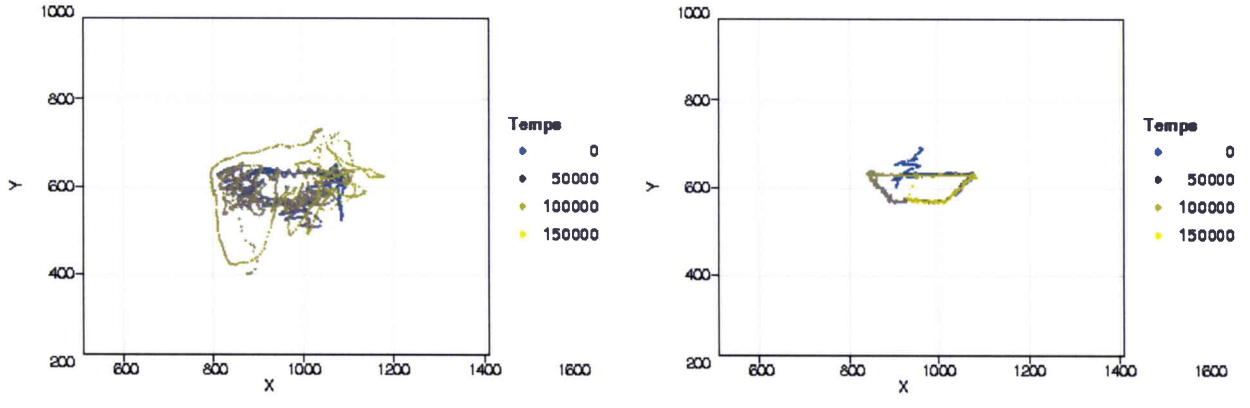


Figure 24 – Deux stratégies d'exploration de la forme 21 en condition Pv

Résultats par réponse.

Le dernier point important de cette étude concerne les résultats des utilisateurs selon si leur réponse est juste ou fausse. Dans un premier temps l'étude des indices de confiance montre que les utilisateurs sont significativement plus confiants en leur réponse quand ils ont donné la bonne réponse que quand ils se sont trompés. En effet lorsque leur réponse est erronée, les utilisateurs ont un indice de confiance médian de 3,71 alors qu'il est de 5,99 lorsque la réponse est correcte. En ce qui concerne les temps de réponse, les utilisateurs ont répondu significativement plus rapidement lorsque leur réponse était correcte que lorsqu'elle était erronée. Le temps de réponse médian pour les bonnes réponses est de $101s$, alors qu'il est de $148s$ pour les mauvaises réponses. Ces résultats montrent que les utilisateurs étaient plus à l'aise lorsqu'ils ont répondu correctement, et au contraire doutaient fortement lorsqu'ils se sont trompés. Ceci explique aussi les différences de résultats entre utilisateurs. Ceux qui ont rapidement compris les techniques d'interaction et qui ont rapidement trouvé une méthode d'exploration ont fait peu d'erreurs, avec des indices de confiance élevés, et ont répondu rapidement. Ces méthodes d'exploration font partie des stratégies analysées plus en détails par d'autres auteurs [67, 179, 6, 7]. Ces méthodes sont le suivi continu, le micro balayage et le nano balayage. Pour plus de détails l'auteur pourra se référer aux études susmentionnées.

5.5. Conclusion

Nous avons proposé une nouvelle méthode de présentation de formes géométriques, basée sur un guidage utilisant des Tactons multi-dimensionnels. Nous avons comparé cette méthode à une méthode plus traditionnelle consistant à lever les picots correspondant aux pixels sombres autour du curseur. D'après nos résultats, aucune méthode ne se démarque de l'autre. Nous avons donc proposé des améliorations à notre méthode, grâce auxquelles nous espérons obtenir de meilleurs résultats. Nous avons aussi constaté que les utilisateurs ont été débordés par la quantité d'instructions à assimiler. En effet notre méthode n'étant pas intuitive, plusieurs utilisateurs ont avoué qu'un temps d'adaptation était nécessaire pour exploiter pleinement les informations fournies par les Tactons. En particulier les informations de distance ont été moins utilisées que les informations de direction.

Nous avons aussi comparé deux méthodes d'exploration : avec une tablette graphique et avec une souris VTPlayer. En effet certaines études déconseillent l'usage de la souris VTPlayer [84] et d'autres suggèrent d'utiliser une tablette graphique [171]. Cependant nos résultats montrent que les utilisateurs ont eu moins de difficultés en utilisant la souris. Il n'est pas à exclure que ce résultat dépende de la tâche, voire de l'expertise des utilisateurs. En effet les non voyants n'utilisent pas de souris. Il est donc possible que l'expertise des utilisateurs voyants vis-à-vis de la souris ait eu un rôle prépondérant dans ces résultats.

5.6. Travaux futurs.

Les résultats que nous avons obtenus sont encourageants, mais laissent place à des améliorations. L'observation des utilisateurs et les discussions que nous avons eu avec eux après les expériences nous ont permis de trouver des améliorations à ces techniques d'interaction. Dans un premier temps

nous allons ajouter un hystérésis[6] dans le mode Tacton entre la zone de la forme et l'extérieur de la forme. En effet actuellement si l'utilisateur se déplace sur la limite entre la forme et l'extérieur de la forme, les Tactons oscillaient entre les deux types d'informations. Certains utilisateurs avouent avoir été perturbés par ce phénomène. Le système d'hystérésis permettrait de supprimer cet effet désagréable, en guidant l'utilisateur jusque sur le trait du segment lorsque celui-ci sort de la zone de tolérance. Ainsi pour ressortir l'utilisateur doit parcourir à nouveau l'épaisseur du segment.

Certains utilisateurs souhaiteraient avoir un retour permettant d'être informés quand ils ont effectué un tour complet sur la forme. Nous préférons éviter d'utiliser d'autres canaux pour transmettre l'information afin de les laisser libre à d'autres interactions dans le cas où cette technique serait implémentée dans un système de plus grande envergure. Nous réfléchissons donc à un moyen d'indiquer cette information, sans toutefois surcharger le système par une quantité trop importante d'informations ni multiplier les canaux de communication. Par ailleurs nous ne pouvons explorer que des formes vectorielles. Nous réfléchirons donc aussi à une méthode permettant d'extraire un chemin de guidage directement à partir d'une image bitmap.

Au niveau des expérimentations futures, nous envisageons d'ajouter d'autres formes telles que les parallélogrammes et les trapèzes rectangles. Ainsi que nous l'avons déjà souligné nous réduirons la taille des formes dans le mode de présentation avec la transcription des pixels afin qu'elles paraissent de la même taille qu'avec le mode de présentation par Tactons. Enfin nous souhaitons effectuer ces tests avec des utilisateurs déficients visuels. Cependant ces tests sont longs et nécessitent un grand nombre d'utilisateurs, ce qui limite les possibilités. Ensuite, nous n'avons utilisé que des formes creuses. Il serait intéressant de comparer l'exploration de formes pleines à l'exploration de formes creuses. Enfin nous allons réfléchir à une solution pour représenter d'autres angles et les segments courbes à l'aide de Tacons afin de permettre l'exploration de plus de formes.

Nous avons défini de nouvelles icônes, exploré leurs paramètres et nous les avons utilisé dans une application. Nous allons aborder dans la section suivante une étude préparatoire qui pourra être utilisée à l'avenir pour examiner la manière dont nos icônes peuvent être utilisées conjointement à d'autres icônes utilisant d'autres modalités.

6. Analogie musicale

Jusque là nous avons utilisé ces techniques d'interaction de manière assignée : la modalité était imposée à l'utilisateur. Pour pouvoir représenter une information de manière exclusive ou redondante, il est important de définir un lien entre les différentes modalités. Ainsi si la construction d'icônes se fait de manière analogue, il est possible de choisir indépendamment la modalité pour afficher une information. Ce principe a conduit Hoggan et Brewster à développer des icônes crossmodales pour représenter des informations sur un PDA avec la modalité auditive (earcons) ou avec la modalité

[6]Selon le Trésor de la Langue Française Informatisé (`http://atilf.atilf.fr/`) : « Persistance d'un phénomène quand cesse la cause qui l'a produit. »

tactile (Tactons vibrotactiles) [70]. Les earcons sont basés sur la musique, donc chaque earcon peut être décrit grâce à une partition. De même les Tactons vibrotactiles peuvent aussi être décrits par une partition car une analogie entre les paramètres a été définie par Hoggan et Brewster. Nous allons présenter une manière de décrire les Tactons à matrices de picots avec une notation musicale afin de pouvoir à l'avenir les comparer aux Earcons et aux Tactons vibrotactiles. Ceci fera l'objet d'une étude future non traitée dans cette thèse.

Notes et accords.

Une *note* est définie d'une part par un *son* (ou *hauteur de note*) et par une *durée* (ou *figure de note*) [47]. Un *accord* est un ensemble de notes jouées simultanément, alors qu'un *silence* correspond à un moment où aucune note n'est jouée. Nous pouvons relier les picots aux sons, et ainsi un motif correspond à un accord. Si nous ajoutons une durée nous obtenons une image, qui correspond soit à une note, soit à un accord soit à un silence. La note correspond au cas particulier où un seul picot est levé et le silence à une image dont le motif n'a aucun picot levé. Le terme « accord » en musique fait à la fois référence à l'ensemble de sons (sans durée) et à l'ensemble de plusieurs notes jouées simultanément (avec durée). Par convention, nous définissons la correspondance entre les durées des images et les figures de notes comme suit (Table 1).

Durée de l'image	4	2	1	1/2	1/4	1/8	1/16
Figure de note	Ronde	Blanche	Noire	Croche	Double-croche	Triple-croche	Quadruple-croche
Silence	Pause	Demi-pause	Soupir	Demi-soupir	Quart de soupir	Huitième de soupir	Seizième de soupir

Table 1 – Correspondance entre les durées des images et les figures de notes

L'analogie entre les picots et les hauteurs de notes peut sembler incomplète dans le sens où les matrices que nous utilisons ont deux dimensions, alors que nous n'en définissons qu'une. Ceci peut être compensé par la manière dont les notes sont réparties. En effet en musique occidentale, on utilise 12 notes (do, $do\#$, re, $re\#$, mi, fa, $fa\#$, sol, $sol\#$, la, $la\#$ et si), que l'on réplique par octaves. Les notes peuvent donc être réparties sur deux dimensions : l'octave, et la note. Par exemple un clavier de trois octaves peut être vu en deux dimensions comme 3×12 touches ou en une dimension comme 1×36 touches.

La correspondance entre une image et un accord est illustrée par la figure 25. La matrice utilisée possède 4×4 picots dont 4 sont levés, et la durée est fixée à la valeur 1. L'accord correspondant utilise 4 notes et la figure de note est une noire. Notre analogie concerne les paramètres et non les valeurs. Le choix des hauteurs de notes sur la figure est donc arbitraire.

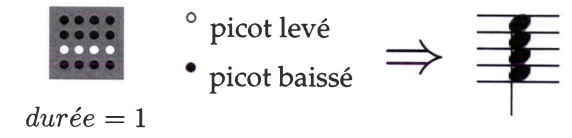


Figure 25 – Correspondance entre une image et un accord

Mesures.

En musique, un *temps* représente une unité de mesure de durée, qui est mise en relation avec un figure de note définie par référence. Une *mesure* est une succession de temps qui commence par un *temps fort*. Le *tempo* représente le nombre de temps joués en une minute, et permet donc de définir la vitesse d'exécution du morceau. Chaque mesure est étiquetée par le nombre de temps qu'elle possède, la figure de note associée à un temps, ainsi que le tempo. Ces différentes informations ne sont écrites sur une partition que lorsqu'elles changent. Les notes sont réparties sur une mesure de telle manière que la somme de leurs durées soit égale au nombre de temps dans la mesure, pondérée par la durée des figure de note de base (voir l'exemple ci-dessous illustré par la figure 26).

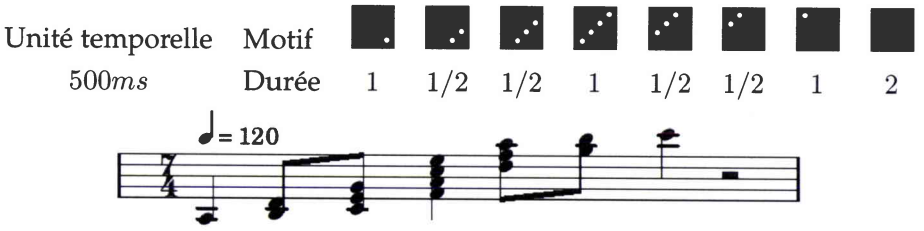


Figure 26 – Correspondance entre un Tacton dynamique et une mesure

Dans notre analogie, nous allons mettre les Tactons en correspondance avec une mesure. L'unité temporelle du Tacton correspond au tempo de la mesure. Pour simplifier la notation nous allons toujours utiliser la noire comme figure de note représentant la durée d'image 1. La nombre de temps dans la mesure est calculé en additionnant les durées des images du Tacton. Si cette somme n'est pas entière, il suffit de la multiplier par son dénominateur afin de la rendre entière, et de multiplier l'unité temporelle (et donc le tempo) par cette valeur afin de garder la même vitesse d'exécution.

L'exemple de la figure 26 montre l'analogie d'un Tacton dynamique de 8 images. L'unité de temps est de $500ms$, c'est-à-dire qu'une image de durée 1 sera affichée $500ms$, une image de durée 2 sera affichée $1s$, et une image de durée $1/2$ sera affichée $250ms$. Le tempo se déduit de l'unité temporelle : une image de durée 1 est affichée $500ms$, donc en une minute sont affichées $60000/500 = 120$ images de durée 1. On dit alors que le tempo est de 120 à la noire. Les durées d'image sont converties en figures de note en utilisant la table 1. Le nombre de temps est calculé en additionnant les durées des images : $1+1/2+1/2+1+1/2+1/2+1+2 = 7$. Une mesure de 7 noires se note $\frac{7}{4}$, le 4 représentant la noire. La blanche est notée 2, la ronde 1, la croche 8, etc. La première et la septième image n'ont qu'un

seul picot de levé. Elles sont donc représentées par de simples notes. La dernière image n'a aucun picot levé et est donc représentée par un silence (une demi-pause). Les autres images possèdent plusieurs picots levés et sont donc représentées par des accords, c'est-à-dire des empilements de notes.

Les Tactons statiques sont des cas particuliers où un seul motif est affiché sur une durée plus ou moins déterminée. Il existe une notation musicale pour ce cas : le *point d'orgue*. En musique le point d'orgue correspond à une note dont la durée est déterminée par le chef d'orchestre ou par les interprètes. On retrouve généralement cette notation à la fin d'un morceau ou d'un mouvement. Un Tacton statique correspond à une note avec un point d'orgue. Sa durée est déterminée par le concepteur de l'interface (que l'on peut considérer comme le compositeur) ou par l'utilisateur (que l'on peut considérer comme l'interprète).

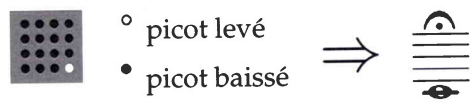


Figure 27 – Correspondance entre un Tacton statique et un point d'orgue

Portées et partitions.

Une *portée* est une suite de mesures jouées en séquence. Une *partition* est un ensemble de portées jouées simultanément. Ordinairement sur une partition de piano, il y a une portée pour chaque main. Sur une partition de direction, chaque instrument a sa portée, ainsi le chef d'orchestre peut voir ce que chaque musicien est en train de jouer. Dans notre cas, une portée correspond à la succession des Tactons jouées sur une matrice de picots. En musique, il est possible de rejouer plusieurs fois une mesure ou un ensemble de mesures grâce à plusieurs notations que nous ne détaillerons pas, comme les *barres de reprise*, les *coda*, les *da capo*, etc. Les différentes portées d'une partition se rapportent dans notre cas aux Tactons joués sur les différentes matrices du dispositif utilisé, et éventuellement de la partie du corps en contact avec cette matrice.

Travaux futurs.

L'analogie décrite ci-dessus couvre l'ensemble des paramètres que nous avons définis pour nos Tactons. Ils permettent de faire la liaison entre les paramètres tactiles et les paramètres sonores d'icônes telles que les earcons. Cette partie théorique effectuée, il reste à vérifier en pratique si les utilisateurs parviennent à faire le lien entre ces paramètres. Le cas échéant, nous pourrions construire des icônes crossmodales ainsi que l'ont fait Hoggan et Brewster avec les tactons vibrotactiles et les earcons [70]. Ce travail reste à faire, et nous nous y attacherons prochainement.

7. Conclusion

Les matrices de picots sont un moyen efficace de transmission d'information. Nous avons proposé des définitions permettant de formaliser la notion de Tacton basé sur des matrices de picots. Deux

types de Tactons ont été définis : des Tactons statiques qui sont des motifs affichés de manière continue, et des Tactons dynamiques qui sont des animations dont les motifs, le rythme et le tempo sont définis. La description des Tactons peut se faire à deux niveaux. Lorsque l'on souhaite décrire la manière dont on affiche les Tactons, on parle alors de ses paramètres. Les paramètres sont les caractéristiques physiques d'un Tacton : motifs, durées, etc. Par contre si on décrit l'information véhiculée par un Tacton on parle de dimensions. Dans ce cas on s'intéresse aux informations transmises par le Tacton en question. Chaque dimension peut être matérialisée à l'aide d'un ou plusieurs paramètres. À l'inverse un paramètre peut être utilisé par plusieurs dimensions. Il faut cependant veiller à choisir les dimensions avec soin afin de ne pas créer d'ambiguïtés au niveau de l'interprétation.

Deux études ont été menées pour concevoir et évaluer des Tactons. La première consiste à créer des Tactons n'utilisant qu'une seule dimension. L'objectif est de trouver dans un premier temps des motifs faciles à interpréter, donc nous n'avons pas travaillé sur les autres paramètres tels que le rythme ou le tempo. Nous avons évalué trois sortes de Tactons : les Tactons statiques qui sont des motifs affichés de manière continue, les Tactons clignotants qui alternent un motif avec une configuration sans picots levés, et les Tactons en vagues qui déplacent un motif sur la matrice. Nous avons mis en évidence des ensembles de Tactons permettant de transmettre efficacement de l'information. Pour les Tactons statiques, il apparaît que les Tactons à peu de picots levés sont difficiles à identifier. À l'inverse les Tactons où trop de picots sont levés sont aussi difficiles à interpréter. Le meilleur compromis est donc de faire des formes allongées, ce qui empêche de confondre deux directions adjacentes, et de localiser le motif sur la matrice. Les Tactons clignotants ont été peu étudiés dans cette expérience car leur reconnaissance dépend du motif utilisé, ce qui a été étudié avec les Tactons statiques. Elle dépend aussi de la vitesse de clignotement, ce qui a été étudié dans l'expérience suivante. Les Tactons en vagues sont moins évidents à identifier, notamment pour certains utilisateurs qui trouvent la sensation désagréable. Les résultats avec les ensembles testés montrent qu'il faut concevoir les Tactons de telle manière que les Tactons adjacents soient suffisamment différents. Par exemple en utilisant un différent nombre d'images pour l'animation. Nous pourrons essayer de modifier la vitesse de lecture des Tactons afin d'améliorer la sensation.

La deuxième série d'expériences vise à évaluer des ensembles de Tactons dont l'information est représentée sur trois dimensions. Ce sont des Tactons clignotants dont les trois dimensions sont la direction, qui agit sur la forme affichée sur le motif, la taille qui fait varier le nombre de picots levés et la vitesse qui agit sur l'unité temporelle du Tacton. Les directions utilisent les formes les plus efficaces trouvées dans les expériences précédentes. Nous avons analysé les résultats obtenus sous deux points de vue différents. En se basant sur les taux d'erreurs nous déduisons que ces Tactons sont plus difficiles à reconnaître que les Tactons à une seule dimension. De plus nous avons montré que l'utilisation de trois valeurs de vitesse induit significativement plus d'erreurs que deux valeurs. Ceci est du à l'ambiguïté introduite par la présence d'une valeur intermédiaire, qui n'avait pas été décelée lors des tests pilotes où seule la vitesse variait. Ainsi nous retiendrons que la combinaison de plusieurs dimensions introduit des difficultés d'interprétation des dimensions

individuelles. C'est pour cette raison que nous avons cherché à analyser la quantité d'information transmise à l'utilisateur de manière globale. En effet dans un sens nous envoyons plus d'information à l'utilisateur, et de l'autre côté cela engendre plus d'erreurs. Nous sommes donc en droit de nous demander quelle est la quantité d'information effectivement transmise à l'utilisateur. Cette fois-ci nous observons une supériorité des Tactons à trois valeurs de vitesse, suivi des Tactons à deux vitesses et enfin les Tactons à une seule dimension. Le gain apporté par l'augmentation de la quantité d'information apportée à l'utilisateur compense donc la perte due à l'incapacité de l'utilisateur de capter toute l'information qui lui est envoyée.

Nous avons utilisé ces Tactons pour concevoir un mode d'exploration de formes géométriques. Cette méthode consiste à guider l'utilisateur de point en point selon un parcours imposé. Cette méthode permet d'une part d'aider l'utilisateur à trouver la forme quelle que soit sa position, et de parcourir la forme en entier. L'expérience que nous avons conduite visait à comparer l'exploration en utilisant cette méthode, et en utilisant une méthode plus traditionnelle consistant à convertir les pixels. Les résultats que nous avons obtenus ne permettent pas d'affirmer qu'une méthode est meilleure que l'autre. Nous devons donc d'une part améliorer notre solution, et d'autre part effectuer plus de tests afin d'avoir plus de données à comparer. D'autre part nous avons utilisé deux modes d'explorations : un avec une tablette graphique et l'autre avec une souris VTPlayer. Cependant les résultats n'ont pas montré de gain apporté par cette solution. Suite à ces résultats, nous avons proposé d'une part plusieurs améliorations pour la technique d'interaction utilisant les Tactons. D'autre part des améliorations dans le protocole expérimental pour les prochaines expérimentations.

La section suivante présentait une analogie de ces Tactons avec la notation musicale. Ainsi il serait intéressant d'étudier la correspondance entre les Tactons que nous avons défini, les Earcons [22], les Tactons vibrotactiles [25] et les PICOB (définis dans le chapitre III). En effet si nous trouvons des dimensions exploitables communes, nous pourrons afficher de manière crossmodale la même information, comme le font Hoggan et Brewster avec les Earcons et les Tactons vibrotactiles [70]. Ceci permettrait d'adapter l'affichage en fonction du contexte, des préférences de l'utilisateur, des périphériques disponibles, etc. Le travail dans cette voie reste à faire.

Pour terminer, il reste des pistes à explorer dans la veine de ce travail. En plus de la comparaison à d'autre types d'icônes déjà évoquées, il serait intéressant d'expérimenter d'autres paramètres telle que la position sur le corps ou l'utilisation de plusieurs rythmes. Pour ce qui est de la position sur le corps, nous utilisons déjà ces Tactons avec soit la main dominante comme dans la première série d'expériences, soit dans la main non-dominante comme dans la seconde. Il serait intéressant de vérifier que la sensation se fait avec la même efficacité. Pour les rythmes, nous avons jusque là juste évalué si certains rythmes permettent de différencier des valeurs de tempo plus facilement que d'autres. L'intérêt de ce paramètre est qu'il est déjà utilisé avec succès dans les Earcons [28] et les Tactons [30]. L'étude sur l'exploration de formes géométriques n'est qu'un prélude à une série de nouvelles expériences en cours de conception. Nous souhaitons modifier l'interaction afin de simplifier la tâche à l'utilisateur, et à lui permettre de savoir quand il a effectué un tour complet sur

la forme. Nous prévoyons aussi de trouver une manière de représenter les segments courbes et les angles inférieurs à 45°.

Dans ce chapitre nous avons introduit un nouveau système d'icônes tactiles, utilisant une matrice de picots. Dans le chapitre suivant nous allons décrire un nouveau système d'icônes kinesthésiques utilisant un périphérique de pointage à retour de force.

Chapitre III

Icônes kinesthésiques

> Nous ne pouvons pas prédire où nous conduira la Révolution Informatique. Tout ce que nous savons avec certitude, c'est que, quand on y sera enfin, on n'aura pas assez de RAM.
>
> Dave Barry

Résumé

Dans ce chapitre nous décrivons une technique d'interaction à retour de force permettant de transmettre de l'information structurée à l'utilisateur. Cette technique d'interaction est basée sur des impulsions dont les paramètres sont la direction, l'amplitude, la longueur et la forme. Les évaluations décrites ci-dessous concernent deux types d'impulsions répondant à ces critères : d'une part des impulsions guidées où l'utilisateur est entièrement guidé par le système au cours du temps. D'autre part nous avons créé des impulsions semi-guidées que l'utilisateur ressent en se déplaçant sur un axe dans lequel il est contraint. Les résultats montrent que les utilisateurs parviennent à reconnaître efficacement la direction et l'amplitude d'impulsions guidées prenant leurs valeurs parmi 6 directions et 2 amplitudes. Ils arrivent aussi à reconnaître des impulsions semi-guidées construites avec 4 valeurs de direction et 2 valeurs d'amplitude. Enfin nous avons évalué des séquences d'impulsions semi-guidées lorsque l'utilisateur est contraint sur un plan horizontal. Les résultats montrent que les utilisateurs ont de grandes difficultés à distinguer les 9 séquences proposées. Des solutions ont été proposées afin de réduire les confusions.

1. Introduction

Il est difficile de positionner les informations dans une environnement virtuel, et particulièrement si l'exploration se fait de manière non-visuelle. Plusieurs recommandations de Sjöström [158] insistent sur le fait que l'utilisateur doit rester en contact avec les zones d'informations et de disposer de points de repère. Il faut donc que les informations soient disponibles directement aux endroits explorés par l'utilisateur. Dans ce chapitre nous définissons un système d'icônes kinesthésiques lues grâce à un périphérique de pointage à retour de force. Ainsi les informations peuvent être lues directement pendant l'exploration. Ce chapitre commence par une description de la technique d'interac-

tion et de son implémentation. Nous aborderons ensuite trois séries d'expériences visant à explorer certains paramètres de ces icônes.

2. Le concept de PICOB

Les périphériques de pointage à retour de force tels que le PHANToM permettent à la fois d'explorer l'environnement et de le ressentir. Il est donc intéressant d'essayer de placer des informations codées dans l'environnement pouvant servir d'annotation ou d'étiquettes. Ainsi l'identification d'un objet peut se faire en lisant l'étiquette. L'idée de PICOB, qui signifie Post-It Haptiques par Codes-Barres, est de placer des informations haptiques dans la scène que l'utilisateur va lire à l'aide d'un geste. Nous allons commencer par définir cette interaction, puis nous détaillerons les expérimentations.

2.1. Définitions

Définition 1 Une *impulsion* est une force exercée par le périphérique à retour de force, dont les caractéristiques sont sa *direction*, son *amplitude*, sa *longueur* et sa *forme*.

La direction est celle dans laquelle l'utilisateur est attiré. La forme correspond à la trajectoire de l'utilisateur, dont l'amplitude est la distance maximale parcourue et la longueur correspond aux limites de la lecture de l'impulsion. La figure 1 montre une impulsion à forme sinusoïdale dont la direction a été fixée vers le haut.

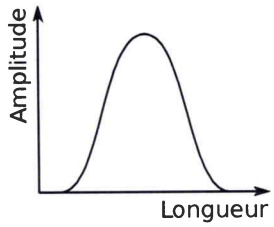


Figure 1 – Longueur et Amplitude d'une impulsion de direction et de forme fixée

Définition 2 Une impulsion *guidée* est une impulsion dont la longueur est une unité temporelle : la position de départ de la lecture est fixée. L'impulsion est ressentie comme une déviation par rapport à la position de départ, qui varie en fonction des autres paramètres.

Définition 3 Une impulsion *semi-guidée* est une impulsion dont la longueur est une unité spatiale : l'utilisateur est contraint sur un axe, et la déviation dépend de la position sur cet axe. Les impulsions sont donc ressenties comme des déformations de l'axe.

Ces deux types d'impulsions sont utilisées dans un contexte différent. Les impulsions guidées se substituent à l'exploration normale de l'environnement, alors que les impulsions semi-guidées peuvent être ressenties pendant l'exploration car elles laissent à l'utilisateur un degré de liberté.

Définition 4 Une *séquence d'impulsions* ou *PICOB* est une série d'impulsions lues les unes après les autres. Les impulsions sont concaténées selon l'axe de la longueur, et espacées par des *pauses* de longueur éventuellement nulle.

2.2. Implémentation

Ces techniques d'interaction ont été développées afin d'être utilisée avec un PHANToM (figure 2). Bien que très performants, les versions de ces périphériques que nous allons utiliser ne permettent pas de créer des forces supérieures à celles exercées par un poignet humain. De ce fait il est impossible de contraindre la position du poignet de l'utilisateur à l'aide des forces rendues par ces PHANToMs. Nous différencierons donc la trajectoire théorique induite par les impulsions et la trajectoire réelle de l'utilisateur.



Figure 2 – PHANToM Desktop et OMNI

L'utilisateur est attiré vers une projection de la position du PHANToM sur la trajectoire qu'il est censé suivre. Afin de réduire l'écart entre la trajectoire théorique et la trajectoire réelle de l'utilisateur nous maximisons la force appliquée. Cependant la force doit être choisie soigneusement. Si on utilise un modèle à force constante, on s'expose à des vibrations car si la force pousse l'utilisateur au delà de la cible, à l'instant suivant l'utilisateur est poussé avec la même force dans la direction opposée, ce qui conduit rapidement à une vibration désagréable qui perturbe l'exploration. Il faut donc décroître la force appliquée lorsque l'on s'approche de la cible. Nous avons donc choisi d'utiliser une force linéaire proportionnelle à la distance entre le PHANToM et la cible (figure 3). Cette force est celle générée par un ressort, notée

$$\overrightarrow{F} = k\overrightarrow{\Delta l}$$

où $\overrightarrow{F}$ est le vecteur force généré, k la constante de raideur et $\overrightarrow{\Delta l}$ le vecteur d'élongation du ressort. L'élongation du ressort correspond au vecteur entre la cible et la position du PHANToM. Il nous reste à choisir une constante de raideur. Une valeur trop faible peut perturber l'identification des propriétés des impulsions car le ressenti est mou, et donc la trajectoire réelle est éloignée de la trajectoire théorique. En outre les trajectoires réelles sur des impulsions différentes peuvent être similaires et donc conduire à des confusions. À l'inverse si la constante est trop élevée, on s'expose à des perturbations telles que les vibrations mentionnées ci-dessus. De plus si la force à appliquer est trop importante, le pilote du PHANToM débraye le périphérique par sécurité pour l'utilisateur et pour le matériel. Le choix de cette constante est donc fait arbitrairement selon le périphérique grâce à des tests empiriques.

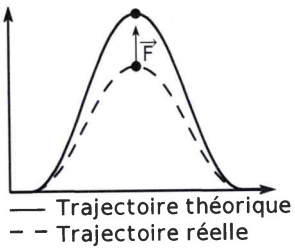


Figure 3 – Longueur et Amplitude d'une impulsion de direction et de forme fixée

Notre but étant de permettre aux utilisateurs de différencier des valeurs relatives pour l'amplitude, la différence entre la trajectoire théorique et la trajectoire réelle n'est pas importante en soi. L'essentiel est d'avoir le plus d'amplitudes distinguables entre elles. L'objectif n'est pas d'identifier les amplitudes, mais de pouvoir distinguer le plus grand nombre possible d'amplitudes. Parmi les quatre paramètres évoqués nous n'allons en faire varier que deux. En effet pour la forme du signal nous allons garder une forme sinusoïdale car nous estimons après quelques essais que d'autres formes telles que des triangles ou des rectangles peuvent être brusques pour les utilisateurs. Ensuite, il est difficile de faire varier en même temps l'amplitude et la longueur d'une impulsion car ces deux propriétés sont proches et risquent d'être difficilement distinguables. Nous avons choisi de faire varier l'amplitude car nous souhaitons créer des séquences, qui se construisent selon l'axe de la longueur. Donc en faisant varier les impulsions sur la longueur nous limitons le potentiel des séquences. En effet, dans le cas des impulsions guidées, la durée d'une séquence peut devenir longue et le système perdrait en efficacité. Pour les impulsion semi-guidées les séquences peuvent prendre beaucoup de place, ce qui peut être une forte contrainte de conception d'un système utilisant cette interaction. Pour finir la direction est potentiellement intéressante. Dans le cas des impulsions semi-guidées il est cependant exclu d'utiliser la direction, déjà utilisée par l'axe de la longueur.

Les expériences présentées ont pour but d'examiner l'efficacité de distinction de valeurs de direction et d'amplitude pour des impulsions guidées puis semi-guidées. Enfin nous étudierons des séquences d'impulsions semi-guidées.

3. Les impulsions guidées

La première étude a été réalisée dans le but d'examiner les impulsions guidées [137]. Nous allons étudier la reconnaissance des deux paramètres à tester séparément dans un premier temps. Ensuite nous allons examiner la faculté des utilisateurs à reconnaître les deux paramètres simultanément.

3.1. Conception

Dans un premier temps, comme nous l'avons expliqué précédemment la forme de l'impulsion choisie est la sinusoïde. Ceci permet d'avoir un ressenti lisse et agréable pour l'utilisateur. La longueur d'une impulsion a elle été fixée à $250ms$ suite à un processus de conception itérative à travers de tests pilotes informels. Les valeurs trop faibles empêchent de reconnaître clairement la direction

et l'amplitude. De plus les grandes amplitudes sont ressenties plus violemment. À l'inverse si la valeur est trop élevée, alors les amplitudes sont difficiles à distinguer car la sinusoïde est trop aplatie.

Les possibilités pour la direction ne sont pas limitées car pour cette interaction nous disposons de tout l'espace autour du curseur. De plus le périphérique utilisé peut exercer des force sur trois degrés de liberté. Nous avons choisi d'utiliser les trois directions orthogonales de l'espace euclidien, et de leurs opposés. Nous avons donc six directions : *haut*, *bas*, *gauche*, *droit*, *avant* et *arrière*.

Le choix des amplitudes a été fait incrémentalement suite à des essais faits par l'expérimentateur. Le seuil inférieur d'amplitude a été limité à $0{,}4cm$. La précision du périphérique avec le type de force exercée empêche de reconnaître clairement la direction pour des valeurs inférieures. À l'inverse des valeurs au dessus de $2{,}25cm$ donnent des impulsions violentes et désagréables à l'utilisateur. De plus cela déclenche parfois le système de sécurité du périphérique car la force appliquée est trop importante. Nous avons utilisé au maximum trois valeurs d'amplitude. En effet à partir de 4 valeurs certaines valeurs dans l'intervalle défini sont très difficiles à distinguer.

3.2. Méthodologie

Un protocole commun a été appliqué pour les trois expériences présentées ci-dessous. 6 utilisateurs âgés de 23 à 47 ans (médiane : 25 ans) recrutés parmi le personnel et les étudiants du département informatique de l'Université Paul Verlaine – Metz ont eu la gentillesse de se prêter à ces expérimentations. Aucun d'entre eux n'a de problèmes majeurs de vision ou de sensibilité kinesthésique. Nous n'avons constaté aucun gaucher ce qui fait que tous ont utilisé leur main droite en tant que main dominante. Afin d'annihiler toute aide visuelle, les utilisateurs ont eu les yeux bandés pendant les tests. Le PHANToM utilisé dans cette expérience est la version Desktop.

L'expérimentateur était présent afin de déclencher la lecture des impulsions et enregistrer les réponses des utilisateurs. Chaque expérience consistait en une ou plusieurs séries ayant chacune des paramètres spécifiques. Ceci sera détaillé lors de la description des différentes expériences. Dans chaque série les utilisateurs devaient reconnaître la direction ou l'amplitude (selon l'expérience) de 150 impulsions. Les réponses possibles pour la direction étaient *haut*, *bas*, *gauche*, *droit*, *avant* et *arrière* ; et les réponses possibles pour les amplitudes étaient 1 et 2 ou 1, 2 et 3 selon les expériences où il y avait 2 ou 3 amplitudes.

Chaque série commence par une explication du protocole par l'expérimentateur. Ensuite lors d'une courte session d'apprentissage, l'expérimentateur fait ressentir à l'utilisateur les impulsions qu'il pourra ressentir pendant l'expérience tout en précisant la réponse qu'on attendra de lui. Pendant le test, l'expérimentateur déclenche chaque impulsion une par une et l'utilisateur donne oralement le nom de l'impulsion qu'il a ressentie. L'expérimentateur note la réponse, ce qui déclenche l'impulsion suivante. À la fin de l'expérience l'utilisateur doit remplir un questionnaire (voir annexe A) afin de donner son avis sur la technique d'interaction.

L'analyse des résultats se fera à l'aide de statistiques non paramétriques car les distributions de valeurs obtenues ne suivent pas toujours une loi normale. Le test de Kruskal-Wallis sera utilisé pour détecter les différences significatives, et les comparaisons par paires seront effectuées avec un test de la somme des rangs de Wilcoxon avec un ajustement des p-values de Holm.

3.3. Expérience 1 : reconnaissance de direction

La première expérimentation a pour but de déterminer si les utilisateurs arrivent à reconnaître les 6 directions que nous avons sélectionnées : *haut*, *bas*, *gauche*, *droite*, *avant* et *arrière*. Afin de vérifier que la reconnaissance se fait indépendamment de l'amplitude, nous avons réalisé cinq séries de tests, avec des amplitudes différentes : 0,4cm, 0,7cm, 1cm, 1,5cm et 2,25cm. Le choix des valeurs a été fait suite à des tests préliminaires. Dans un premier temps nous avons commencé par 1cm, puis nous avons ajouté la moitié, et ensuite encore la moitié. Les tests conduits ont eu des résultats satisfaisants. Nous voulions essayer d'autres valeurs, cependant comme les grandes valeurs commençaient à être désagréables, les seules valeurs possibles à explorer sont les valeurs inférieures à 1cm. Nous avons donc cherché un seuil minimal avec des tests pilotes, ce qui a donné 0,4cm. Nous avons donc testé cette valeur et la valeur intermédiaire 0,7cm.

Les hypothèses de cette expérience sont les suivantes :

- Les taux d'erreur seront très faibles. En effet, les tests pilotes ont donné l'impression que les directions sont faciles à reconnaître. Nous pensons que les tests confirmeront ces impressions.
- Il y aura une différence significative entre les résultats des amplitudes extrêmes par rapport aux autres amplitudes. Nous posons cette hypothèse car les valeurs extrémales sont aux limites du confort d'utilisation. Nous pensons donc que ces valeurs provoqueront plus d'erreurs.

Résultats.

La figure 4 montre les taux d'erreur pour chacune des 5 amplitudes amplitudes testées. Toutes les séries ont eu des taux d'erreur pratiquement nuls. En effet seule la série d'amplitude 0,7cm a obtenu une médiane de 1 erreur sur 150, alors que les autres série ont toutes eu une médiane de 0 erreurs. Le test de Kruskal-Wallis ne montre pas de différences significatives entre les séries ($\chi^2 = 7{,}57, p = 0{,}10$).

Discussion.

Ainsi que nous l'avions prévu après les tests pilotes, les directions semblent faciles à reconnaître. Aucun utilisateur n'a fait plus d'une erreur sur le total des cinq séries, et deux utilisateurs n'ont fait aucune erreur. Concernant les avis des utilisateurs, seul un utilisateur avoue avoir parfois des difficultés à déterminer la direction, et trois autres utilisateurs ont admis avoir quelques hésitations de temps en temps. Deux utilisateurs ont le sentiment d'avoir fait légèrement plus de fautes qu'ils n'en ont fait en réalité. D'après les résultats et les sentiments des utilisateurs, nous pouvons conclure que les directions sont aisément distinguables et ce faisant nous validons la première hypothèse. Concernant la seconde hypothèse, les résultats montrent qu'il y a si peu d'erreurs d'identification que les outils statistiques montrent qu'il n'y a pas de différence entre les différentes amplitudes. En

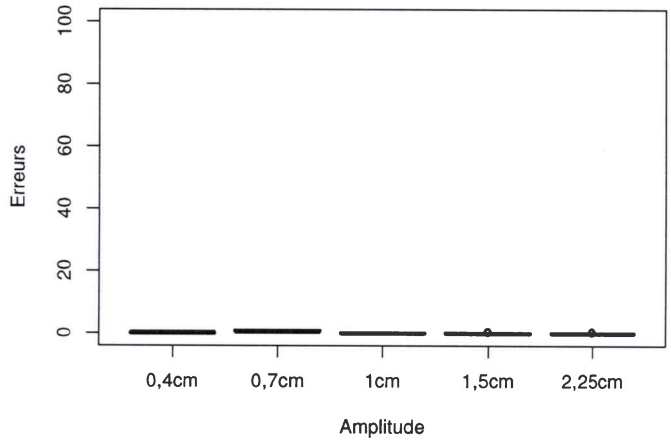


Figure 4 – Expérience 1 : erreurs pour les différentes valeurs d'amplitude

outre les amplitudes extrêmes n'ont pas provoqué significativement plus d'erreurs que les autres, ce qui invalide la deuxième hypothèse.

Deux utilisateurs ont signalé que les amplitudes ressenties n'étaient pas les mêmes d'une direction à l'autre. Selon le premier, les impulsions vers la droite semblent plus grandes que celles vers haut et celles vers le bas. Selon le second, les impulsions vers le haut, vers le bas, vers la gauche et vers la droite sont plus petits que celles vers l'avant et vers l'arrière. Ceci est tout a fait explicable avec la différence entre la trajectoire théorique et la trajectoire réelle. Plus la résistance de la main est grande, plus la force ressentie est grande et la distance parcourue est petite. Les impulsions avant et arrière sont dans le prolongement du bras donc la résistance est naturellement plus forte que dans les autres directions. Ce phénomène n'est pas gênant en soi car la distinction d'amplitude dans les expérience suivantes est relative et non absolue. Cependant ceci peut être une source de confusion lors de l'identification de l'amplitude, et nous devons donc limiter le nombre de valeurs. La valeur de $2{,}25cm$ a été jugée trop violente par les utilisateurs. Elle impose aux utilisateurs de tenir le stylet très fermement.

3.4. Expérience 2 : reconnaissance d'amplitude

Le but de cette seconde expérimentation est d'étudier la discrimination d'amplitudes. Suite à des tests informels, nous avons choisi de n'utiliser que 3 valeurs d'amplitudes car la tâche s'annonçait plus difficile que l'identification de direction. Les utilisateurs ne devaient identifier que des valeurs relatives : *petit*, *moyen* et *grand*, sans se soucier de la valeur réelle de chaque amplitude. Les valeurs sélectionnées utilisent la plus petite valeur utilisée jusque là : $0{,}4cm$, la seconde plus grande (étant donné que la plus grande a été jugée trop violente) : $1{,}5cm$, et une valeur intermédiaire : $0{,}95cm$. Nous avons jugé qu'une seule série était nécessaire car la discrimination de direction est aisée quelle que soit la direction. Nous avons donc choisi de manière arbitraire de n'utiliser pour cette série que des impulsions vers le haut.

Les hypothèses pour cette expérience sont les suivantes :

- Le taux d'erreur sera faible, et la discrimination de trois valeurs d'amplitude est donc aisée. En effet, nous avons volontairement réduit le nombre d'amplitudes utilisées pour éviter d'avoir un surplus de confusions.
- Il n'y aura pas de confusions entre les deux valeurs extrêmes. En effet les valeurs extrêmes sont suffisamment différentes pour que les utilisateurs n'aient pas de confusion entre elles.

Résultats.

Les taux d'erreur au total, ainsi que les détails pour chaque amplitude proposée sont présentés dans la figure 5. Les utilisateurs ont fait une médiane de 6,66% d'erreurs d'identification. En examinant de plus près les valeurs pour chaque amplitude présentée, ils ont réalisé 1,04% d'erreurs avec les impulsions de $0{,}4cm$, 6,36% avec les amplitudes de $0{,}95cm$ et 10,63% avec les amplitudes de $1{,}5cm$. Le test de Kruskal-Wallis montre une différence significative de reconnaissance selon l'amplitude proposée ($\chi^2 = 11{,}78$, $p = 0{,}002$). Grâce à la comparaison des paires, on découvre que les impulsions de $0{,}4cm$ provoquent significativement moins d'erreurs que les autres amplitudes ($p = 0{,}013$ pour $0{,}95cm$ et pour $1{,}5cm$). Si nous examinons la quantité d'information transmise, l'information envoyée à l'utilisateur contenait 1,58 bits, et les utilisateurs ont reçu une médiane de 1,25 bits.

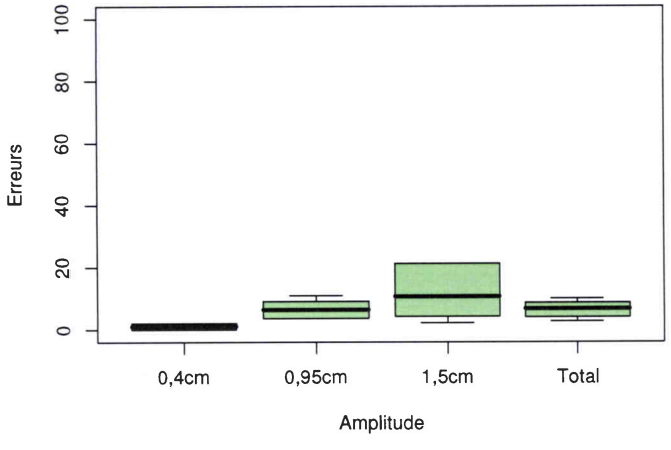


Figure 5 – Expérience 2 : erreurs

La table 1 montre les confusions des utilisateurs en fonction des amplitudes proposées. Nous notons que la majorité des confusions se font entre les amplitudes de $0{,}95cm$ et $1{,}5cm$. D'ailleurs il n'y a aucune erreur commise entre les amplitudes de $0{,}4cm$ et celles de $1{,}5cm$.

Discussion.

Les taux d'erreur sont relativement faibles (moins de 10%), en particulier lorsque l'amplitude de $0{,}4cm$ était proposée. Par contre la distinction entre les deux plus grandes amplitudes pose quelques soucis. En effet la grande amplitude est interprétée comme étant l'amplitude moyenne dans 11,70% des cas. Ceci est probablement dû au phénomène de l'écart entre les trajectoires théoriques et réelles

		Réponse attendue		
		$0{,}4cm$	$0{,}95cm$	$1{,}5cm$
Réponse de l'utilisateur	$0{,}4cm$	98,96%	1,21%	0,00%
	$0{,}95cm$	1,04%	93,33%	11,70%
	$1{,}5cm$	0,00%	5,45%	88,30%

Table 1 – Expérience 2 : moyenne des matrices de confusion

déjà évoqué. Si l'utilisateur résiste trop, l'amplitude est ressentie plus petite. Une solution serait d'augmenter la valeur de la grande amplitude, mais les utilisateurs trouvent cette valeur suffisamment violente. Si nous examinons l'information transmise, nous avons une médiane de 1,25 bits, et un minimum de 1,14 bits. Nous transmettons donc plus d'informations que si on n'utilisait que deux valeurs, car dans ce cas au mieux on transmettrait 1 bit d'information. Pour ces raisons nous pouvons accepter la première hypothèse selon laquelle la distinction d'amplitude est aisée.

La moyenne des matrices de confusion de la table 1 montre qu'il n'y a aucune erreur entre les petites et les grandes amplitudes. En effet les erreurs de reconnaissance impliquaient toujours la valeur intermédiaire. Des efforts seront faits pour équilibrer les erreurs afin d'essayer de faire baisser le taux d'erreur total. Néanmoins nous pouvons accepter la seconde hypothèse car il n'y a aucune confusion entre les valeurs extrêmes.

3.5. Expérience 3 : reconnaissance simultanée de direction et d'amplitude

La troisième expérience a pour objectif d'étudier l'identification simultanée de la direction et l'amplitude d'impulsions. Trois séries ont été réalisées. Comme la discrimination de direction était jugée aisée par les utilisateurs lors de l'expérience 1, nous avons gardé les 6 mêmes valeurs. Par contre, même si la distinction des 3 amplitudes s'est faite avec succès, des progrès sont possibles. Nous avons donc dans un premier temps conduit une série avec les 3 mêmes valeurs d'amplitude que dans l'expérience précédente : $0{,}4cm$, $0{,}95cm$ et $1{,}5cm$. Dans la deuxième série nous avons essayé de rapprocher la valeur intermédiaire de la plus petite valeur afin d'équilibrer les erreurs. Pour cela nous avons choisi une progression de valeurs exponentielle : $0{,}4cm$, $0{,}8cm$ et $1{,}6cm$. Enfin dans la dernière série nous n'utilisons que deux valeurs, qui sont les extrêmes de la deuxième série : $0{,}4cm$ et $1{,}6cm$.

Les hypothèses concernant cette expérience sont énoncées ci-dessous :

- Les taux d'erreur de direction ne seront pas affectés par la présence de plusieurs amplitudes. En effet c'était le cas pour les impulsions guidées et la reconnaissance des directions était aisée dans la première expérience.
- Les utilisateurs feront moins d'erreurs avec deux valeurs d'amplitude qu'avec trois valeurs d'amplitude. En effet la présence de plusieurs amplitudes et plusieurs directions simultanément risque de compliquer la tâche aux utilisateurs.

- La progression de valeurs exponentielle provoquera significativement moins d'erreurs que la progression de valeurs linéaire. En effet nous avons conçu cet ensemble de valeurs pour améliorer le taux d'erreur justement.
- L'information transmise par les Tactons à 3 amplitudes sera significativement plus importante que celle transmise par les Tactons à 2 amplitudes. Les impulsions à 3 amplitudes possibles possèdent 4,17 bits d'information, alors que ceux à 2 amplitudes en possèdent 3,58 bits. Nous pensons que la différence de taux d'erreur entre les séries à 3 amplitudes et la série à deux amplitudes sera suffisamment faible pour que l'information transmise soit différente.

Résultats.

Les erreurs d'identification de direction sont très faibles : 3 utilisateurs n'ont fait aucune faute sur les 3 séries, deux utilisateurs n'ont fait que 2 fautes, et un autre a fait 4 erreurs. La médiane est de 0,33 erreurs pour la série à deux amplitudes et de 0 pour les deux séries à trois amplitudes. Le test de Mann-Whitney montre qu'il n'y a pas de différence de reconnaissance de direction entre les résultats de l'expérience 1 et celle-ci ($W = 263$, $p = 0{,}8$). L'étude des erreurs d'identification d'amplitude est plus intéressante. Les taux d'erreur pour les trois séries sont représentés dans la figure 6. Les utilisateurs ont réalisé une médiane de 15,33% d'erreurs avec la progression linéaire, 6,00% avec la progression exponentielle et 0,33% avec deux valeurs d'amplitude. Le test de Kruskal-Wallis révèle qu'il y a des différences significatives entre les trois conditions ($\chi^2 = 12{,}41$, $p = 0{,}002$). La comparaison des paires montre que la condition à deux valeurs provoque significativement moins d'erreurs que les deux autres conditions ($p = 0{,}013$ dans les deux cas). Il n'y a pas de différences significatives entre les deux conditions utilisant trois valeurs d'amplitude.

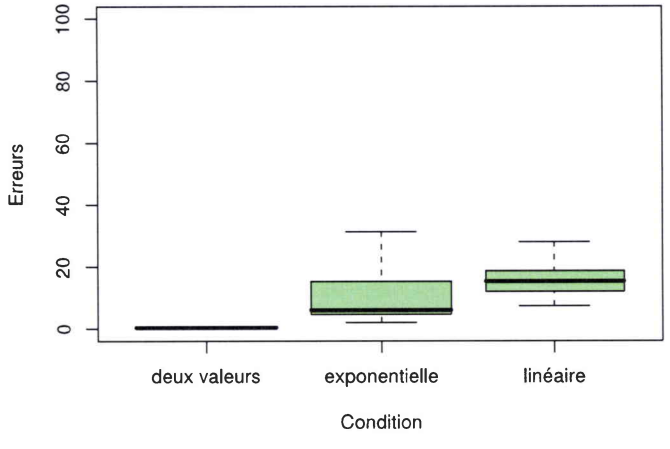


Figure 6 – Expérience 3 : erreurs

Les erreurs entre les différentes valeurs d'amplitudes des deux conditions utilisant 3 amplitudes sont représentées sur la figure 7. Dans la série avec les valeurs d'amplitude espacée de manière linéaire, les utilisateurs ont fait un taux d'erreur médian de 0,96% lorsque l'amplitude était de $0{,}4cm$, 16,66% quand elle était de $0{,}95cm$ et 28,30% quand l'amplitude était de $1{,}5cm$. Le test de Kruskal-

Wallis montre une différence significative de reconnaissance de ces trois amplitudes ($\chi^2 = 10{,}67$, $p = 0{,}004$). Les comparaisons par paires montrent une fois de plus que l'amplitude de $0{,}4cm$ est significativement mieux reconnue que celle de $0{,}95cm$ ($p = 0{,}038$) et celle de $1{,}5cm$ ($p = 0{,}014$). Dans la série dont la progression de valeurs est exponentielle, les utilisateurs ont réalisé un taux d'erreur médian de 2,08% quand l'amplitude proposée était de $0{,}4cm$, 3,63% lorsqu'elle était de $0{,}8cm$ et 12,76% lorsqu'elle était de $1{,}6cm$. Le test de Kruskal-Wallis ne montre pas de différences significatives cette fois ($\chi^2 = 5{,}83$, $p = 0{,}054$). Les erreurs sont donc plus équilibrées.

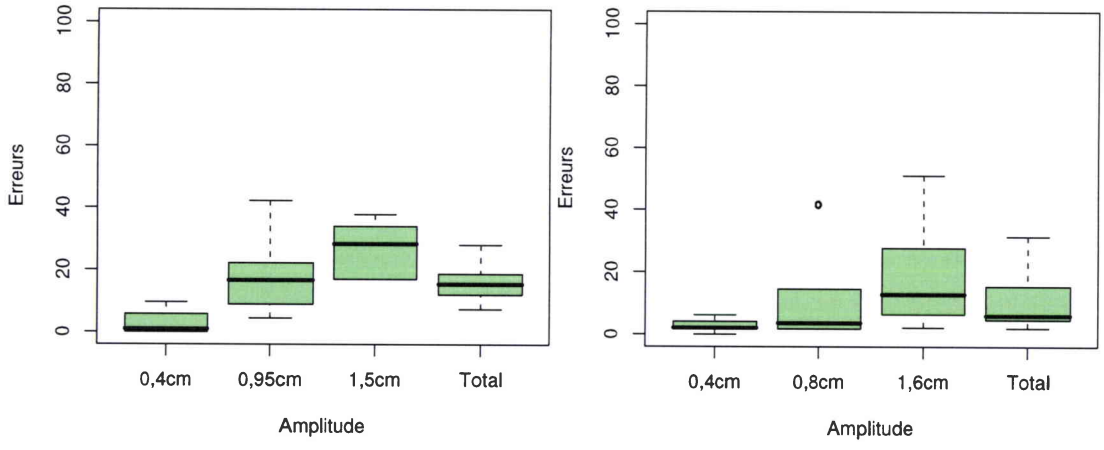


Figure 7 – Expérience 3 : erreurs des conditions à 3 amplitudes

Les résultats de l'analyse de transmission d'information sont représentés sur la figure 8. Les deux séries utilisant trois valeurs d'amplitude avaient un potentiel de 4,16 bits d'information. La quantité médiane d'informations transmises aux utilisateurs est de 3,75 bits avec la progression de valeurs linéaire, et 3,94 bits avec la progression exponentielle. La série utilisant 2 valeurs d'amplitude avait un potentiel de 3,58 bits, et les utilisateurs ont reçu une médiane de 3,54 bits d'information, soit une perte de seulement 0,04 bits. Le test de Kruskal-Wallis montre une différence significative entre les trois séries ($\chi^2 = 6{,}88$, $p = 0{,}032$). Cependant les comparaisons par paires ne permettent pas de trouver de différence significative entre les trois conditions ($p > 0{,}1$).

Discussion.

Premièrement nous remarquons que la présence de plusieurs amplitudes ne perturbe pas l'identification des directions. En effet les résultats de la reconnaissance de direction de cette expérience sont statistiquement équivalents à ceux de la première expérience où seule la direction variait. Nous pouvons donc accepter la première hypothèse. Ensuite, nous remarquons que la série utilisant deux valeurs d'amplitude a effectivement conduit à moins d'erreurs que les deux séries à trois amplitudes. Ainsi nous validons la seconde hypothèse. Le taux d'erreur de la série à progression de valeurs d'amplitude exponentielle est plus faible que celui de la série à progression de valeurs linéaire. Ceci est dû en particulier à la baisse du taux d'erreur lorsque l'amplitude moyenne était proposée. Cependant il n'a pas été possible de dégager de différence entre les résultats de ces deux séries. Nous devons donc

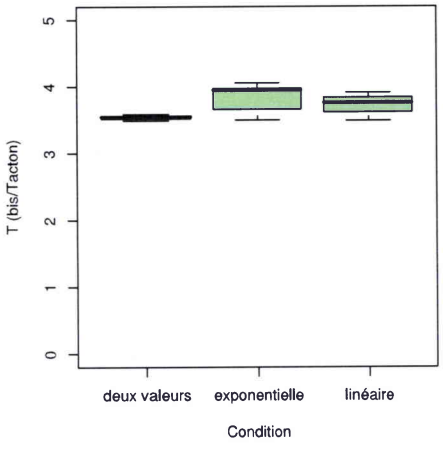


Figure 8 – Expérience 3 : information transmise

rejeter la troisième hypothèse. Nous notons cependant au passage que les erreurs se sont équilibrées. Les résultats fournis par l'analyse de transmission d'information sont très intéressants dans le sens où ils montrent que très peu d'informations ont été perdues en combinant la reconnaissance de direction et d'amplitude. Si nous faisons la somme de la médiane de la quantité d'information de direction transmise dans la première expérience avec la médiane de la quantité d'information d'amplitude transmise dans la seconde, nous obtenons environ 3,83 bits d'information. Or en combinant ces deux informations dans la série à progression linéaire nous obtenons 3,75 bits, soit une perte de seulement 0,08 bits. Nous pouvons en déduire que selon la quantité d'information transmise, l'ajout des directions ne perturbe pas la reconnaissance d'amplitude. L'analyse de la transmission d'information ne révèle cependant pas de différences significatives entre les trois méthodes testées. Nous rejetons donc la quatrième hypothèse. De plus comme l'ajout d'une troisième amplitude n'ajoute pas significativement plus d'information mais ajoute significativement plus d'erreurs, nous éviterons d'utiliser plus de deux valeurs d'amplitude à l'avenir.

3.6. Conclusion

Cette étude avait pour objectif d'examiner les paramètres de directions et d'amplitude pour des impulsions guidées. Trois expériences ont été menées : la première vise à examiner la distinction de direction, la seconde examine la discrimination d'amplitude, et la troisième examine l'identification simultanée de direction et d'amplitude. La première expérience consistait à reconnaître des impulsions de 6 directions orthogonales, dans plusieurs séries dont chacune propose des impulsions d'une amplitude différente des autres. Il en résulte que quelle que soit l'amplitude testée, les utilisateurs arrivent à reconnaître la direction donnée aisément. Cette expérience a permis d'une part de découvrir des valeurs d'amplitudes utilisables, et d'autre part de se rendre compte que l'amplitude peut être ressentie différemment selon la direction. Dans la deuxième expérience les utilisateurs n'ont réalisé qu'une seule série, dans laquelle ils ont pu ressentir des impulsions vers le haut de trois amplitudes différentes, espacées équitablement. Les résultats montrent que les utilisateurs arrivent à distinguer efficacement les trois valeurs. Cependant il y a quelques confusions

par rapport à la valeur médiane. Un nouvel ensemble de valeurs a été créé afin de tenter d'améliorer les résultats dans la troisième expérience. Dans cette expérience les utilisateurs devaient reconnaître simultanément la direction et l'amplitude des impulsions. Les utilisateurs arrivent à reconnaître toujours aussi facilement la direction. Le taux d'erreur d'amplitude de la série utilisant les mêmes valeurs que l'expérience précédente est supérieur à celui de l'expérience précédente. Par contre en examinant la transmission d'information, peu d'informations ont été perdues par rapport aux séries où la direction et l'amplitude étaient reconnues séparément. Cependant la présence d'une troisième valeur d'amplitude n'apporte pas significativement plus d'informations. De plus les résultats montrent que la série utilisant deux valeurs d'amplitude provoque moins d'erreurs que les séries à trois amplitudes. Il est donc évident que les bénéfices des impulsions à trois amplitudes en terme d'information transmise ne compensent pas les pertes induites par le taux d'erreur plus élevé que celui des impulsions à deux amplitudes. Nous n'utiliserons donc à l'avenir que des impulsions guidées à 6 directions et 2 amplitudes.

Après avoir étudié des paramètres pour les impulsions guidées, nous allons étudier les impulsions semi-guidées. Dans la section suivante nous allons expérimenter la reconnaissance des paramètres de direction et d'amplitude pour ces impulsions.

4. Les impulsions semi-guidées

La seconde étude menée avait pour objectif d'étudier les impulsions semi-guidées [138]. Cette fois encore nous avons étudié les paramètres de direction et d'amplitude. En premier nous avons examiné le paramètre de dimension, puis nous avons ajouté d'autres amplitudes.

4.1. Conception

Dans cette expérience nous faisons varier les mêmes paramètres que dans l'expérience précédente : la direction et l'amplitude. Nous maintenons une forme sinusoïdale à l'instar de la section précédente. En effet nous n'avons pas eu de remarques négatives concernant la forme, ou de faits laissant à penser qu'il faudrait la changer. La forme sinusoïdale sera donc utilisée une fois de plus. Concernant la longueur, nous utilisons une unité spatiale et non temporelle cette fois. Avant tout nous avons décidé de représenter ces impulsions sur une ligne horizontale s'étalant de droite à gauche. La technique reste valable avec d'autres orientations, et nous en utiliserons d'autres dans des études ultérieures (section 5). Au déclenchement de l'impulsion, l'utilisateur est contraint sur une ligne horizontale passant par sa position actuelle. Elle s'étend d'une part à gauche, et d'autre part à droite où se trouve l'impulsion après $5mm$. L'impulsion fait $5mm$ de longueur et après un espace de $2mm$ la contrainte sur la ligne est arrêtée. La figure 9 résume la conception de ces impulsions.

Concernant les amplitudes nous avons décidé d'utiliser les valeurs ayant eu le plus de succès avec les impulsions guidées : $0{,}4cm$ et $1{,}6cm$ lorsque nous utilisons 2 amplitudes, et nous ajoutons $0{,}8mm$ lorsque nous utilisons 3 amplitudes. Le paramètre de direction est plus limité dans cette expérience car la droite et la gauche ne peuvent pas être utilisées. Nous souhaitons rester sur des directions orthogonales car d'une part nous n'estimons pas avoir besoin des autres directions, et d'autre part

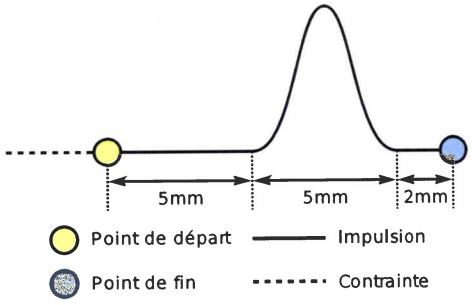


Figure 9 – Impulsion semi-guidée

la multiplication des valeurs risque d'augmenter dramatiquement le taux d'erreur. Nous avons donc restreint les directions à quatre valeurs : *haut*, *bas*, *avant* et *arrière*.

4.2. Méthodologie

Le protocole utilisé est similaire à celui utilisé dans l'expérience précédente. 6 utilisateurs âgés de 23 à 37 ans (médiane : 25 ans) et recrutés parmi le personnel et les étudiants du département informatique de l'Université Paul Verlaine – Metz ont participé aux tests. Aucun de ces utilisateurs n'est gaucher, et aucun n'a de problèmes majeurs de vision ou de sensibilité kiesthésique. Les utilisateurs avaient les yeux bandés pendant la durée des tests afin d'éviter à tout indice visuel d'affecter les résultats. Le même type de PHANToM que dans l'expérience précédente a été utilisé, c'est-à-dire la version Desktop.

Une fois de plus l'expérimentateur était présent pendant les tests afin de lancer les impulsions et recueillir les réponses des utilisateurs. Plusieurs séries ont été effectuées dans chacune des expériences. Cette fois aussi les détails de ces séries seront fournis lors de la description des expériences elles-mêmes. Les séries contenaient 150 impulsions à reconnaître. Les réponses possibles pour la direction étaient *haut* ou *bas* lorsqu'il n'y avait que deux directions et *haut*, *bas*, *avant* et *arrière* lorsqu'il y en avait quatre. Pour les amplitudes les réponses étaient 1 ou 2 lorsqu'il y avait deux amplitudes et 1, 2 ou 3 lorsqu'il y en avait trois.

L'expérimentateur expliquait le protocole avant de commencer chaque série. Ensuite il montrait à l'utilisateur les impulsions qu'il allait ressentir lors d'une courte session d'apprentissage. Le déroulement du test est identique à celui de l'expérience précédente : l'expérimentateur déclenche les impulsions une à une, l'utilisateur les parcourt et donne sa réponse. L'impulsion suivante est déclenchée, etc. L'utilisateur remplit un questionnaire (voir annexe A) à la fin de chaque série afin de donner ses impressions.

Nous utilisons toujours des statistiques paramétriques, avec des tests de la somme des rangs de Wilcoxon avec un ajustement des p-values de Holm afin de détecter les différences significatives entre

les paires de résultats. Nous examinerons aussi la quantité d'information transmise afin d'examiner le potentiel de cette technique d'interaction.

4.3. Expérience 1

À l'instar de la première expérience pour les impulsions guidées, cette première expérience va examiner l'identification de la direction des impulsions. Comme nous l'avons déjà expliqué, nous ne pouvons pas utiliser les directions *droite* et *gauche* car elles correspondent à l'axe de contrainte. Nous nous contentons donc de quatre directions : *haut*, *bas*, *avant* et *arrière*. De plus d'après des tests pilotes effectués lors de la conception du protocole de test, il est apparu que la reconnaissance de direction semblait légèrement plus difficile que dans le cas guidé. Nous avons donc décidé de commencer par deux directions : *haut* et *bas*, puis nous avons utilisé les quatre. Nous avons aussi testé les directions sur deux amplitudes : 0,4*cm* et 1,6*cm*, ce qui donne quatre séries de tests.

Les hypothèses de cette expérience sont les suivantes :

- Les utilisateurs feront plus d'erreurs avec les impulsions à quatre directions qu'avec les impulsions à deux directions. En effet les tests pilotes suggèrent que la reconnaissance de la direction n'est pas aussi triviale qu'avec les impulsions guidées.
- Il n'y aura pas de différence entre les séries utilisant une amplitude de 0,4*cm* et celles utilisant une amplitude de 1,6*cm*. En effet rien ne laisse penser que l'amplitude influe sur la reconnaissance de la direction.
- La quantité d'information transmise par les Tactons à quatre directions sera plus importante que celle transmise par les Tactons à deux directions. En effet lorsqu'il n'y a que deux valeurs d'amplitude, le potentiel d'information n'est que de 1 bit alors qu'il est de 2 bits lorsqu'il y a quatre directions. Nous pensons que les erreurs ne seront pas suffisamment importantes dans les séries à quatre directions pour que l'information transmise baisse de moitié.

Résultats.

Les taux d'erreur pour ces quatre sessions sont représentés sur les graphiques de la figure 10. Les utilisateurs ont réalisé une médiane de 1% d'erreurs dans les deux séries où les amplitudes étaient de 0,4*cm*. Dans les séries de 1,6*cm*, les utilisateurs ont réalisé une médiane de 1,66% pour la série à 2 directions, et 0,66% pour la série à 4 directions. Le test de Wilcoxon ne permet pas de trouver de différence significative entre les séries à amplitude de 0,4*cm* et les séries à amplitude de 1,6*cm* ($W = 72, p = 1$). De plus un autre test de Wilcoxon ne montre pas de différences significatives en terme de taux d'erreur entre les séries à 2 directions et celles à 4 directions ($W = 84, p = 0{,}49$).

Les résultats de transmission d'information sont représentés sur la figure 11. Les utilisateurs on capté une médiane de 0,92 bits d'information dans les séries à 2 directions contre 1,93 bits pour les séries à 4 directions. Le test de Wilcoxon montre que la différence d'information transmise est significative ($W = 0, p < 0{,}001$).

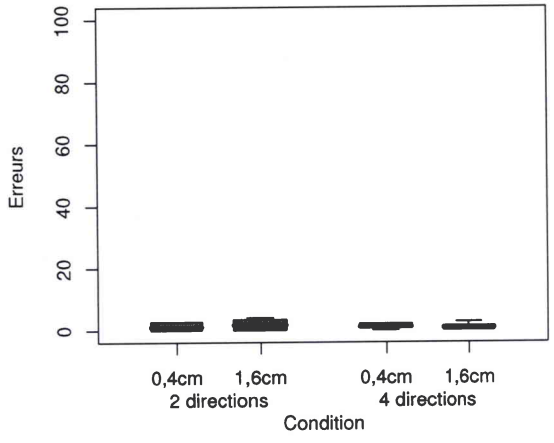


Figure 10 – Expérience 1 : erreurs de direction

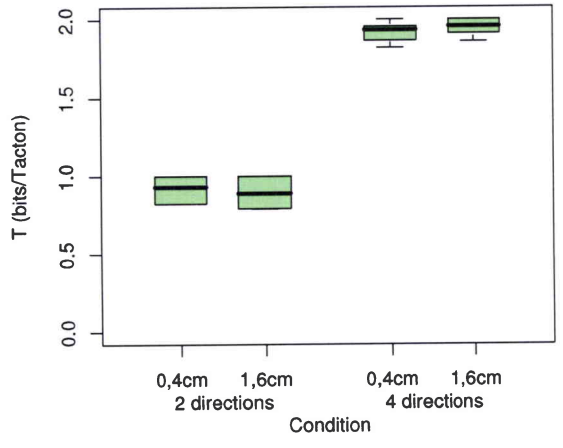


Figure 11 – Expérience 1 : information transmise

De manière générale les utilisateurs avouent préférer la petite amplitude à la grande. De plus, comme dans le cas des impulsions guidées, les utilisateurs ont remarqué que le ressenti de l'amplitude n'était pas le même selon la direction donnée.

Discussion.

La première constatation est que les taux d'erreur sont faibles : moins de 4%, avec des médianes autour de 1%. Le test de Wilcoxon montre qu'il n'y a pas de différences significatives entre les séries à 2 directions et les séries à 4 directions. Le nombre de directions n'influe pas sur les résultats donc nous devons rejeter notre première hypothèse. Il n'y a pas non plus de différence significative entre les séries utilisant des impulsions d'amplitude $0{,}4cm$ et les séries utilisant des impulsions de $1{,}6cm$. L'amplitude n'influe pas sur la reconnaissance de direction, ce qui nous permet de valider la seconde hypothèse. L'information transmise dans les séries à 2 directions varie entre 0,79 bits et 1 bit, alors que dans les séries à 4 directions elle varie entre 1,82 et 2 bits. D'ailleurs le test de Wilcoxon montre que

cette différence est significative. Nous pouvons donc accepter la troisième hypothèse. Pour conclure, nous pouvons utiliser quatre directions sans souffrir de problèmes de reconnaissance majeur.

4.4. Expérience 2

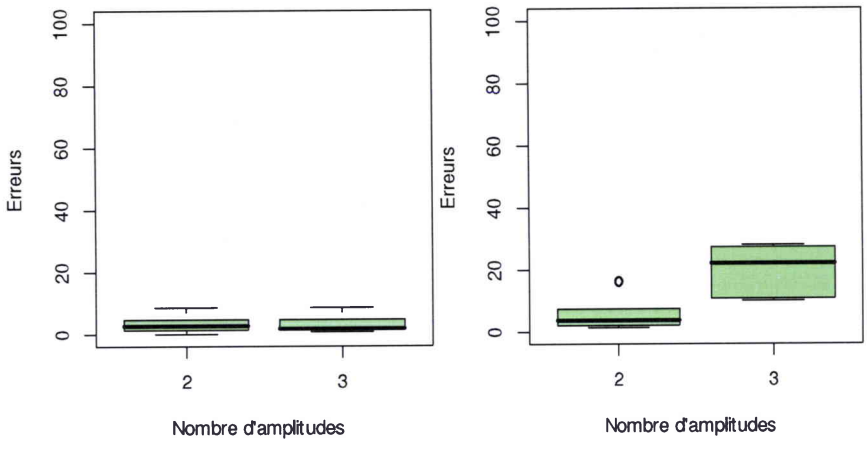
Dans la seconde expérience nous avons étudié la reconnaissance simultanée de plusieurs directions et plusieurs amplitudes. Deux séries ont été réalisées : dans la première nous n'avons utilisé que deux amplitudes : 0,4*cm* et 1,2*cm*. Nous n'avons pas pris 1,6*cm* comme grande valeur car elle était jugée violente. Dans la seconde série nous avons utilisé 3 valeurs d'amplitude : 0,4*cm*, 0,8*cm* et 1,6*cm*. Nous avons été obligés d'utiliser l'amplitude de 1,6*cm* car en dessous il est impossible d'avoir trois valeurs suffisamment espacées. Dans ces deux séries nous avons utilisé les quatre directions : *haut*, *bas*, *avant* et *arrière*. En effet nous avons pu vérifier dans la première expérience que les utilisateurs reconnaissent efficacement aussi bien les impulsions à 2 directions que les impulsions à 4 directions.

Les hypothèses pour cette seconde expérience sont énoncées ci-dessous :

- Il n'y aura pas de différence de taux d'erreur de direction entre les deux séries. En effet, la reconnaissance de direction semble être la tâche la moins difficile à réaliser pour les utilisateurs. De même que dans le cas des impulsions guidées, nous pensons que les utilisateurs n'auront pas plus de difficultés à reconnaître les directions, que ce soit avec 2 amplitudes ou 3 amplitudes.
- La différence entre les résultats de l'expérience où seule la direction varie et l'expérience où la direction varie en même temps que l'amplitude ne sera pas significative. Ainsi nous pensons que l'amplitude n'influe pas du tout sur la reconnaissance de direction.
- Le taux d'erreur de reconnaissance d'amplitude sera significativement plus élevé dans la série à 3 amplitudes que dans la série à 2 amplitudes. Cette hypothèse est motivée par la crainte que la reconnaissance des amplitudes risque d'être problématique que la reconnaissance de direction. En effet dans le cas des 3 amplitudes, nous avons été forcés de prendre une valeur jugée violente par les utilisateurs, ce qui risque de les perturber.
- Il n'y aura pas de différences significatives entre la quantité d'information transmise par les impulsions à 2 amplitudes et la quantité d'information transmise par les impulsions à 3 amplitudes. Nous avons un potentiel de 3 bits pour les impulsions à 2 amplitudes et environ 3,58 bits pour les impulsions à 3 amplitudes. La différence est faible, et le taux d'erreur de la seconde série risque d'être largement supérieur à celui de la première.

Résultats.

Dans un premier temps nous allons examiner les taux d'erreur d'identification de la direction, résumés sur la figure 12. Les utilisateurs ont réalisé une médiane de 2,66% dans la série à deux amplitudes et 1,66% dans la série à trois amplitudes. Le test de Wilcoxon montre que cette différence n'est pas significative ($W = 19{,}5$, $p = 0{,}87$). Par contre en comparant les taux d'erreur de cette expérience avec ceux de la première, on constate que les utilisateurs ont fait significativement plus d'erreurs dans cette seconde expérience ($W = 79{,}5$, $p = 0{,}029$). Cette différence n'est pas significative si on ne considère que la série utilisant deux amplitudes ($W = 38{,}5$, $p = 0{,}07$).



(a) Direction (b) Amplitude

Figure 12 – Expérience 2 : erreurs de direction et d'amplitude

Examinons ensuite les taux d'erreur de distinction de valeur d'amplitude, représentés sur la figure 12. Le taux d'erreur médian obtenu par les utilisateurs est de 3,66% dans la série à deux amplitudes et 22,00% dans la série à trois amplitudes. Cette différence est significative selon le test de Wilcoxon ($W = 2, p < 0{,}01$).

Enfin la figure 13 montre les résultats de l'analyse de la quantité d'information transmise. Dans la série à deux amplitudes, les utilisateurs ont capté une médiane de 2,67 bits d'information sur 3 bits potentiels. Dans la série à trois amplitudes, les utilisateurs ont reçu 2,77 bits d'information sur les 3,58 potentiels. Le test de Wilcoxon montre que la différence entre l'information reçue dans les deux séries n'est pas significative ($W = 12, p = 0{,}39$).

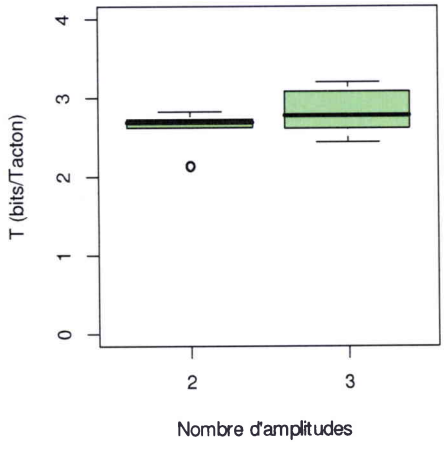


Figure 13 – Expérience 2 : information transmise

Discussion.

Les résultats montrent que les directions sont toujours évidentes à reconnaître. Que ce soit avec 2 amplitudes ou avec 3 amplitudes, les utilisateurs réalisent moins de 10% d'erreurs, avec une médiane de 2,66% dans le pire des cas. En outre il n'y a pas de différence de résultats entre les deux séries, ce qui valide notre première hypothèse. Par contre la comparaison entre les taux d'erreur en direction dans les deux expériences montre que les utilisateurs ont fait significativement plus d'erreurs de reconnaissance de direction lorsqu'il fallait reconnaître l'amplitude en plus de la direction. Nous rejetons donc notre seconde hypothèse. Notons que cette différence n'est pas significative si on ne considère que la série utilisant deux valeurs d'amplitudes. Ceci nous donne une première raison de n'utiliser au plus que deux amplitudes. Concernant les amplitudes, les utilisateurs ont répondu de manière honorable pendant la série à deux amplitudes avec un taux d'erreur inférieur à 10%. Par contre pendant la série à trois amplitudes, les utilisateurs ont eu plus de difficultés et ont réalisé 22% d'erreurs. Cette différence est significative selon le test de Wilcoxon, ce qui valide notre troisième hypothèse. Nous avons une deuxième raison de n'utiliser que deux amplitudes au maximum.

D'après deux utilisateurs, la direction ou l'amplitude est plus facile à distinguer selon la manière dont on tient le stylet du PHANToM. En effet en déplaçant le stylet lentement la direction semble plus facile à reconnaître, alors qu'en le déplaçant rapidement c'est l'amplitude qui semble plus facile à reconnaître. De plus d'après certains utilisateurs les impulsions ne sont pas ressenties de la même manière selon l'amplitude de l'impulsion précédente. En effet quand l'impulsion a une grande amplitude les utilisateurs ont tendance à tenir le stylet plus fermement car l'impulsion est plus violente, et quand le stylet est maintenu plus fermement l'amplitude ressentie est plus faible.

En ce qui concerne la quantité d'information transmise, le potentiel d'information était de 3 bits pour les impulsions à deux amplitudes et 3,58 bits pour les impulsions à trois amplitudes. La différence de 0,58 bits entre les deux a été réduite à 0,09 bits en pratique, avec une médiane de 2,68 bits pour les impulsions à deux amplitudes et 2,77 bits pour les impulsions à trois amplitudes. Cette différence n'est pas significative selon le test de Wilcoxon, ce qui nous permet de valider notre quatrième hypothèse. De plus nous avons une troisième raison de n'utiliser au maximum que deux valeurs d'amplitude. Nous n'utiliserons donc à l'avenir que quatre directions et deux amplitudes.

4.5. Conclusion

Nous avons évalué deux paramètres d'impulsions semi-guidées. Dans une première expérience nous avons évalué la faculté des utilisateurs à reconnaître la direction des impulsions. Quatre séries ont été réalisées, en faisant varier deux paramètres : le nombre de directions deux ou quatre et l'amplitude utilisée pour les impulsions. Les résultats montrent qu'il n'y a de différence ni entre les séries utilisant deux directions et les séries utilisant quatre directions, ni entre les séries utilisant des impulsions de $0{,}4cm$ et celles utilisant des impulsions de $1{,}6cm$. Dans tous les cas les taux d'erreur restent très faibles. On peut constater que la quantité d'information transmise dans les séries où il y avait quatre directions possibles était significativement plus importante que la quantité d'information transmise dans les séries où il y avait deux directions possibles. De plus la quantité

d'information transmise avoisine la quantité d'information maximale, ce qui laisse à penser qu'il est possible de transmettre plus d'information en augmentant la quantité d'information envoyée.

Lors de la seconde expérience nous avons examiné la reconnaissance simultanée de la direction et l'amplitude d'une impulsion. L'expérience a été divisée en deux séries. Dans la première nous avons utilisé quatre directions et deux amplitudes et dans la deuxième quatre directions et trois amplitudes. Tout d'abord les résultats montrent que les utilisateurs arrivent toujours à reconnaître la direction efficacement, en faisant toutefois plus d'erreurs que lorsqu'il s'agissait de ne reconnaître que la direction. Par contre si on ne considère que la série à deux amplitudes, il n'y a pas de différence de résultats. C'est une première raison pour n'utiliser au maximum que deux amplitudes. Les taux d'erreur de reconnaissance d'amplitude montrent que les utilisateurs font plus d'erreurs lorsqu'il s'agit de distinguer trois amplitudes que lorsqu'il s'agit d'en distinguer deux. Nous avons donc une seconde raison de ne pas utiliser trois amplitudes. Selon certains utilisateurs, la direction est plus facile à reconnaître lorsqu'ils déplacent le stylet lentement et l'amplitude est plus facile à reconnaître lorsqu'ils déplacent le stylet rapidement. Ceci empêche de reconnaître efficacement la direction en même temps que l'amplitude. Par ailleurs, les résultats de l'analyse de la quantité d'information transmise montrent que les utilisateurs captent autant d'information avec des impulsions de deux amplitudes qu'avec des impulsions de trois amplitudes. Nous avons donc une troisième raison pour ne plus utiliser trois amplitudes à l'avenir. Nous nous contenterons donc d'utiliser des impulsions à au plus deux amplitudes et quatre directions.

Dans les deux sections précédentes nous avons expérimenté les impulsions individuellement. Dans la section suivante nous allons expérimenter des séquences d'impulsions.

5. Les séquences d'impulsions : PICOB

Le but des impulsions étudiées jusqu'ici est de les utiliser en séquences afin d'encoder l'information de la même manière que l'on crée des mots en concaténant des lettres. Nous présentons ici une étude pratique qui vise à créer des séquences d'impulsions semi-guidées afin de coder des composants électriques pour une application que nous décrirons dans le chapitre IV [140]. Globalement, le but est d'utiliser la multimodalité, et en particulier l'haptique, pour représenter des schémas de circuits électriques. Cette application est destinée aux enfants déficients visuels, afin de les aider à comprendre la notion de circuits électriques. Nous cherchons à représenter les principaux composants en utilisant l'haptique. Nous avons donc décidé d'utiliser des séquences d'impulsions semi-guidées. Cette expérience a pour objectif d'évaluer la conception initiale de nos séquences, et de détecter les problèmes pouvant être causés par la concaténation de plusieurs impulsion.

5.1. Conception

Nous avons choisi les paramètres et leurs valeurs d'une part par rapport aux résultats des expériences sur les impulsions individuelles décrites précédemment. Ainsi nous avons privilégié le paramètre de direction car dans l'étude précédente c'était le paramètre le plus facilement reconnu.

D'autre part nous avons évalué nos besoins. Les circuits étant destinés à des enfants d'école primaire et de collège, nous n'avons besoin au maximum que d'une dizaine de composants.

Nous avons vu dans les expériences précédentes que la perception de l'amplitude peut varier selon l'amplitude précédente à cause de la prise en main du stylet. Cet effet risque de se renforcer et ainsi d'empêcher une reconnaissance efficace. Nous avons donc décidé de n'utiliser qu'une seule amplitude dans cette expérience ($0{,}8cm$). Concernant la direction, les circuits utilisant ces séquences se trouveront sur le plan horizontal. Afin de ne pas interférer entre l'exploration du circuit et la reconnaissance des composants, nous n'allons utiliser que les directions perpendiculaire à ce plan. Nous utilisons donc deux directions : $haut$ et bas. En utilisant de une à trois impulsions nous pouvons donc créer 14 séquences, et de une à quatre impulsions nous pouvons créer 30 séquences d'impulsions. Cependant nous voulons prendre en compte les symétries (figure 14). En effet, si nous ne prenons pas en compte la première symétrie, le sens peut différer selon le sens de lecture. Cette symétrie peut être utilisée dans notre cas pour coder les composants polarisés : les composants non polarisés peuvent utiliser des séquences symétriques et les composants polarisés peuvent utiliser l'une ou l'autre séquence asymétrique selon sa disposition. Ainsi quel que soit le sens de lecture, le composant pourra être identifié, et l'assymétrie pourra être utilisée pour identifier le pôle positif. La seconde symétrie donne des codes inversés : les impulsions vers le haut deviennent des impulsions vers le bas, et vice-versa. Cette symétrie peut être mise à profit pour coder le sens du courant. Notons que certaines séquences peuvent être ambiguës si on utilise cette symétrie, car la composition de cette symétrie avec la première donne l'identité (exemple : $haut - haut - bas - bas$).

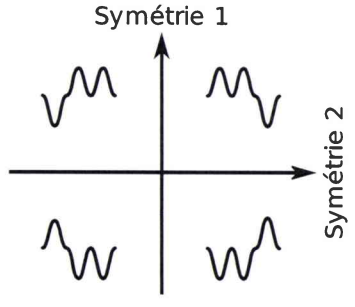


Figure 14 – Symétrie des séquences

Dans cette étude nous allons donc utiliser des séquences de 1 à 4 impulsions, en supprimant les symétries : sur les quatre séquences de la figure 14 nous ne retenons que la première. Nous supprimons aussi deux séquences de 4 impulsions possédant une ambiguïté au niveau de la composition des deux symétries lors de la lecture. Nous obtenons les 9 séquences présentées dans sur la figure 15.

5.2. Méthodologie

Cette expérience a été réalisée avec 11 utilisateurs déficients visuels âgés de 9 à 23 ans ($moyenne =$ 14 ans, $\sigma = 4{,}5$ ans, voir table 2). Deux d'entre eux étaient aveugles, les autres ayant une vision

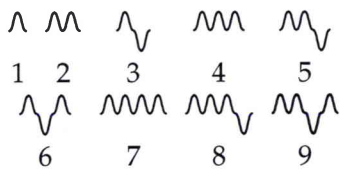


Figure 15 – Séquences d'impulsions utilisées

résiduelle. Ces derniers avaient les yeux bandés pendant les expériences, non pas pour les comparer aux aveugles mais pour ne mesurer que l'impact du retour haptique en éliminant les éventuels indices visuels. Les utilisateurs avaient pour instruction d'utiliser leur main dominante pour tenir le stylet du PHANToM pendant les tests.

Utilisateur	Sexe	Âge	Vision
1	M	23	Vision résiduelle
2	M	20	Aveugle
3	M	18	Aveugle
4	F	15	Vision résiduelle
5	M	14	Vision résiduelle
6	M	12	Vision résiduelle
7	F	12	Vision résiduelle
8	M	12	Vision résiduelle
9	F	10	Vision résiduelle
10	M	10	Vision résiduelle
11	M	9	Vision résiduelle

Table 2 – Utilisateurs

Les sessions de tests commencent par des explications par l'expérimentateur, puis par une session d'entraînement. Dans cette session l'expérimentateur montre à l'utilisateur quelques séquences. Ensuite pendant le test, l'expérimentateur fait sentir une séquence que l'utilisateur peut parcourir autant de fois qu'il le souhaite. Une fois qu'il l'a mémorisée, l'utilisateur peut explorer les 4 propositions de réponse autant de fois qu'il le désire avant d'indiquer laquelle est identique.

Comme les séquences testées dans cette expérience seront utilisées dans un logiciel d'exploration de circuits électriques et que ces circuits seront dans le plan horizontal, les séquences pourront être lues selon diverses orientations dans ce plan. Nous n'utilisons que deux orientations : d'avant en arrière et de droite à gauche. Nous avons donc divisé l'expérience en deux séries : dans la première les propositions sont orientées comme la séquence initiale, c'est-à-dire de gauche à droite. Dans la seconde série la séquence initiale est toujours orientée de gauche à droite, mais les propositions sont d'avant en arrière.

Série 1				
Séquence	**Propositions**			
1	2	1	3	6
4	3	2	4	8
6	2	6	3	5
8	7	8	4	5
9	9	7	6	4

Série 2				
Séquences	**Propositions**			
4	4	8	3	2
1	6	2	3	1
6	5	2	6	3
9	9	7	6	4
8	7	8	4	5

Table 3 – Séquences à reconnaître et les propositions

Les séquences à reconnaître et les propositions des deux séries sont résumées dans la table 3. Les séquences à reconnaître ont été choisies afin de varier le nombre d'impulsions, la direction des impulsions et le nombre de parties croissantes et décroissantes. Les séquences à comparer ont été choisies de la même manière.

Ces tests ont été réalisés avec des enfants déficients visuels. Cette particularité a plusieurs implications sur le protocole, les résultats et les analyses. Au niveau du protocole, nous ne pouvons effectuer qu'une seule session à cause de la disponibilité des utilisateurs. De plus les sessions ne peuvent pas durer trop longtemps, d'une part car les utilisateurs ne sont pas disponibles très longtemps, et d'autre part car nous voulons éviter d'introduire un facteur de fatigue. Pour ces raisons, nous avons peu de données sur les utilisateurs. De plus la difficulté à trouver des utilisateurs déficients visuels fait que nous avons une tranche d'âge large, et une variété de déficiences. En particulier les jeunes utilisateurs sont plus lents pour réaliser les tâches, et moins concentrés. C'est la raison pour laquelle il y avait peu de séquences à identifier, et que les propositions étaient restreintes. Chaque utilisateur apporte donc des éléments pour l'analyse, mais il est difficile de raisonner sur le groupe en entier. La faible quantité de données récoltée ne permet pas de faire une étude statistique poussée. Cette étude s'apparie donc plus à une étude de cas. Notre but est donc de déceler les erreurs de conception des séquences en analysant les résultats de chaque utilisateur. Nous avons pour cela enregistré le nombre de fois que chaque séquence a été parcourue en plus des réponses données par les utilisateurs. Ainsi nous pourrons déceler les hésitations des utilisateurs.

En plus des constatations qui constitueront la majeure partie de notre analyse, nous allons vérifier les hypothèses suivantes :

- Une grande partie des erreurs seront commises par les utilisateurs les plus jeunes. Cette hypothèse est motivée par les faits que les utilisateurs les plus jeunes manquent d'expérience, et ont plus difficultés à se concentrer que leurs aînés.
- Il n'y aura pas de différence de reconnaissance entre les deux séries. En effet nous n'utilisons qu'une seule amplitude, et les utilisateurs n'ont eu aucune difficulté à reconnaître les directions dans les expériences précédentes. Nous n'estimons donc pas que l'orientation influe sur ces résultats.

5.3. Résultats

Première série.

Les réponses de chaque utilisateur pour la première série sont reportées dans la table 4. Les utilisateurs ont commis 22 erreurs sur les 55 séquences à reconnaître. Cela correspond à un taux d'erreur de 40%, c'est-à-dire 2 erreurs sur 5 par utilisateur en moyenne. D'une part 10 des erreurs ont été commises alors que l'utilisateur a répondu avant d'avoir parcouru la bonne proposition. D'autre part 5 erreurs ont été commises alors que l'utilisateur a plus parcouru la bonne réponse que celle qu'il a donné. Nous pouvons en conclure que ces utilisateurs ont hésité avec la bonne réponse. Dans 7 autres cas les utilisateurs n'ont lu la séquence à reconnaître que 3 à 5 fois. Ceci est peu en comparaison de la médiane de 8 fois (détails par séquence sur la figure 16, discutée en page 122). Par contre d'après un test de Kruskal-Wallis, aucune différence du nombre de lecture entre séquences n'est significative ($\chi^2 = 1{,}80$, $p = 0{,}77$).

Correct	**Utilisateur (réponse)**										
	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	2	2	2
4	4	3	8	2	4	4	4	2	4	2	4
6	6	6	6	3	2	2	6	6	2	6	2
8	8	8	5	7	8	7	8	8	5	4	7
9	9	9	9	9	9	9	9	7	6	7	9

Table 4 – Réponses des utilisateurs dans la série 1

Il est aussi intéressant de se pencher sur les confusions. Par exemple la séquence 6 a été confondue 4 fois avec la séquence 2. La séquence 8 a été confondue 2 fois avec la séquence 4. Ces confusions ont comme point commun qu'elles ont lieu entre séquences ayant le même nombre d'allers et de retours. Par ailleurs, seulement 2 séquences n'avaient que des impulsions vers le haut. Or 16 des 22 réponses

fausses étaient des séquences avec seulement des impulsions vers le haut. 6 sur 8 d'entre elles ont été données pour les séquences n'ayant que des impulsions vers le haut, et 10 sur 14 pour les autres séquences.

Les jeunes utilisateurs ont eu plus de difficultés à reconnaître les séquences : les 3 utilisateurs les plus jeunes ont fait ensemble 11 erreurs, ce qui correspond à la quantité d'erreurs commises par l'ensemble des 8 autres utilisateurs. Un test de la somme des rangs de Wilcoxon montre que la différence de résultats entre ces trois utilisateurs et les autres est significative ($W = 0{,}5$, $p = 0{,}02$).

Deuxième série.

La table 5 présente les réponses des utilisateurs pour la seconde série. Les utilisateurs 7 et 9 n'ont pas participé à cette série à cause de problèmes de disponibilité. Les utilisateurs ont fait un total de 29 erreurs sur 45 dans cette série, ce qui correspond à 3,22 erreurs sur 5 par utilisateur ou encore 64% d'erreurs. En comparant les résultats des deux séries, on constate que la première série induit significativement moins d'erreurs que la seconde, que ce soit en examinant le nombre d'erreurs par utilisateur ($W = 303$, $p = 0{,}003$) ou le nombre d'erreurs par séquence proposée ($W = 100$, $p < 0{,}001$).

Correct	Utilisateur (réponse)								
	1	2	3	4	5	6	8	10	11
4	4	8	4	4	3	2	4	2	4
1	1	1	1	6	3	3	1	2	2
6	2	3	5	6	2	2	5	3	2
9	9	6	6	6	6	7	6	9	7
8	8	8	5	7	7	4	7	8	8

Table 5 – Réponses des utilisateurs dans la série 2

Cette fois, les utilisateurs ont fait 9 erreurs sur les 29 alors qu'ils n'avaient pas encore lu la bonne proposition. 14 autres erreurs ont été commises alors que l'utilisateur a plus lu la bonne réponse que celle qu'il a répondu. Encore une fois les utilisateurs ont eu des hésitations.

Dans cette série, Les utilisateurs ont lu la séquence à reconnaître plus souvent (figure 16). Le nombre médian de lectures de la séquence à reconnaître est de 12 fois, et la différence entre les deux séries est significative d'après un test de Wilcoxon ($W = 851{,}5$, $p < 0{,}01$). En particulier la différence entre les deux séries avec la séquence 9 est significative ($W = 22{,}5$, $p = 0{,}04$). Par contre, à l'instar

de la première série le test de Kruskal-Wallis ne montre pas de différences significatives entre les séquences ($\chi^2 = 2{,}57$, $p = 0{,}63$).

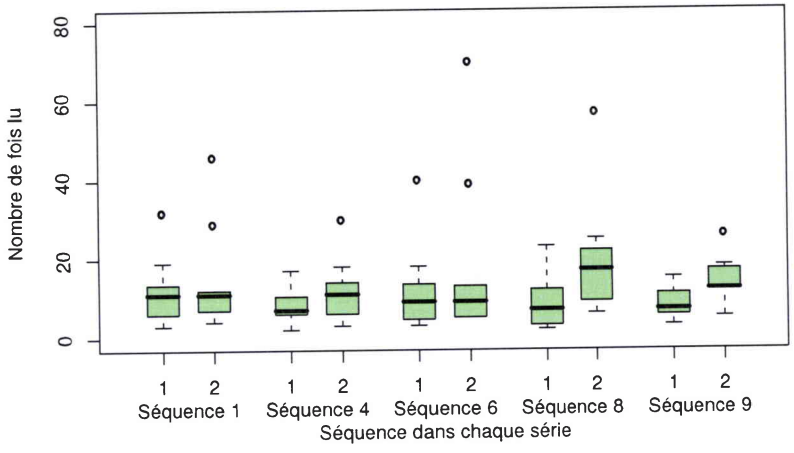


Figure 16 – Nombre de fois que chaque séquence a été lue

L'examen des confusions révèle des résultats similaires à ceux de la série précédente. Les utilisateurs ont confondu deux fois les séquences 4 et 8 : une fois lorsque la séquence 4 était proposée, et une autre fois alors que c'était la séquence 8 qui était proposée. De manière analogue les séquences 1 et 3 ont été confondues deux fois, les séquences 5 et 6 ont été confondues deux fois et les séquences 2 et 6 ont été confondues 4 fois. 14 des 29 mauvaises réponses étaient des réponses avec uniquement des impulsions vers le haut, soit 4 sur 9 pour les séquences où il n'y avait effectivement que des impulsions vers le haut et 10 sur 20 pour les autres.

Dans cette série, la différence entre les âges n'apparaît pas. À part l'utilisateur de 23 ans qui n'a fait que 1 erreur, et deux utilisateurs de 14 et 12 ans qui ont fait 5 erreurs, tous les autres utilisateurs ont fait 3 erreurs chacun.

5.4. Discussion

Les taux d'erreur réalisés dans ces deux séries sont élevés : 40% des réponses de la première série et 64% de la seconde sont erronées. Le taux d'erreur de la seconde série est proche du taux de chance de 80%, ce qui laisse à penser que la tâche est difficile. L'analyse statistique des erreurs effectuées dans les deux séries révèle qu'il y a une différence de résultat entre ces deux séries. L'orientation joue donc un rôle dans l'efficacité des utilisateurs à reconnaître les séquences, et nous devons donc rejeter notre deuxième hypothèse. La complexité et la similarité des séquences proposées n'a pas simplifié cette tâche. Nous chercherons donc à l'avenir à simplifier ces séquences.

Une grosse partie de ces erreurs est due à des excès de confiance des utilisateurs : 19 erreurs sur 51 ont été commises alors que l'utilisateur a répondu avant même de lire la bonne réponse. Une autre partie des erreurs est due à des hésitations : 19 erreurs sur 51 ont été réalisées alors que l'utilisateur a parcouru en majorité la bonne réponse. Il faut donc retenir que les séquences proposées sont trop

proches. Dans le premier cas l'utilisateur répond tôt car il a senti une séquence proche, et dans le second l'utilisateur hésite entre deux séquences qu'il ressent comme proches. Un effort devra être fait pour différencier les séquences.

Le nombre de lectures de la séquence à reconnaître peut expliquer certaines erreurs : dans la première série, les utilisateurs ont fait 7 erreurs alors qu'ils n'avaient lu la séquence à reconnaître que 3 à 5 fois. Dans ces cas les utilisateurs ont pu oublier la séquence à reconnaître et répondre au hasard. Aucune séquence n'a été plus lue que les autres, que ce soit dans la première ou dans la seconde série. Ceci suggère que le problème ne vient pas de la difficulté des séquences elles mêmes, mais de la similarité qu'il y a parmi elles. La différence entre le nombre de fois que les séquences à reconnaître ont été lues dans la première série et le nombre de fois où elles ont été lues dans la seconde est aussi significative, et particulièrement pour la séquence 9. C'est un autre indicateur permettant de montrer que la tâche était plus difficile dans la seconde série que dans la première.

L'examen des confusions des utilisateurs donne des indices sur les modifications à apporter sur les séquences. Plusieurs séquences ont été confondues avec d'autres séquences possédant le même nombre d'allers et de retours. La figure 17 montre un exemple de confusion : les séquences $haut - bas - haut$ et $haut - haut$. L'impulsion vers le bas au milieu des deux impulsions vers le haut de la première séquence peut être confondue avec le retour entre les deux impulsions vers le haut de la seconde séquence. Nous avons plusieurs choix possibles pour remédier à ce problème. Le premier est d'espacer les impulsions au sein d'une séquence. Ainsi le point d'inflexion lors du changement de direction sera plus marqué. Le désavantage de cette solution est que cela agrandit les séquences. Une autre solution serait de n'utiliser que des impulsions vers le haut. Le pouvoir expressif est du coup plus limité. Chaque impulsion codant une quantité d'information, il serait intéressant d'étudier la limite de quantité d'information codable par centimètre.



$haut - bas - haut$ $\qquad$ $haut - haut$

Figure 17 – Confusion entre séquences ayant le même nombre d'allers et de retours

Parmi les autres confusions constatées, on note que les utilisateurs ont souvent tendance à répondre des séquences n'ayant que des impulsions vers le haut. Cela renforce l'idée que les utilisateurs examinent les séquences en ne comptant que les « bosses », sans se soucier des directions. Cette constatation laisse supposer que la seconde solution donnée précédemment serait plus adaptée que la première.

Enfin même si les utilisateurs les plus jeunes (entre 9 et 10 ans) ont eu plus difficultés que leurs aînés (entre 12 et 23 ans) sur la première série, cet écart s'est compensé dans la seconde série. Nous devons donc rejeter la première hypothèse.

6. Conclusion

La technique d'interaction présentée dans ce chapitre utilise le retour de force afin d'indiquer des informations dans un environnement multimodal. Cette technique est basée sur des impulsions données par un périphérique de pointage à retour de force tel qu'un PHANToM. Ces impulsions sont paramétrables en forme, longueur, amplitude et direction. La longueur peut se décliner soit de manière temporelle, donnant des impulsions guidées ; soit de manière spatiale, donnant des impulsions semi-guidées. Dans un premier temps nous avons conçu et évalué des impulsions guidées puis semi-guidées, en faisant varier les paramètres de direction et d'amplitude. Dans un deuxième temps nous avons évalué des séquences d'impulsions semi-guidées appelées PICOB. Ces PICOB sont des post-it haptiques par codes-barres. L'idée est de représenter l'information par des bosses que l'on ressent en passant dessus, comme si on passe un stylo sur un peigne.

Trois expériences ont été réalisées pour évaluer les impulsions guidées. Dans la première nous avons examiné la reconnaissance de la direction avec plusieurs valeurs d'amplitude. Dans la deuxième expérience nous avons étudié la reconnaissance du paramètre d'amplitude. Enfin dans la troisième nous avons expérimenté la reconnaissance simultanée de direction et d'amplitude. Les résultats montrent que la direction est aisément reconnaissable, que ce soit avec une valeur d'amplitude, ou quand plusieurs amplitudes sont possibles. Les utilisateurs ont réussi à distinguer trois valeurs d'amplitude lorsqu'une seule direction était possible. Par contre ils ont eu quelques difficultés à reconnaitre les amplitudes lorsque toutes les directions étaient proposées. D'après certains utilisateurs l'amplitude ressentie est différente selon la direction de l'impulsion. Ce phénomène s'explique par le fait que la force exercée par le PHANToM est inférieure à la force du poignet, et que donc celui-ci peut s'opposer au mouvement du stylet. Cette résistance n'est pas la même dans toutes les directions. En effet pour les directions en avant et en arrière, le stylet se déplace dans le prolongement du bras, ce qui fait que la résistance est potentiellement élevée. Par contre de droite à gauche, le mouvement est perpendiculaire au bras donc la résistance est plus faible. De haut en bas la gravité exerce une force, ce qui peut aussi pousser ou tirer le stylet dans une direction autre que celle exercée par le système. Nous remarquons que les utilisateurs font moins d'erreurs lorsqu'il y a deux amplitudes que quand il y en a trois. De plus les utilisateurs n'ont pas reçu significativement plus d'informations dans les séries à trois amplitudes que dans la série à deux amplitudes. Nous en déduisons donc qu'il n'est pas nécessaire d'utiliser plus de deux amplitudes. Nous n'utiliserons donc au plus que 6 directions et 2 amplitudes pour les impulsions guidées.

Les impulsions semi-guidées ont été évaluées dans deux expériences. La première expérience vise à examiner les directions à utiliser, et la deuxième sert à investiguer sur la reconnaissance simultanée de la direction et de l'amplitude de ces impulsions. Le choix des directions est plus restreint que dans le cas des impulsions guidées car un axe est déjà utilisé pour l'exploration. De plus les tests pilotes informels indiquent que la reconnaissance de la direction semble plus difficile dans ce cas qu'avec les impulsions guidées. Nous avons aussi testé deux valeurs d'amplitude basées sur les résultats précédents afin de savoir si ces valeurs sont aussi confortables avec ce type

d'impulsions. Nous avons donc réalisé quatre séries, deux avec deux directions et deux autres avec quatre directions. Les résultats montrent des taux d'erreur faibles, et surtout il n'y a ni de différence significative entre les séries ayant un nombre de directions différentes, ni entre les séries utilisant des amplitudes différentes. L'analyse de la quantité d'information transmise indique que les utilisateurs ont reçu significativement plus d'information avec les impulsions à 4 directions qu'avec celles à 2 amplitudes. Nous avons donc utilisé 4 directions dans l'expérience suivante. Deux séries ont été testées : dans la première les impulsions avaient 4 directions et 2 amplitudes, et dans la seconde elles avaient 4 directions et 3 amplitudes. Les utilisateurs n'ont pas plus de difficultés à reconnaître les directions dans une série que dans l'autre. Par contre il a été plus difficile pour eux de reconnaître les directions dans la deuxième série que dans l'expérience précédente, ce qui signifie que la présence de trois amplitudes perturbe la reconnaissance de direction. Le taux d'erreur dans la série à deux amplitudes est raisonnable, alors que celui de la série à trois amplitudes est élevé. Cette différence est significative. Enfin l'analyse de la quantité d'information transmise montre qu'il n'y a pas de différences significatives entre la quantité d'information transmise dans la série à 2 amplitudes et la série à 3 amplitudes. Pour ces trois raisons, nous n'utiliserons au maximum que deux amplitudes. En complément, nous retiendrons une remarque de certains utilisateurs disant que l'amplitude est plus facile à reconnaître quand on déplace le stylet lentement, et la direction est plus facile à reconnaître quand on déplace le stylet rapidement. D'autres utilisateurs remarquent que la perception de l'amplitude diffère selon l'amplitude de l'impulsion ressentie précédemment. Ceci s'explique par le fait que lorsque l'amplitude est grande on a tendance à tenir le stylet plus fermement que si elle est petite. Et l'amplitude ressentie est plus faible si on tient le stylet plus fermement.

Enfin l'évaluation suivante concerne les séquences d'impulsions semi-guidées. Les impulsions sélectionnées n'utilisent que deux directions, perpendiculaires au plan horizontal dans lequel se trouvera le circuit électrique. D'après nos besoins, une seule amplitude est nécessaire. Nous n'en avons donc pas utilisé plus afin d'éviter de compliquer la tâche inutilement. Les séquences sont construites en concaténant de une à quatre impulsions semi-guidées. Deux séries de tests ont été réalisées : dans une les réponses étaient présentées de droite à gauche et dans la seconde d'avant en arrière. Les utilisateurs de ces tests correspondent au public visé par l'application évoquée précédemment : des enfants déficients visuels. La tâche s'est avérée plus difficile que prévue. Les utilisateurs ont réalisé 40% d'erreurs dans la première série et 64% dans la seconde, pour un seuil de chance à 80%. La différence de reconnaissance entre ces deux séries est significative, ce qui indique que l'interprétation des séquences dans une orientation différente apporte une difficulté supplémentaire. L'analyse des erreurs montre que d'une part les utilisateurs ont hésité, et d'autre part que certaines séquences sont trop proches. L'analyse des confusions permet de mettre en évidence quelques problèmes de conception. En effet les séquences ayant le même nombre de sommets et de creux sont facilement confondus. De plus les utilisateurs ont tendance à compter les sommets, et donc donnent en réponse majoritairement les séquences n'ayant que des impulsions vers le haut. Nous avons imaginé deux solutions pour ce problème : la première est d'insérer des pauses entre les impulsions, et la seconde est de n'utiliser que des impulsions vers le haut.

Au niveau du travail restant à effectuer dans cette voie, nous souhaitons étudier les séquences d'impulsions guidées. Autant dans le cas des impulsions guidées que de celui des impulsions semi-guidées nous souhaitons examiner les autres paramètres des impulsions. Par exemple pour la forme nous pouvons essayer d'utiliser des impulsions en triangle voire en carré. Nous n'avons pas fait varier la longueur des impulsions, que ce soit la durée des impulsions guidées ou la longueur des impulsions semi-guidées. Il serait intéressant d'examiner s'il est possible d'utiliser des rythmes en faisant varier ce paramètre.

Dans le chapitre suivant, nous allons dans un premier temps détailler la conception d'une architecture multimodale permettant de faciliter le développer d'applications multimodales. Nous utiliserons ensuite cette architecture, les icônes tactiles du chapitre précédent et les icônes kinesthésiques du chapitre présent afin de de représenter des schémas de circuits électriques de manière haptique.

Chapitre IV

Applications multimodales pour déficients visuels

Chercher n'est pas une chose et trouver
une autre, mais le gain de la recherche,
c'est la recherche même.
Saint Grégoire de Nysse

Résumé

Dans ce chapitre nous décrivons tout d'abord l'architecture multimodale que nous avons conçue. Elle est construite sous forme d'une collection d'agents communiquant par le biais d'un bus logiciel. Elle est dotée d'un framework pour la gestion de périphériques d'entrée et de sortie permettant de développer aisément des applications multimodales. Nous nous sommes concentrés sur la modalité haptique en sortie, même si l'architecture offre des services utilisant d'autres modalités. Ensuite nous étudions le développement incrémental d'une application d'exploration de schémas de circuits électriques. Nous décrivons successivment deux prototypes, puis l'application finale qui utilise l'architecture. Ce développement s'est fait en parallèle des expérimentations sur les techniques d'interaction des chapitres précédents. C'est pourquoi nous les avons ajoutées au fur et à mesure. Ainsi les techniques d'interaction ayant déjà été testées, les évaluations que nous avons réalisées visent à nous renseigner sur leur intégration au sein de l'application, et la manière de les combiner.

1. Introduction

Le développement d'applications multimodales utilisant plusieurs périphériques est une tâche compliquée. Il faut d'une part maîtriser les aspects techniques comme la programmation de chaque périphérique. D'autre part il faut s'intéresser aux aspects conceptuels des techniques d'interaction offertes. Comment sont combinées les modalités, quel est leur niveau d'abstraction, etc. La première partie de ce chapitre traite de l'architecture multimodale que nous avons conçue. Nous étudierons tout d'abord son système d'agent, ainsi que le bus logiciel qu'ils utilisent pour communiquer. Ensuite nous examinerons le framework multimodal que nous avons conçu afin de simplifier la gestion des périphériques d'entrée et de sortie. Dans la seconde partie de ce chapitre nous étudierons la conception et l'évaluation d'une application d'exploration de schémas de circuits électriques. Nous y étudierons les deux prototypes, ainsi que l'application finale qui utilise l'architecture décrite dans la première partie.

2. Architecture multimodale et collaborative MICOLE

Il existe de nombreux systèmes multimodaux pour aider des utilisateurs déficients visuels à réaliser certaines tâches. Nous avons évoqué certains de ces systèmes dans la section 4 du chapitre I. Le système PC-Access utilise le retour de force et le retour sonore pour reproduire les objets de l'interface [145]. Le retour de force offert par ce système est cependant limité car il n'utilise qu'un pantographe. D'une part ce périphérique n'est pas répandu, et de plus il ne possède que deux degrés de liberté. Le système TACIS quant à lui utilise un grand touchpad sur lequel est placé un feuille de papier en relief sur laquelle est imprimée une vue statique du système à explorer [61]. Ce système utilise aussi un rendu sonore, avec notamment une synthèse vocale. Son inconvénient principal est son côté statique. Il n'est pas possible de placer et de modifier des informations en temps réel, ce que permet un système informatique tel que celui que nous allons décrire plus loin. Le système TeDub permet d'analyser, puis de parcourir plusieurs sortes de schémas à l'aide d'un clavier, d'un joystick à retour de force et d'un retour sonore. Le retour haptique offert par ce système est encore plus limité que celui du système PC-Access. Dans notre système l'haptique a un rôle principal, avec du retour tactile et du retour de force. En effet les joysticks à retour de force ont une faible résolution en comparaison d'un périphérique comme le PHANToM.

Saarinen *et al.* [153] ont développé une architecture multimodale basée sur un système d'agents. Cette architecture a pour but de développer des environnements d'apprentissage pour enfants déficients visuels. Le système d'agents a plusieurs buts. Premièrement il permet de répartir la charge de calcul sur plusieurs machines si nécessaire. Deuxièmement il permet à l'architecture d'être flexible et de pouvoir être étendue, pour supporter du nouveau matériel par exemple. Troisièmement le développement peut être incrémental : les fonctions vitales étant développées en premier, et les fonctions annexes ensuite. L'architecture est divisée en trois composants principaux. Le premier est le contrôleur. Son rôle est de faire le lien entre l'utilisateur et le système d'agents. Une partie des agents, agissant pour le contrôleur, ont pour rôle de recueillir et envoyer des informations de la part de l'utilisateur. L'API Reachin [2] est utilisée afin de contrôler des périphériques tels que le PHANToM. Les applications étant formées de plusieurs scènes, le contrôleur assure aussi le lien entre ces scènes. Le deuxième composant est le canal de messages : il enregistre les agents et les conteneurs d'agents du système, ainsi que les services qu'ils proposent. Ainsi chaque agent peut envoyer un message, en précisant le service attendu. Les agents fournissant le service en question reçoivent une copie du message. Enfin le troisième composant est le conteneur d'agents, et de manière plus générale l'ensemble des agents. Le conteneur d'agents possède une boîte aux lettres pour chaque agent, et permet à chaque agent d'envoyer des messages à d'autres agents. Les conteneurs d'agents se déclarent au canal de message à leur création, et reçoivent un identifiant unique. Ceci permet aux conteneurs de communiquer entre eux. Après sa création, un conteneur d'agents instancie les agents dont il a la charge, et qui sont configurés dans un fichiers. Il est aussi possible de créer des agents dynamiquement. Cette architecture a été utilisée pour créer l'environnement d'apprentissage du système solaire évoqué dans la section 4.3.1 du chapitre I.

L'architecture MICOLE que nous avons conçue en collaboration avec l'université de Tampere [139] est basée sur le principe de celle que nous venons de décrire. Le système d'agents et de communication ont été simplifiés, et une librairie d'agents a été développer afin de permettre de développer de nombreuses applications multimodales sans duplication de code. La notion de multimodalité a été introduite dans la section 2.2 du chapitre I. Il s'agit de la capacité d'un système à communiquer avec ses utilisateurs à l'aide de plusieurs canaux de communication et à manipuler l'information qui circule. La multimodalité est utilisée afin de compléter ou remplacer le canal visuel avec d'autres canaux tels que le canal auditif et le canal haptique. Les systèmes collaboratifs permettent à plusieurs utilisateur d'agir ensemble sur le système. Cet aspect est important dans le contexte de notre objectif car il correspond à plusieurs types de scénarios typiques :

- Un enseignant montre à un élève ce qu'il faut faire.
- Un élève travaille avec un autre pour réaliser la tâche.

Que ce soit pour la multimodalité ou pour la collaboration, les applications que nous souhaitons développer ont besoin de gérer le plus grand nombre possible de périphériques. Dans le cas de la multimodalité le but est de proposer une grande variété de retours : forces, vibrations, sons, parole, etc. Chaque retour peut être rendu par un ou plusieurs périphériques, et donc nous devons faire en sorte que notre architecture prenne en charge le plus de périphériques possibles, et permette d'en gérer d'autres facilement. Dans le cas de la collaboration, il faut que chaque utilisateur puisse agir sur le système. Or le type d'informations envoyées et reçues par l'utilisateur dépend du type d'utilisateurs. Un utilisateur voyant est habitué à utiliser la souris et l'écran, alors qu'un utilisateur non-voyant sera plus à l'aise avec une tablette braille, un PHANToM ou une synthèse vocale. L'utilisation de plusieurs périphériques est contraignant pour un développeur car il doit se documenter sur l'interface de programmation (API) de chaque périphérique. De plus, il est parfois difficile de les interfacer ensemble. Nous avons conçu notre architecture de manière à ce qu'elle puisse servir d'intermédiaire entre le développeur d'application et les API des différents périphériques. Ainsi avec une seule syntaxe il est possible de communiquer avec plusieurs périphériques. D'un point de vue technique l'API est programmée en C++, et l'API Reachin [2] a été utilisée pour piloter les périphériques haptiques tels que le PHANToM, ainsi que pour gérer la scène.

2.1. Le système multi-agents

Nous avons décidé de construire notre architecture sur un modèle multi-agents communiquant grâce à un bus logiciel. Le système multi-agents permet d'avoir un système souple sur lequel il est possible d'ajouter des entités à volonté. Ainsi si chaque périphérique est géré par un ou plusieurs agents, il suffit d'instancier ces agents pour gérer ce périphérique. Le système de communication par bus permet aux agents de communiquer simultanément avec plusieurs autres agents. En particulier si nous voulons afficher la même information parallèlement sur plusieurs périphériques, il suffit d'envoyer un message contenant l'information qui sera traité par plusieurs agents, chacun gérant l'affichage sur un périphérique.

2.1.1. Les agents

Dans la section 2 du chapitre I, nous avons évoqué quelques systèmes d'agents. Nous y avons repris la définition d'un agent donnée par Coutaz *et al.* [42] :

Définition 1 Un *agent* est un utilisateur, un système ou une partie d'un système capable d'initier une action.

Les systèmes d'agents ont de nombreux avantages. Dans un premier temps le fait que chacun représente une tâche propre permet à chaque agent d'être exécuté de manière plus ou moins indépendante. Des contraintes peuvent apparaître au moment où plusieurs agents doivent communiquer entre eux. L'intérêt est que chaque agent peut être exécuté par un thread différent, voire un processus ou même une machine différente. Cette indépendance a un autre avantage : si on modifie un agent, le reste des agents ne peut être affecté que si la modification concerne la communication.

La notion d'agent ci-dessus est utilisée par Coutaz *et al.* afin de définir comment un ou plusieurs agents sont mis en jeu afin de réaliser une tâche à l'aide des propriétés CARE [42]. Ainsi si plusieurs agents exécutent tous une partie de la tâche, on dit alors qu'ils sont complémentaires. Si une tâche ne peut être réalisée que par un agent bien précis, elle est alors assignée. Par contre si une tâche peut être réalisée par plusieurs agents indépendamment on dit alors qu'elles sont équivalentes. Et enfin si une tâche peut être réalisée par plusieurs agents séquentiellement ou parallèlement on dit alors qu'elles sont redondantes.

Coutaz *et al.* [41] utilisent cette même notion d'agent pour créer l'architecture PAC-Amodeus, qui combine le modèle Arch [4] et le modèle PAC [38]. Les agents PAC permettent de diviser les tâches de l'application en trois catégories : la présentation pour ce qui est entrées et sorties, l'abstraction pour le noyau fonctionnel et le contrôle pour ce qui correspond à la communication entre la présentation et l'abstraction. Le modèle Arch quant à lui relie les interactions avec l'utilisateur au noyau fonctionnel de l'application grâce à un contrôleur de dialogue. Ce contrôleur de dialogue est composé d'agents PAC ayant chacun les trois facettes : présentation, abstraction et contrôle. La facette abstraction est reliée au noyau fonctionnel, la facette présentation est reliée aux entrées et sorties, et le contrôle permet à plusieurs agents de communiquer entre eux.

La plupart des systèmes d'agents possèdent un serveur central qui se charge de l'initialisation et de la connexion des agents entre eux. De plus les agents appartiennent à un conteneur qui gère la communication entre les agents, ainsi que leur cycle de vie. La *FIPA* (Fondation for intelligent Physical Agents) [58] est un programme de standardisation de systèmes d'agents. Plusieurs systèmes d'agents ont été développés en se basant sur les recommandations FIPA. JADE en est un exemple [15]. Ce système d'agents est implémenté en JAVA, et suit les spécifications FIPA97. Le système peut être réparti sur plusieurs machines, avec un conteneur d'agents s'exécutant sur chacune d'elles. Un conteneur principal fournit un dictionnaire des services et un système de communication entre les agents. Chaque agent possède sa propre thread, ce qui lui permet d'initier des actions en plus de pouvoir réagir à des évènements. Chaque agent se charge d'une ou plusieurs

tâches appelées comportements. Dans le cas d'agents ayant un seul comportement, celui-ci est soit exécuté une fois, soit exécuté indéfiniment de manière cyclique. Dans le cas des agents à plusieurs comportements, l'agent s'occupe de gérer l'ordonnancement. Les deux méthodes proposées sont l'exécution séquentielle et le tourniquet (round-robin). L'idée générale de ce système est intéressante. Cependant elle ne convient pas à nos besoins car elle utilise une machine virtuelle JAVA, ce qui n'est pas adapté à des systèmes temps réels comme celui que nous avons réalisé.

ABLE est un autre système d'agent basé sur les spécifications FIPA et développé en JAVA [21]. Comme dans la plupart des autres systèmes d'agents, les agents appartiennent à des conteneurs. Par contre ce système se démarque des autres par le fait que la structure est récursive car chaque agent est un conteneur d'agents, c'est-à-dire qu'un agent peut en contenir d'autres. La communication entre agents peut se faire de trois manières différentes. Pour la première, les agents peuvent demander à être informés des changements d'état d'un autre agent. Avec la deuxième, les agents peuvent être reliés de telle manière que la sortie d'un agent soit redirigée vers l'entrée d'un autre agent. Enfin la troisième méthode consiste à faire partager la même propriété par plusieurs agents. Ainsi si un agent modifie la propriété, les modifications sont répercutées pour les autres agents. Cette structure récursive est intéressante. Cependant ce système utilise aussi JAVA, et n'est donc pas adapté à nos besoins.

OAA (Open Agent Architecture) est un système d'agent dédié aux systèmes multimodaux [117]. Ce système utilise une structure classique déjà évoquée, avec un serveur central qui répartit les tâches et annonce les services proposés par les agents connectés au système. Il est spécialisé dans les entrées, avec un système de reconnaissance vocale, de gestes et d'écriture. Notre approche est différente car nous nous focalisons sur les modalités en sortie.

Nous avons conçu un système d'agents en s'inspirant de la conception des systèmes d'agents existants. Cette architecture s'inspire en particulier de l'architecture développée par Saarinen *et al.* [153]. Le but est de faire collaborer des agents et des conteneurs d'agents grâce à un canal de communication. Chaque conteneur peut être localisé sur des machines différentes. Un des conteneurs est appelé le Contrôleur. Il assure la liaison avec les librairies qui permettent de communiquer avec les périphériques. L'API Reachin [2] fournit une scène 3D en OpenGL, dont les composants sont organisés sous forme de structure VRML à laquelle sont ajoutées des propriétés haptiques. Ainsi la scène peut être explorée avec des périphériques de pointage à retour de force tel qu'un PHANToM ou un Omega. La librairie BASS est utilisée pour un retour sonore [13].

2.1.2. Le bus

Nous avons choisi le bus logiciel comme méthode de communication. Ceci nous permet de faire communiquer chaque agent avec les autres agents simultanément. Cet aspect est important dans notre contexte car nous voulons avoir la possibilité d'afficher simultanément la même information sur plusieurs périphériques. L'avantage de ce système est que l'agent qui génère le message à afficher n'a qu'à envoyer un seul message, qui sera récupéré par plusieurs autres agents.

Le système de communication est étroitement lié aux agents, ce qui fait que l'implémentation des agents inclut des méthodes permettant d'envoyer et de recevoir des messages. Dans un premier temps nous avons utilisé le bus Ivy [33]. Ce bus est un canal de message sur lequel une entité peut écrire un message en texte, ou exécuter une action en réaction à une classe de message. Les messages sur lesquels une entité réagit sont spécifiées sous formes d'expressions régulières. Des parties des messages peuvent être capturées et passées en paramètre à la fonction appelée lorsqu'un message correspondant à l'expression régulière transite sur le bus.

Définition 2 Une fonction exécutée lorsqu'un message correspondant à une expression régulière donnée transite sur le bus est appelée *callback*.

Définition 3 La relation entre une expression régulière et un callback est appelée un *binding*.

La figure 1 montre un exemple de communication entre deux agents. L'agent A envoit le message `IN MSE: pos=(5,20);` sur le bus. L'agent B possède un binding qui permet d'exécuter la fonction CB lorsqu'un message correspondant à l'expression régulière `IN MSE: pos=\((.*),(.*)\);` transite sur le bus. Le message envoyé par l'agent A correspond donc la fonction CB est appelée, avec les paramètres `5` et `20` récupérés grâce aux parenthèses dans l'expression régulière. Le message continue son transit sur le bus, et peut être récupéré par d'autres bindings créés soit par le même agent soit par un autre agent.

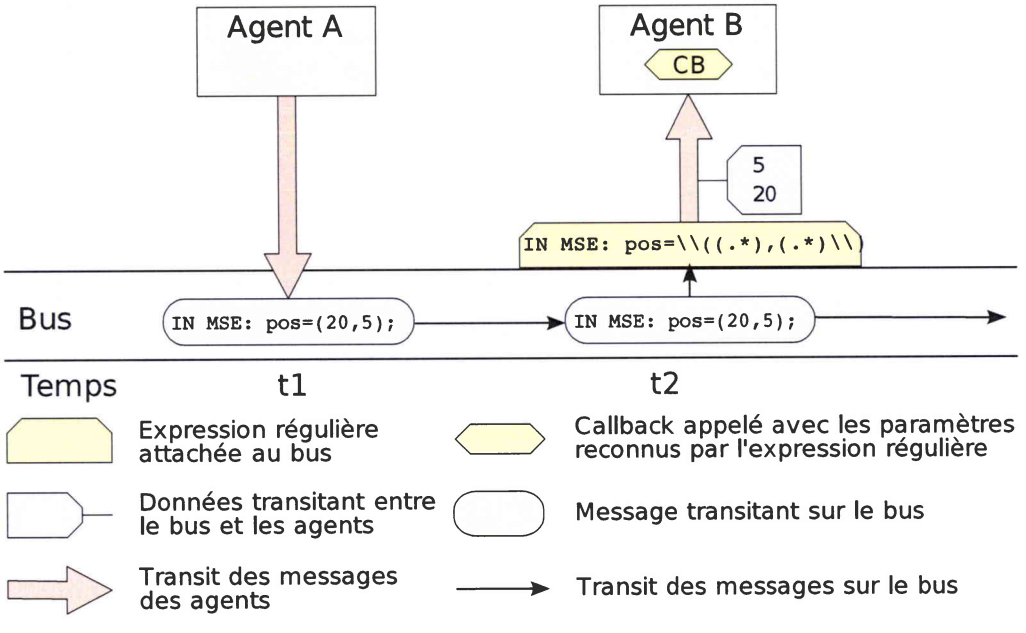


Figure 1 – Système de communication entre agents

Envoi de messages.

Chaque agent possède donc d'une part une méthode `sendMessage` pour envoyer un message sur le bus. En reprenant l'exemple de la figure 1, l'agent A envoie un message contenant les coordonnées

de la souris grâce au code de la figure 2. La syntaxe du message est définie par le concepteur de l'agent. En l'occurrence l'agent qui envoie les coordonnées de la souris appartient à l'API, donc la syntaxe utilisée est celle définie dans ses spécifications.

```
... AgentA::...(...)
{
  ...
  sendMessage("IN MSE: pos=(5,20);");
  ...
}
```

Figure 2 – Exemple de message envoyé par un agent

Création de binding.

Chaque agent possède aussi une méthode `bindMessage` afin de créer un binding entre une expression régulière et une de ses méthodes. L'agent B de notre exemple souhaite être informé des changements de coordonnées dans la souris. Le code correspondant est visible sur la figure 3. Il crée donc un binding pour recevoir tous les messages respectant la syntaxe convenue pour les coordonnées de la souris. La notation des expressions régulières utilisée est similaire à la syntaxe utilisée par la plupart des systèmes. Les parenthèses sont utilisées afin de définir les morceaux de chaîne données en paramètre au callback[1]. La syntaxe du callback est fixe, et comprend trois paramètres :

- `app` correspond au bus sur lequel est créé le binding.
- `argc` est le nombre de paramètres récupérées par l'expression régulière.
- `argv` est un tableau contenant les paramètres récupérés par l'expression régulière.

Dans Ivy, les messages peuvent être envoyés par une application, et les callbacks peuvent être créés sur une autre machine. Ceci peut sembler être un avantage dans notre cas et ainsi faciliter la création d'applications multimodales distribuées. Cependant ce système crée plusieurs sockets par binding. Il est donc conçu pour n'utiliser que peu de bindings et servir de canal de communication entre plusieurs programmes. Or dans notre cas il peut y avoir beaucoup de bindings : chaque agent ayant besoin des coordonnées d'un périphérique doit créer un binding, chaque agent utilisant des données d'un autre agent aussi, etc. Une application basée sur notre architecture utilise rapidement plusieurs dizaines de bindings, ce qui donne rapidement près d'une centaine de sockets. La consommation de ressources liées à la communication devient très importante et pénalise notre système. De plus nous avons besoin d'un temps de réponse très rapide pour les périphériques haptiques comme nous allons le voir plus loin. C'est pour cette raison que nous avons ré-implémenté la partie d'Ivy qui nous intéresse, sans utiliser de sockets pour optimiser la vitesse. Au passage nous avons pu vérifier

[1]Si la chaîne contient une parenthèse il faut l'échapper avec deux antislash : un pour le compilateur C++, et un pour l'interpréteur d'expressions régulières

```
AgentB::AgentB()
{
  ...
  bindMessage("^IN MSE: pos=\\((.*),(.*)\\);", \
    BUS_CALLBACK_OF(AgentB, monCallback));
  ...
}

void AgentB::monCallback(MicoleBus *app, int argc, const char **argv)
{
  // argc == 2;
  // argv[0] == "5"
  // argv[1] == "20"
}
```

Figure 3 – Exemple de binding créé par un agent

l'abstraction des applications basées sur l'architecture, car en changeant le bus les applications n'ont pas eu besoin d'être modifiées.

2.1.3. Les agents threadés

Deux catégories d'agents ont été conçus : les agents threadés et les agents non-threadés. Les agents non-threadés sont les agents de base qui ont un comportement passif : ils n'agissent que lorsque leurs callbacks sont appelés. À l'inverse les agents threadés possèdent leur propre boucle d'exécution. Ces agents peuvent initier des comportements en envoyant des messages sur le bus à partir de leur boucle d'exécution. Typiquement ces agents sont utiles pour les agents qui interrogent les périphériques afin d'obtenir leurs coordonnées. Le code de la thread est écrit dans la méthode `run` que le développeur doit surcharger. La figure 4 montre celle de l'agent qui envoie les coordonnées de la souris.

La thread boucle tant que l'agent est marqué comme actif. Dans cette boucle l'agent récupère les coordonnées de la souris, construit un message avec ces coordonnées en respectant la syntaxe prédéfinie, et envoie le message sur le bus.

2.1.4. Le réseau

Le fait d'avoir développé notre propre système de bus basé sur les mécanismes d'Ivy mais sans socket nous a privé de la possibilité de connecter plusieurs applications voire plusieurs machines au bus. Pour compenser cette perte nous avons créé des agents permettant à plusieurs bus de se connecter et communiquer entre eux. Ainsi en créant un agent `NetworkAgent` on précise l'adresse du bus qui joue le rôle de serveur. Si aucun serveur n'est présent, le bus courant devient le serveur et d'autres bus peuvent se connecter dessus. Chaque bus créé est donc un conteneur d'agents. Ensuite pour envoyer un message aux autres bus il suffit de le préfixer par la chaîne « `Net:` ».

```
void MouseSenderAgent::run()
{
  while(getState() == ACTIVE)
  {
    Vec2f p = _mouseMoveListener->get();
    ...
    sendMessage("IN MSE : pos=(" + p.x + ", " + p.y + ");");
    ...
  }
}
```

Figure 4 – Exemple d'agent threadé

À titre d'illustration complémentaire au mécanisme du bus et d'explication du fonctionnement du réseau, la figure 5 montre les mécanismes principaux du fonctionnement de l'interface réseau. Au début de la création de cet agent, on initialise la socket. La méthode `initSocket` tente de se connecter à l'adresse `domaine` sur le port `port`. Si la connexion échoue c'est qu'il n'y a aucun serveur à cette adresse ou qu'il n'est pas joignable. Un serveur est donc créé en local. Ensuite on crée un binding sur les messages commençant par « `Net:` », le reste étant capturé afin d'être passé en paramètre au callback. Ce callback est très simple : il appelle une méthode qui envoie la chaîne capturée sur la socket initialisée ci-dessus. Enfin la boucle de la thread de l'agent est lancée. Cette boucle se contente de lire de manière bloquante sur la socket. Quand un message est reçu, il est envoyé sur le bus.

2.2. Le framework multimodal

L'intérêt de cette architecture consiste non seulement en sa conception, mais aussi en la collection d'agents facilitant le développement d'applications multimodales. En effet sans ce genre d'architecture il faut utiliser autant d'API et de pilotes que de périphériques à utiliser. Cette architecture implémente des agents utilisant chacun le driver ou l'API approprié pour communiquer avec le périphérique. De plus nous avons intégré des techniques d'interaction telles que celles que nous avons développées dans les chapitres II et III. Nous avons aussi inclus des techniques de guidage pour des périphériques tels que le PHANToM. Enfin la méthode utilisée pour gérer les périphériques d'entrée et de sortie décrite ci-dessous peut être utilisée pour implémenter d'autres agents gérant d'autres périphériques.

Nous avons défini une convention de conception pour les agents d'entrée et les agents de sortie. Cette convention permet d'uniformiser à la fois les agents nécessaires pour gérer un périphérique d'entrée ou de sortie, et la syntaxe des messages permettant de communiquer avec eux. Cette convention est résumé sur le schéma de la figure 6.

```
NetworkAgent::NetworkAgent(char * domaine, int port) : ...
{
  initSocket(domaine, port);
  bindMessage("^Net: (.*)$", \
    BUS_CALLBACK_OF(NetworkAgent, netCallback));
  start();
}

void NetworkAgent::netCallback(MicoleBus *app,
                               int argc, const char **argv)
{
  sendOnNetwork(string(argv[0]));
}

void NetworkAgent::run ()
{
  char *buffer = new char[1024];
  while(getState() == ACTIVE)
  {
    recv(_sock, buffer, 1023, 0);
    sendMessage(buffer);
  }
  closesocket(_sock);
  ...
}
```

Figure 5 – Fonctionnement du réseau

2.2.1. Les agents d'entrée

Pour gérer les entrées nous définissons deux types d'agents : les *agents sender* et les *agents input*.

Les agents sender.

Le rôle de l'agent sender est d'interroger le driver du périphérique d'entrée et de renvoyer les informations reçues sur le bus, selon la syntaxe définie dans la figure 7. La ligne commence par `IN` afin de permettre à un agent désirant recevoir toutes les informations entrantes de les recevoir grâce à une simple expression régulière. Ensuite il y a un identifiant de périphérique, ce qui permet de recevoir tout ce qui provient d'un périphérique particulier. Enfin après les double points les paramètres et leurs valeurs sont séparées par un point-virgule. Dans l'exemple présenté sur la figure 4 nous avons montré le fonctionnement de l'agent sender de la souris.

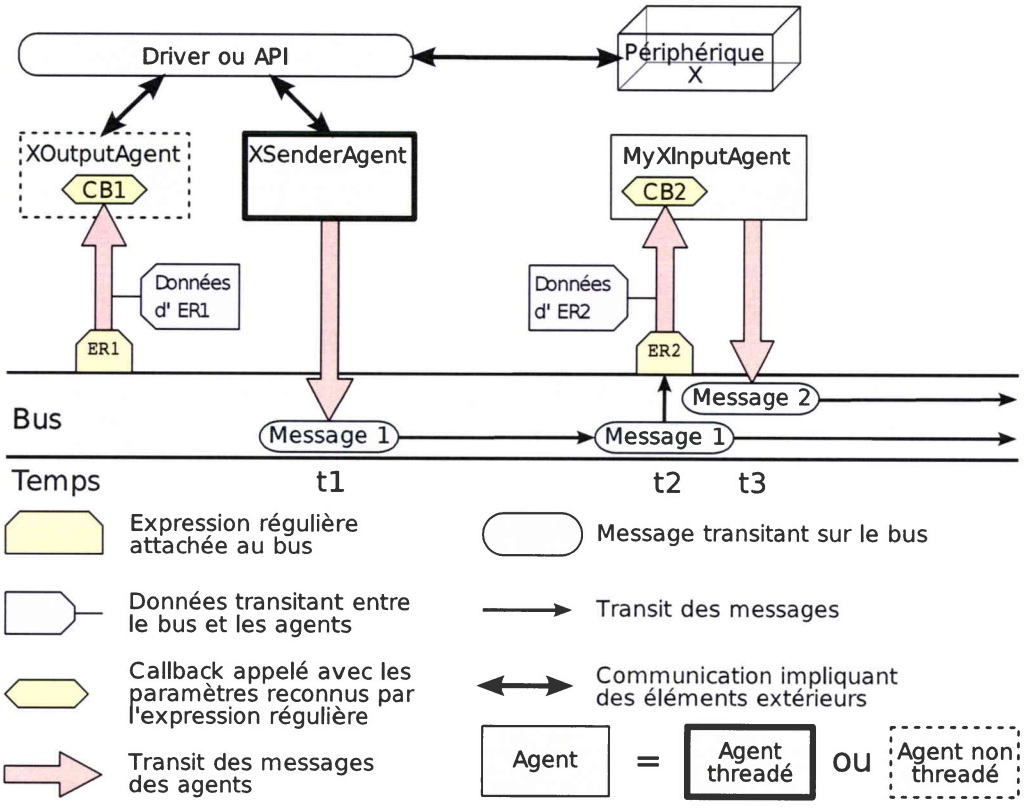


Figure 6 – Agents d'entrée et sortie

```
IN PERIPHERIQUE : par1=val1; par2=val2; ...; parn=valn;
```

Figure 7 – Syntaxe des périphériques d'entrée

Les agents input.

L'agent input est un agent de l'application qui utilise les messages ci-dessus. Afin de faciliter et d'accélérer la tâche de développement, nous développons un agent input générique sous forme de classe abstraite. Cet agent générique crée des bindings pour tous les messages envoyés par le sender correspondant et se charge de stocker les valeurs des différentes informations provenant du périphérique dans des données membre. De plus cette classe doit fournir des méthodes événementielles appelées lorsque ces informations sont modifiées. Ainsi le concepteur d'application n'a qu'à hériter la classe de cet agent générique, et peut utiliser directement les valeurs, et réagir aux évènements sans se soucier des messages du bus. Il peut cependant implémenter lui même les bindings pour un contrôle plus fin.

La figure 8 montre l'exemple d'un agent input pour la souris. Nous ne détaillons que la gestion des coordonnées de la souris. Le binding est créé dans le constructeur, le callback récupère les coordonnées et appelle la méthode événementielle. Celle-ci est virtuelle et ne fait rien par défaut : c'est au

```
class MouseInputAgent : public MicoleAgent
{
  public:
    MouseInputAgent()
    : MicoleAgent("MouseInputAgent","InputAgent")
    {
      bindMessage("^IN MSE : pos=\\((.*), (.*)\\);$",\
        BUS_CALLBACK_OF(MouseInputAgent, handleMouseMove));
      ...
    }

    void handleMouseMove(MicoleBus *app, int argc, const char **argv)
    {
      position.x = atof(argv[0]);
      position.y = atof(argv[1]);
      onMove(position.x, position.y);
    }

    virtual void onMove(const float &x, const float &y){}

  protected:
    Vec2f position;
};
```

Figure 8 – Agent input pour la souris

concepteur de l'application de la surcharger pour définir son comportement. La gestion des boutons n'est pas détaillée, mais fonctionne de manière analogue.

2.2.2. Les agents de sortie

Un seul agent est nécessaire pour gérer les sorties. Ce type d'agents appelé *agents output* crée un binding par interaction. Ceci permet à n'importe quel agent de déclencher l'interaction à l'aide d'un simple message sur le bus. La syntaxe des messages de sortie est illustrée sur la figure 9. Elle est similaire à celle pour les entrées, seul le `IN` est remplacé par `OUT`. Notons au passage qu'il est donc possible d'une part pour un périphérique d'être en même temps un périphérique d'entrée et un périphérique de sortie, et d'autre part qu'il est possible de récupérer les messages d'un périphérique en entrée, en sortie ou les deux en ajustant l'expression régulière comme il se doit.

La figure 10 montre un exemple d'interaction pour le PHANToM. L'agent gérant les sorties sur le PHANToM a créé un binding permettant de contraindre le PHANToM sur un point dont les coordonnées sont données en argument dans le message. Le mécanisme du callback est toujours le même :

```
OUT PERIPHERIQUE : par1=val1; par2=val2; ...; parn=valn;
```

Figure 9 – Syntaxe des périphériques de sortie

on crée le binding dans le constructeur, et on récupère les paramètres dans le callback. La méthode `setForceModel` et la classe `PointForceModel` permettent de dialoguer avec l'API Reachin pour piloter le PHANToM. Ce mécanisme sera développé dans la section 2.2.3. Nous remarquons que grâce à ce système le concepteur d'applications n'a plus besoin de dialoguer directement avec l'API du périphérique, mais uniquement avec les agents output.

```
PhantomOutputAgent::PhantomOutputAgent()
: MicoleAgent("PhantomOutputAgent","Output"), ...
{
  ...
  bindMessage("^OUT PHTM : point=\\((.*), (.*), (.*)\\);$",\
              BUS_CALLBACK_OF(PhantomOutputAgent, handlePointFM));
      ...
}

void FF3DDeviceOutputAgent::handlePointFM(MicoleBus *app,
                                          int argc, const char **argv)
{
  Vec3f position;
  position.x = atof(argv[0]);
  position.y = atof(argv[1]);
  position.z = atof(argv[2]);
  ...
  setForceModel(new PointForceModel(position));
}
```

Figure 10 – Agent output pour le PHANToM

2.2.3. Les principaux agents

L'API gère plusieurs périphériques d'entrée et de sortie : un clavier, une souris, un PHANToM, une souris VTPlayer, etc. Nous allons étudier les mécanismes principaux de ces agents.

PHANToM.

Le PHANToM est un périphérique de pointage à six degrés de liberté en entrée et trois en sortie sous forme de retour de force. Nous utilisons l'API Reachin pour piloter ce périphérique. Cette API

gère une scène sous forme de graphe, appelé *scene graph* dont la structure est basée sur VRML. Les nœuds de ce graphe sont des objets, des périphériques, des conteneurs, etc. Chaque nœud peut interagir avec la scène grâce à des champs d'entrée ou de sortie. La communication entre deux nœuds se fait en reliant un champ de sortie du premier vers un champ de sortie du second. Il est possible de placer un script entre les deux afin d'interpéter l'information transitant entre les deux. Un exemple simple de script consiste à traduire la sortie d'un champ de type chaîne de caractères afin de pouvoir la relier à un champ de type entier. Le transit de l'information entre ces nœuds et l'exécution des scripts se fait par la *boucle du scene graph*. Cette boucle tourne le plus rapidement possible, selon la charge du processeur et l'encombrement de la mémoire. Ordinairement sa fréquence se situe au-dessus de $60Hz$. Or afin d'avoir un confort de sensation haptique, les forces du PHANToM doivent être calculées à $1000Hz$. L'API Reachin possède donc une deuxième boucle d'exécution en parallèle appelée *boucle temps réel* qui s'exécute à $1000Hz$. Cette boucle est utilisée pour calculer les forces à appliquer au PHANToM. Chaque objet de la scène peut définir un *modèle de force*, qui consiste principalement en une fonction qui calcule un vecteur force, généralement en fonction de la position du PHANToM. Chaque modèle de force est associé à un objet *collider* qui indique si le PHANToM est soumis à ce modèle de force. Afin de calculer les forces, l'API Reachin effectue une somme pondérée des vecteurs force produits par les modèles de force activés.

Les agents input du PHANToM n'ont rien de particulier. L'agent sender envoie régulièrement les coordonnées du PHANToM ainsi que l'état de ses boutons sur le bus. Les agents input stockent les coordonnées et l'état des boutons et fournit des méthodes événementielles pour les évènements bouton et les déplacements.

L'agent output doit accéder à la boucle temps réel afin de calculer les forces. Pour cela il crée des modèles de force. Deux modèles de force sont utilisés : le premier sert à contraindre le PHANToM, et le second sert à le guider sur une trajectoire. Ainsi il est possible de guider tout en contraignant sur la trajectoire. En dehors des modèles de force il n'est pas nécessaire d'accéder à la boucle haptique. En effet même si nous avons besoin de calculer les forces à $1000Hz$, il n'est pas nécessaire d'être aussi réactif pour changer le modèle de force, c'est-à-dire la formule de calcul du vecteur force à appliquer. Le changement du modèle de force peut donc se faire par un message sur le bus, et qui est traité par la boucle du *scene graph*. Les modèles de force implémentés permettent de contraindre un utilisateur sur un point, sur une ligne, sur un plan, sur un disque ou dans une sphère. Les deux messages de la figure 11 permettent respectivement de contraindre le PHANToM sur le point de coordonnées $(1, 2, 1)$, et supprimer la contrainte. Ces instructions permettent au concepteur d'application de changer le modèle de force utilisé pour la contrainte.

```
OUT PHANToM : point=(1, 2, 1);
OUT PHANToM : none;
```

Figure 11 – Messages pour la sortie sur le PHANToM

Deux autres modèles de force particuliers de contrainte ont été programmés. Ils permettent de gérer les interactions PICOB présentées dans le chapitre III. Le modèle de force gérant les PICOB semi-guidés est basé sur le modèle de la contrainte sur la ligne, et celui gérant les PICOB guidés est basé sur le modèle du point. La déviation en fonction de la position ou du temps est matérialisée par un déplacement du point d'attraction. Les messages ordonnant la création des séquences de PICOB semi-guidés et guidés sont visibles sur la figure 12. Dans le cas des PICOB semi-guidés, le premier paramètre indique les extrémités du segment sur lequel se trouve la séquence d'impulsions. Dans le cas des PICOB guidés, ce paramètre indique la position de départ de la séquence d'impulsions. Les autres paramètres sont communs : `bumps` est une liste de vecteurs donnant la direction et l'amplitude des impulsions, `bumplength` est la longueur d'une impulsion (spatiale pour les PICOB semi-guidés et temporelle pour les PICOB guidés) et `pauselength` est la longueur de l'espace entre deux impulsions.

```
OUT PHANTOM : picobhg=((0, 0, 0),(1, 0, 0));
bumps=((0, 1, 0), (0, 0, -1)); bumplength=0.5; pauselength=0.1;

OUT PHANTOM : picobg=(0, 1, 0); bumps=((-1, 0, 0), (0, 0, 1));
bumplength=0.4; pauselength=0.2;
```

Figure 12 – Messages pour la création de PICOB

La technique de guidage que nous avons implémentée permet de déplacer le PHANToM sur une trajectoire. Elle consiste à guider le PHANToM vers un point d'attraction que l'on déplace à volonté. Ceci peut être utilisé par exemple par un enseignant pour guider un élève à travers la scène. Cette interaction utilise un modèle de force dérivé de l'attirance vers un point. Ce modèle de force est utilisé par le deuxième modèle de force géré par l'agent output du PHANToM. La figure 13 montre un exemple de message utilisé pour changer le point d'attraction pour le guidage.

```
OUT PHANTOM GUIDE : point=(0, 1, 2);
```

Figure 13 – Message pour le point de guidage

L'agent de guidage permet d'avoir un contrôle de plus haut niveau sur le guidage. Il stocke une liste de points ajoutés soit directement dans le programme, soit par l'intermédiaire du bus. Cette liste joue un rôle de tampon. Lorsque le PHANToM approche suffisamment du point courant, le point suivant est déclenché. Plus la contrainte sur le PHANToM est forte et plus la liste est courte car l'utilisateur a plus de mal à résister. Pour stocker moins de points il est aussi possible de changer la distance de déclenchement du point suivant. Cependant si cette distance est trop grande, les courbes peuvent devenir polygonalisées car l'attraction se fait entre deux points, c'est-à-dire de manière rectiligne. La gestion du point d'attraction utilise une système conçu par Crossan *et al.* [46] qui prend en

compte la vitesse et l'accélération du PHANToM afin de créer une force continue. La fonction de calcul de la force est linéaire par rapport à la distance entre la cible et le PHANToM, comme suggéré par Magnusson *et al.* [103]. Les commandes pour réaliser le guidage à l'aide de cet agent sont reportées dans la figure 14. La commande `PLAY` permet d'initier la guidage sur la liste de points enregistrés, la commande `REC` permet de lancer l'enregistrement des points, les commandes `PAUSE` et `STOP` permettent de contrôler la lecture et l'enregistrement, et la commande `ADD` permet d'ajouter un point, y compris pendant la lecture.

```
OUT PHANTOM GUIDE PLAY;
OUT PHANTOM GUIDE STOP;
OUT PHANTOM GUIDE PAUSE;
OUT PHANTOM GUIDE REC;
OUT PHANTOM GUIDE ADD : pos=(0, 1, 2);
```

Figure 14 – Messages pour la gestion de la liste de points de guidage

VTPlayer.

La souris VTPlayer est intéressante en tant qu'afficheur braille : elle possède deux matrices de 4×4 picots. Il existe plusieurs manières de la piloter. La première consiste à utiliser le pilote officiel, grâce à une dll. Cette méthode n'est pas portable et nécessite d'installer le driver officiel qui n'est pas stable. Un pilote pour Linux a été développé par Jacquet [77]. Ce pilote possède un module pour le noyau du système permettant à la VTPlayer de se comporter comme un périphérique de pointage du système. Or encore une fois ce driver n'est pas portable et nous n'avons pas besoin de l'utiliser comme un périphérique de pointage système. Nous avons donc développé notre propre driver, en utilisant la LibUSB [98]. Cette librairie permet de communiquer avec un port USB, c'est-à-dire envoyer et recevoir des octets. En réception nous recevons les coordonnées relatives et les états des boutons. En sortie nous envoyons quatre octets, qui codent la configuration des deux matrices. Une API a été développée afin de prendre en charge la VTPlayer grâce à des fonctions haut niveau [106]. Avec cette API il est possible d'afficher des Tactons statiques et dynamiques tels que définis dans le chapitre II, ce qui facilite leur intégration à l'architecture. Tout comme pour le PHANToM, nous avons trois agents définis : un agent sender, un agent input et un agent output.

L'agent sender envoie trois types de message : un pour envoyer les coordonnées, un pour la pression d'un bouton et un autre pour le relâchement d'un bouton (figure 15). L'agent input correspondant stocke à la fois la position du curseur et l'état des boutons. De plus il fournit trois méthodes événementielles : `onMove`, `onButtonDown` et `onButtonUp`. Ces méthodes permettent de réagir aux évènements provenant de la VTPlayer sans se soucier des messages transitant sur le bus. Toutefois il est toujours possible d'implémenter un agent input du début pour avoir un contrôle plus fin et ne traiter que les évènements souhaités.

```
IN VTP : pos=(28, 42);
IN VTP : press=(2);
IN VTP : release=(1);
```

Figure 15 – Messages des entrées de la VTPlayer

La partie qui nous intéresse le plus concerne la sortie. Elle s'effectue grâce à un agent output. Cet agent offre trois services : afficher un Tacton statique, afficher un Tacton dynamique, et enfin afficher un Tacton directionnel statique ou dynamique. La figure 16 montre un exemple de chaque service. Le premier exemple permet d'afficher une ligne verticale de 4 picots sur la gauche de la matrice. La chaîne passée en paramètre est une linéarisation de la matrice, les picots levés étant notés `1` et les picots baissés `0`. Le deuxième exemple illustre la création d'un Tacton dynamique dont l'animation est une ligne horizontale de 4 picots en haut de la matrice clignote, et dont l'unité temporelle est de $200ms$. Ces deux services permettent de créer des Tactons personnalisés. Par contre le troisième service permet d'utiliser les résultats des expérimentations du chapitre II. L'exemple donné permet d'afficher un Tacton directionnel statique indiquant le Nord-est sur la matrice de droite. Il est possible d'afficher ainsi 8 directions, avec des Tactons statiques et dynamiques.

```
OUT VTP : left=(1000100010001000);
OUT VTP : right=(1111000000000000,0000000000000000); dur=200;
OUT VTP : dir=NE; type=static; cell=right;
```

Figure 16 – Messages des sorties de la VTPlayer

Clavier et souris.

L'architecture MICOLE permet de gérer des périphériques courants comme le clavier ou la souris. Ces périphériques ne fournissent que des services d'entrée, donc ils ne possèdent pas d'agent output. Les évènements sont récupérés par l'intermédiaire de l'API Reachin. Si nous décidons de changer d'API, seuls les agents ci-dessous devront être modifiés. Cela permet de simplifier la tâche au développeur d'applications.

Le clavier est celui dont le fonctionnement est le plus simple. Lorsqu'une touche est pressée, l'agent sender du clavier envoie son code ASCII sur le bus à travers un message tel que le premier de la figure 17. L'agent input écoute les messages correspondant à ce motif afin de déclencher une méthode événementielle `onKey`. Des améliorations peuvent être apportées afin de gérer les combinaisons de touches et les évènements `onKeyUp` et `onKeyDown`.

Le contrôle de la souris est similaire, mais possède quelques particularités. Le sender envoie deux types de message, dont un exemple de chaque est écrit sur la figure 17. Le premier n'envoit que les coordonnées de la souris, et le deuxième envoie les coordonnées et un évènement bouton.

L'évènement bouton est caractérisé par le numéro du bouton et le type d'évènement : `PRESSED` et `RELEASED`. L'agent input est particulier dans le sens où deux agents génériques ont été créés. Le premier sert à offrir une interface commune à tous les types de souris : souris à deux boutons, trois boutons, VTPlayer, trackball, etc. Elle fournit trois méthodes événementielles : `onMove`, `onButtonUp` et `onButtonDown`. Ainsi il est possible d'utiliser une souris dans une application en choisissant celle qu'on veut utiliser parmi tous les modèles possibles implémentés. L'agent input de la VTPlayer expliqué dans le paragraphe précédent est construit à partir de cet agent générique. L'agent input de la souris « conventionnelle » dont nous parlons ici lit les messages envoyés par le sender décrit ci-dessus et stocke la position de la souris, ainsi que l'état de chaque bouton.

```
IN KBD : key=42;
IN MSE : pos=(27, 36);
IN MSE : pos=(33, 15); evt=(BT1_PRESSED);
```

Figure 17 – Messages des entrées clavier et souris

Audio.

L'architecture fournit des agents qui agissent sur le canal auditif en sortie, que ce soit par des sons ou par la parole. L'agent output pour les sons utilise les messages tels que ceux de la figure 18. La première syntaxe permet de lire un fichier, le mettre en pause, ou l'arrêter selon le paramètre `type`. Les autres paramètres de lecture sont le volume, le haut parleur, et la position dans l'espace sonore. Les autres commandes permettent de suspendre, de reprendre et d'arrêter la lecture de tous les fichiers en cours de lecture. La librairie audio utilisée est la librairie BASS [13], qui permet de gérer le son spatial.

```
OUT SND : name=plouf.wav; type=1; vol=100; spk=FRONT; pan=0;
OUT SND : PAUSE_ALL;
OUT SND : RESUME_ALL);
OUT SND : STOP_ALL;
```

Figure 18 – Messages pour la lecture de sons

L'utilisation de la synthèse vocale se fait aussi simplement à l'aide de messages sur le bus. Deux méthodes de lecture sont possibles. La première consiste à lire une chaîne donnée en paramètre, comme avec le premier message de la figure 19. La deuxième méthode consiste à lire le contenu d'un fichier texte, à l'instar du second message. Enfin il est possible d'arrêter la lecture. Le système utilise la Microsoft Speech API (ou MS SAPI) [114] comme API de synthèse vocale.

L'architecture MICOLE est suffisamment souple pour que l'ajout d'un périphérique ou d'une modalité pas gérée se fasse facilement et rapidement. Les périphériques d'entrée ont besoin de leur

```
OUT SPCH : str="l'haptique c'est fantastique";
OUT SPCH : file=discours.txt;
OUT SPCH : STOP;
```

Figure 19 – Messages pour la synthèse vocale

agent sender qui communique avec l'API qui gère ce périphérique, et l'agent input générique qui sera précisé dans l'application elle-même. Les périphériques de sortie ne nécessitent eux qu'un agent output, qui tient le même rôle que l'agent sender, à savoir d'intermédiaire entre notre architecture et l'API du périphérique.

Nous allons à présent utiliser cette architecture dans un cas concret. La section suivante décrit la conception et l'évaluation d'une application multimodale permettant d'explorer des schémas de circuits électriques. Après avoir présenté deux prototypes, nous étudierons une application finale qui utilise l'architecture MICOLE.

3. Logiciel d'exploration de schémas de circuits électriques

Notre équipe a mené des enquêtes préliminaires auprès des enfants déficients visuels et leurs enseignants au début du projet MICOLE [113]. Elles nous ont révélé que les besoins en matière d'écucation se situent principalement au niveau des schémas et des notations mathématiques. Le deuxième point a été étudié dans les universités de Linz et UPMC à Paris dans le cadre du projet MICOLE pour donner le système MAWEN [8]. Ce système utilise une tablette braille pour représenter les notations mathématiques de manière linéaire. En ce qui concerne les schémas, plusieurs voies ont été suivies. Les partenaires de l'université de Lund se sont concentrés sur un système de dessin multimodal : AHEAD [105]. Le dessin se fait avec un PHANToM, sur une surface malléable. Il est possible d'activer un retour sonore permettant de se repérer sur la surface de dessin. Les figures dessinées peuvent être nommées et une synthèse vocale permet de prononcer le nom de la figure explorée. Il est aussi possible de guider le PHANToM vers une position donnée. Les partenaires de l'université d'Uppsala ont étudié l'exploration de cartes géographiques à l'aide d'une souris VTPlayer et d'un retour sonore [84]. L'équipe de Tampere a conçu un système d'exploration du système solaire avec un PHANToM et une souris Magellan [153]. Ce système permet d'explorer les orbites des planètes grâce à des rainures elliptiques sur un plan. Il est aussi possible d'explorer la surface des planètes pendant que la synthèse vocale donne des informations sur la planète explorée. L'équipe de Glasgow a examiné l'exploration d'un labyrinthe par des utilisateurs avec un PHANToM, avec des Tactons directionnels pour indiquer la direction à suivre [45].

Notre équipe s'est intéressée à l'exploration de schémas de circuits électriques. La figure 20 présente un exemple de schéma que nous souhaitons représenter. Le but est d'utiliser la multimodalité afin d'aider un utilisateur déficient visuel à parcourir un schéma de circuit électrique. Nous sommes spécialisés dans les interactions haptiques donc nous avons choisi d'utiliser principalement

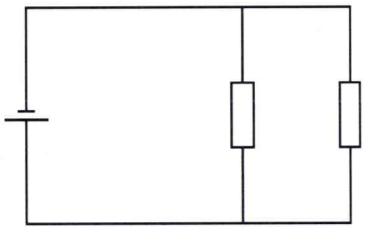


Figure 20 – Exemple de schéma de circuit à représenter

ce canal pour guider les utilisateurs. Les deux notions principales que nous souhaitons mettre en avant sont : la topologie du circuit et les composants électriques. Le PHANToM a été choisi comme périphérique d'exploration. Son retour de force est potentiellement une aide appréciable pour explorer le circuit et coder les composants. La conception de cette application s'est faite de manière incrémentale en parallèle de la conception et l'évaluation des techniques d'interaction présentées dans les chapitres II et III, et de l'architecture détaillée ci-dessus. L'application finale utilise toutes les techniques d'interaction que nous avons conçues, et repose sur l'architecture MICOLE que nous avons développée.

3.1. Premier prototype

La conception de cette application a commencé par la création de la représentation interne d'un circuit. Ensuite nous avons développé l'aspect visuel et enfin les interactions haptiques. L'affichage se fait en utilisant les objets graphiques de l'API Reachin qui utilise OpenGL. Le retour de force se fait aussi grâce à l'API Reachin, pour piloter un PHANToM Desktop.

3.1.1. Conception

Les circuits sont constitués de deux éléments principaux : les nœuds et les fils. Les fils peuvent être remplacés par des composants. Nous ne considérerons que des composants dipôles, c'est-à-dire des composants possédant deux pôles. Les nœuds ont deux fonctions : la première est de relier les fils et les composants, et la deuxième est de marquer les changements de direction.

L'API Reachin utilise un *scene graph* comme représentation interne. Ce graphe utilise une version étendue de VRML. Des nœuds gérant le rendu haptique, les entrées de périphériques tels que le PHANToM, etc. ont été ajoutés. Nous appellerons ces nœuds des nœuds Reachin pour ne pas confondre avec les nœuds des circuits électriques. Il est possible de créer de ses propres nœuds Reachin afin de pouvoir les réutiliser dans une scène, et en particulier dans un fichier VRML décrivant la scène. La partie du graphe correspondant à la scène peut être écrite dans un fichier VRML afin d'être chargée dynamiquement. Il est donc possible de changer une scène sans recompiler le programme. Nous avons créé des nœuds Reachin pour nos fils, composants et nœuds. Pour créer un circuit il suffit donc d'écrire une partie de fichier VRML utilisant ces nœuds Reachin, comme sur la figure 21. Le fichier commence par la définition des nœuds, avec un champ `position`. Ensuite vient la liste des fils et composants, avec deux champs : un pour indiquer le nœud de départ et l'autre pour indiquer

```
DEF ptA Node { position -0.06  0.06  0 }
DEF ptB Node { position  0.06  0.06  0 }
DEF ptC Node { position  0.06 -0.06  0 }
DEF ptD Node { position -0.06 -0.06  0 }
Battery {
  begin "ptA"
  end "ptB"
}
Resistor {
  begin "ptB"
  end "ptC"
}
Lamp {
  begin "ptC"
  end "ptD"
}
Line {
  begin "ptD"
  end "ptA"
}
```

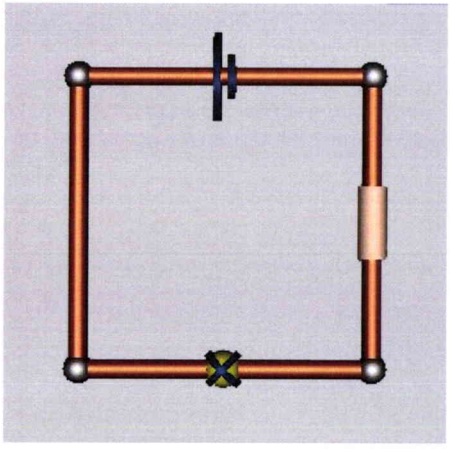


Figure 21 – Source d'un circuit et sa représentation visuelle

le nœud d'arrivée. L'exemple de la figure 21 est un circuit de forme carrée, avec une pile en haut, une résistance à droite, une lampe en bas, et un fil à gauche.

Visuel.

Notre application est destinée aux utilisateurs déficients visuels, donc *a priori* l'aspect visuel n'est pas important. Il y a cependant plusieurs raisons qui rendent l'aspect visuel indispensable. La première est pratique : il est très utile lors de la phase de développement d'avoir un rendu visuel correspondant au rendu haptique afin de contrôler le comportement de l'application. La deuxième raison est que les utilisateurs déficients visuels ne sont pas tous aveugles : certains ont une vision résiduelle et s'en servent. La troisième raison est que notre application doit pouvoir être utilisée en collaboration entre utilisateurs déficients visuels et utilisateurs voyants. En particulier, de manière générale les enfants déficients visuels n'aiment pas avoir de traitement particulier. Il est donc plus rassurant pour eux d'utiliser des outils pouvant être utilisés par tout le monde.

Les deux objets principaux des circuits sont les nœuds et les fils. Leurs représentations visuelles sont présentées sur la figure 22. Les nœuds permettent de relier des fils, et ce dans toutes les directions. Nous avons donc décidé de les représenter sous forme de sphères. Les fils sont eux représentés par des cylindres. Les fils sont toujours rectilignes afin d'éviter les problèmes d'orientation. Pour les

changements de directions, deux fils sont utilisés, et connectés par un nœud. Les nœuds permettent donc d'avertir d'un changement de direction. Un code de couleur est utilisé pour donner des informations relatives à l'exploration des fils. Un fil est :

- Lumineux si l'utilisateur parcourt le fil actuellement.
- Sombre si l'utilisateur a déjà parcouru le fil.
- Normal si l'utilisateur n'a pas encore parcouru le fil.



Figure 22 – Représentation visuelle des fils et des nœuds

Les objets qui restent à représenter sont les composants. Les schémas de circuits électriques que nous souhaitons afficher sont destinés à des enfants d'école primaire et de collège. Très peu de composants sont étudiés à ce niveau. Nous n'avons donc implémenté que quatre composants : la lampe, la résistance, la pile et le condensateur. Cependant l'application a été conçue de manière à ce qu'il soit facile d'ajouter des composants. Les visuels des composants implémentés sont représentés sur la figure 23. L'aspect visuel a été conçu pour ressembler le plus possible aux symboles utilisés sur les schémas imprimés : la lampe est représentée par une sphère barrée par une croix, la résistance est représentée par un cylindre vu du côté, la pile est représentée par une grande barre et une petite barre, et le condensateur est représenté par deux barres de même longueur.

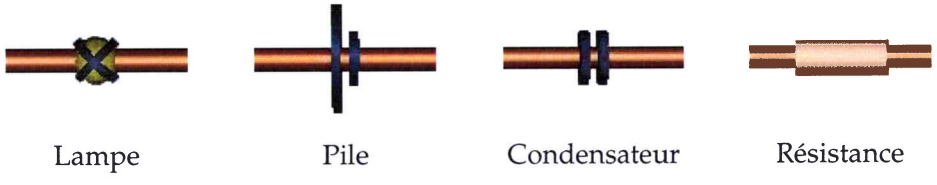


Figure 23 – Représentation visuelle des composants

Haptique.

Pour concevoir le rendu haptique de ces schémas nous avons suivi les recommandations de Sjöström [158] : nous avons aimanté le circuit afin que les utilisateurs soient attirés dessus lorsqu'ils passent à proximité. Ainsi ils peuvent explorer la forme du circuit en se laissant guider par l'aimantation. Pour aimanter le circuit nous avons utilisé un point magnétique, placé sur la projection la plus proche du PHANToM sur le circuit. La force est dirigée du PHANToM vers le point le plus proche sur le circuit, ainsi que le montre la figure 24. Cette aimantation permet de ressentir le circuit et d'en explorer la forme. Le circuit est placé sur le plan vertical, afin que le mouvement du stylet se fasse relativement à l'écran.

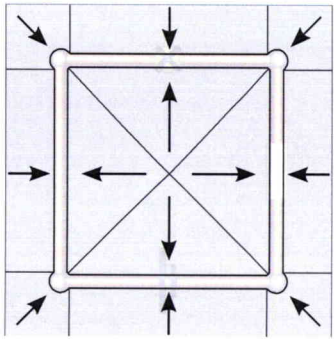


Figure 24 – Aimantation du schéma

La force appliquée est celle d'un aimant, c'est-à-dire qu'elle est inversement proportionnelle au carré de la distance entre le point sur le circuit et le PHANToM. Autrement dit plus le PHANToM est proche du schéma, plus il est attiré. La valeur de cette force est donnée par la formule ci-dessous, où c est une constante et d est la distance entre le point sur le circuit et le PHANToM.

$$||\vec{F}|| = \frac{c}{d^2}$$

Les composants sont codés avec des séquences PICOB semi-guidées (chapitre III, sections 4 et 5). La lampe est représentée par une impulsion, la pile par une petite et une grande impulsion, le condensateur par deux impulsions et la résistance par trois impulsions (figure 25). Les impulsions utilisent le même plan que le circuit, c'est-à-dire le plan vertical. Il existe donc plusieurs directions possibles pour ces impulsions. La direction utilisée est vers l'extérieur si le sens est le même que celui des aiguilles d'une montre, et vers l'intérieur dans le cas contraire.



Figure 25 – Représentation haptique des composants

La figure 26 montre l'aimantation du circuit en prenant en compte les PICOB semi-guidés représentant les composants. Les PICOB sont des déformations des fils sur lesquels il y a un composant.

3.1.2. Évaluation informelle

L'évaluation de ce premier prototype s'est limitée à des utilisations informelles du système. Elle a été réalisée lors de démonstrations et de réunions de travail. Le système a été utilisé tant par des personnes habituées à utiliser le PHANToM que par des utilisateurs ne l'ayant jamais manipulé. Tous sont voyants. Certaines remarques sont unanimes. Premièrement l'attraction n'est pas suffisante : les utilisateurs sortent trop souvent et trop facilement du circuit. Ceci complique la tâche de l'exploration et il est difficile de suivre le circuit, même pour des voyants. Nous allons donc renforcer cet aspect. Le second point est qu'il n'est pas évident de connaître les directions possibles lorsqu'on arrive à un

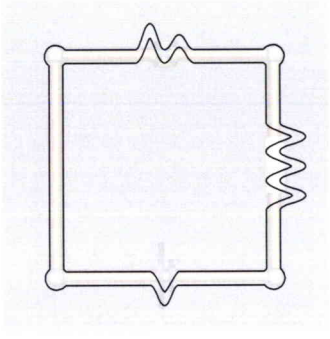


Figure 26 – Aimantation du circuit avec les composants

nœud. Beaucoup d'utilisateurs forcent et se retrouvent en dehors du circuit. Il faut donc indiquer les directions aux utilisateurs. Certains utilisateurs rapportent aussi une fatigue liée à une utilisation prolongée. La cause de cette fatigue est que la main doit rester levée pendant l'exploration.

3.2. Second prototype

Après ces tests informels, nous avons amélioré le prototype en prenant en compte les remarques émises par les utilisateurs. De plus cette fois nous avons utilisé la version OMNI du PHANToM. Il est moins précis mais moins coûteux. Si des établissement s'équipent d'un PHANToM, ce sera vraisemblablement avec cette version, d'où ce choix pour les tests futurs. Cependant l'application fonctionne avec les deux.

3.2.1. Modifications

La première modification concerne l'aimantation : nous avons décidé de projeter en permanence le PHANToM sur le circuit afin de contraindre l'utilisateur à l'intérieur du schéma. Le but est de restreindre l'espace de l'utilisateur au circuit lui-même afin d'éviter que celui-ci puisse se perdre dans une zone vide d'information. La force à appliquer a dû être changée. En effet avec une force magnétique, la force décroît en fonction de la distance. Nous utilisons désormais une force linéaire proportionnelle, telle que la force exercée par un ressort. Le vecteur force créé est donné par la formule ci-dessous, avec k une constante (constante de raideur dans le cas du ressort) et $\overrightarrow{\Delta l}$ le vecteur entre le PHANToM et sa projection sur le circuit.

$$\overrightarrow{F} = k\overrightarrow{\Delta l}$$

En deuxième lieu nous avons décidé de placer le schéma sur le plan horizontal au lieu du plan vertical. Cette décision est motivée par la fatigue des utilisateurs. En effet en travaillant sur le plan horizontal, nous pensons que l'avant-bras de l'utilisateur peut reposer sur la table et ainsi éviter la fatigue. De plus lorsque nous utilisons un crayon pour dessiner, nous nous plaçons souvent sur le plan horizontal. Le fait d'utiliser le même plan permet potentiellement à l'utilisateur de garder ses repères et d'explorer plus facilement.

Le troisième point corrige le manque d'information concernant les directions possibles. En effet nous avons ajouté des séquences de PICOB guidés dans les nœuds. Chaque séquence indique une direction adjacente au nœud. Les paramètres fixés dans le chapitre III gardent leurs valeur, à savoir la forme sinusoïdale et la durée à $250ms$. Les directions utilisées sont toutes les directions du plan horizontal. Cependant les circuits que nous utilisons sont basiques et n'utilisent que les directions de droite à gauche et d'avant en arrière. Pour l'amplitude, nous avons décidé d'utiliser deux valeurs afin de coder les directions déjà explorées (petite amplitude) et les directions non explorées (grande amplitude). La figure 27 montre un exemple de directions de séquences de PICOB guidés jouées aux intersections. La pile en haut et le fil à gauche ont déjà été parcourus alors que la résistance de droite et la lampe du bas ne l'ont pas encore été.

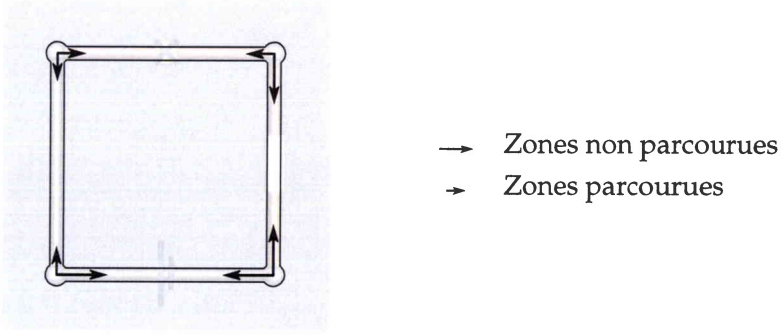


Figure 27 – Indications directionnelles avec des PICOB guidés

Ces séquences de PICOB guidés sont lues automatiquement une fois lorsque l'utilisateur entre dans le nœud. Ensuite il peut les lire à volonté en pressant le bouton du PHANToM.

3.2.2. Évaluation

L'évaluation de ce prototype [140] a été réalisée en même temps que l'évaluation des séquences de PICOB semi-guidées discutée dans la section 5 du chapitre III. Cette expérimentation se focalise sur l'exploration des circuits, la représentation des composants ayant été étudiée dans la section susmentionnée. Ces circuits ne sont donc composés que de fils et de nœuds.

Méthodologie.

Pour cette expérience nous avons créé 9 circuits (table 1) que nous avons classés en 4 catégories. La première catégorie ne contient qu'un circuit de forme carrée. Ce circuit sera utilisé comme démonstration de l'application avant les tests. La seconde catégorie contient 3 circuits de type parallèle. Deux d'entre eux sont identiques à une rotation près. L'utilisation de ces circuits permettra de détecter d'éventuels problèmes de reconnaissance de l'orientation des formes. Le troisième circuit possède une branche de plus que les deux autres. Des confusions entre ces circuits pourraient être le signe de perte de repères dans le circuit. La troisième catégorie regroupe 3 circuits basés sur un carré et un nœud central, avec 2, 3 ou 4 fils adjacents. Ces circuits sont 3 carrés contenant respectivement 1, 2 ou 4 carrés. L'intérêt de ces circuits est de forcer l'exploration et la construction

d'une représentation mentale du schéma. Enfin la quatrième catégorie est constituée de 2 circuits identiques à une symétrie près. Ces circuits représentent un carré accroché sur un des côtés d'un rectangle. L'idée est que ces schémas possèdent une structure imbriquée, ce qui risque de perturber les utilisateurs car ils ne peuvent pas s'aider de leur vision dans cette expérience.

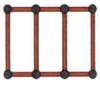
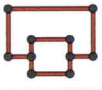
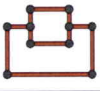
Catégorie 1	Catégorie 2		
1	2	3	4
			
Catégorie 3		Catégorie 4	
5	6	7	8	9
				

Table 1 – Schémas utilisés dans l'expérience

15 utilisateurs déficients visuels, dont 3 aveugles ont participé aux tests (table 2). Ils étaient âgés de 9 à 23 ans (moyenne : 15,4 ans, écart type : 4,3 ans), 5 filles et 10 garçons. Les utilisateurs ayant une vision partielle avaient les yeux bandés afin de ne mesurer que l'apport des interactions haptiques. Tous ont eu pour instruction de manipuler le PHANToM avec leur main dominante.

Avant les tests, les utilisateurs devaient se familiariser avec le périphérique avec quelques logiciels de démonstration fournis avec l'API Reachin. Ensuite ils devaient parcourir le circuit 1 afin de se familiariser avec l'application. L'expérimentateur expliquait le fonctionnement de l'application à l'utilisateur, en particulier les PICOB guidés aux intersections.

Le test consistait à reconnaître 3 circuits. Les utilisateurs pouvaient explorer les circuits aussi longtemps qu'ils le souhaitaient. Ils ont eu pour consigne de faire attention à l'orientation et au nombre d'intersections à chaque nœud. Il leur a été aussi demandé d'explorer les circuits en entier, et de ne pas hésiter à cliquer sur le bouton du PHANToM pour demander à rejouer les séquences de PICOB. Une fois qu'ils ont suffisamment parcouru le circuit, l'application se terminait et ils devaient reconnaître le circuit exploré entre 6 propositions sélectionnées parmi les 9 circuits. Les utilisateurs aveugles exploraient les propositions sur du papier en relief, et les utilisateurs ayant une vision résiduelle utilisaient des schémas imprimés agrandis à fort contraste.

Les 3 schémas à reconnaître ont été choisis dans les trois dernières catégories de circuits. Les 6 propositions montrées aux utilisateurs sont résumées dans la table 3. Le choix des propositions a été fait *a priori* par les expérimentateurs en choisissant les circuits appartenant à la même catégorie que celui à reconnaître, ainsi que des circuits similaires appartenant aux autres catégories. Nous n'avons pas présenté tous les circuits pour gagner du temps car nous ne disposions que de peu de temps par utilisateur. Nous avons donc écarté les solutions que nous avons jugées trop différentes.

Utilisateur	Sexe	Âge	Vision
1	Masculin	23 ans	Vision résiduelle
2	Masculin	22 ans	Vision résiduelle
3	Féminin	21 ans	Aveugle
4	Masculin	20 ans	Aveugle
5	Masculin	18 ans	Aveugle
6	Féminin	15 ans	Vision résiduelle
7	Féminin	15 ans	Vision résiduelle
8	Masculin	14 ans	Vision résiduelle
9	Masculin	14 ans	Vision résiduelle
10	Masculin	13 ans	Vision résiduelle
11	Masculin	12 ans	Vision résiduelle
12	Féminin	12 ans	Vision résiduelle
13	Masculin	12 ans	Vision résiduelle
14	Féminin	11 ans	Vision résiduelle
15	Masculin	9 ans	Vision résiduelle

Table 2 – Utilisateurs

Circuit	Propositions					
3	1	2	3	4	5	7
5	2	4	5	6	7	9
9	4	5	6	7	8	9

Table 3 – Les circuits à reconnaître et les propositions

Le temps d'exploration, ainsi que le parcours des utilisateurs (fils et nœuds traversés) ont été enregistrés afin d'identifier les erreurs de conception persistantes dans ce prototype, et de vérifier les hypothèses suivantes :

- Les utilisateurs parcourront tous les circuits en entier. La motivation de cette hypothèse est que nous contraignons l'utilisateur dans le schéma, et nous lui donnons des indications aux intersections. Nous pensons donc que l'utilisateur a assez d'informations pour explorer les schémas en entier.
- Les utilisateurs reconnaîtront efficacement les schémas. Cette hypothèse vient en corollaire à la précédente En effet si les circuits sont parcourus en entier, on est en droit de penser que les

utilisateurs les ont reconnus. De plus ils ont le droit de les explorer autant de temps qu'ils le désirent.

- Le temps d'exploration du troisième circuit sera plus important que celui des autres circuits. La raison de cette hypothèse est que le troisième circuit semble plus compliqué que les autres : il possède plus de fils et de nœuds que les autres, et il est constitué de deux figures imbriquées.

À l'instar de l'expérience sur les séquences de PICOB dans la section 5 du chapitre III, les utilisateurs sont très variés. Par exemple les utilisateurs de 9 ou 10 ans ne sont pas aussi chevronnés en géométrie que peuvent l'être les utilisateurs de plus de 20 ans. Il n'ont pas non plus forcément la même représentation mentale de l'espace. Les utilisateurs ont aussi tous des handicaps différents. C'est pour cela que la majeure partie de l'analyse ci-dessous sera plus descriptive que statistique. Nous n'avons pas manqué de recueillir les remarques des utilisateurs par un questionnaire (voir annexe A) en fin de session afin de recueillir des informations précieuses sur le confort d'utilisation du logiciel.

Résultats.

Les réponses des utilisateurs sont reportées dans la table 4. Les réponses fausses sont illustrées par le circuit correspondant à la réponse à des fins de comparaisons. Pour rappel, les utilisateurs aveugles sont les utilisateurs 3, 4 et 5. Le nombre de mauvaises réponses, ainsi que les temps médian d'exploration sont écrits dans la table 5. Les chiffres donnés sont ceux des utilisateurs aveugles, les utilisateurs à vision partielle, et l'ensemble.

Correct	Utilisateur														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	5	3	2	2	2	3	3	1	3	3	3	3	7	3	3
5	2	2	6	6	7	5	5	5	5	5	5	5	6	7	7
9	4	4	9	9	6	9	9	9	9	9	9	9	9	8	8

Table 4 – Réponses des utilisateurs

La faible quantité de données sur les erreurs ne permet pas de faire d'analyse statistique solide afin de déterminer si les utilisateurs aveugles ont fait significativement plus d'erreurs que les autres utilisateurs. D'après les résultats que nous avons obtenus, nous pouvons cependant remarquer que les utilisateurs aveugles ont eu des difficultés. En effet un utilisateur aveugle n'a reconnu aucun circuit, et les deux autres n'ont reconnu que le dernier. Par contre la moitié des utilisateurs ayant une vision résiduelle n'ont fait aucune erreur. Un autre de ces utilisateurs n'a confondu que le premier schéma, quatre autres en n'ont reconnu qu'un, et enfin un autre utilisateur à vision résiduelle n'a reconnu aucun circuit.

Circuit	Erreurs			Temps médian (s)		
	Total	Aveugles	Autres	Total	Aveugles	Autres
3	6/15	3/3	3/12	127	225	116
5	8/15	3/3	5/12	115	186	106
9	5/15	1/3	4/12	120	178	115

Table 5 – Résultats pour chaque circuit

Du point de vue des circuits, 6 utilisateurs sur 15 se sont trompés sur le circuit 3, 8 sur 15 ont mal reconnu le circuit 5, et enfin 5 utilisateurs sur 15 n'ont pas reconnu le circuit 9. L'analyse des confusions est intéressante : pour le circuit 3, les trois utilisateurs aveugles ont donné le circuit 2 en réponse. Or le circuit 2 est identique au circuit 3 à une rotation près. Pour les trois autres erreurs, un des utilisateur a reconnu le circuit 1, c'est-à-dire un carré. Or le circuit 1 est exactement le circuit 3 sans le fil horizontal du milieu. Un autre utilisateur a reconnu le circuit 7, qui est aussi similaire au circuit 3 avec un fil vertical au milieu. Enfin un autre utilisateur a confondu ce circuit avec le numéro 5. Les seuls points communs de ces deux circuits sont les directions sortantes de trois nœuds de gauche. Cependant l'utilisateur n'a pas exploré cette partie particulièrement plus souvent que le reste du circuit. Dans le cas du circuit 5, pour 6 des 8 confusions l'utilisateur a répondu un autre circuit de la même catégorie (circuits 6 et 7). Par contre deux utilisateurs ont répondu en faveur du circuit 2. Les constatations sont les mêmes que dans le cas précédent où un utilisateur avait répondu 5 au lieu de 3, et d'ailleurs cet utilisateur fait partie des deux ayant commis cette erreur. Deux erreurs pour le circuit 9 sont dues à des problèmes d'orientation : les utilisateurs ont répondu 8. Un autre utilisateur (aveugle) a répondu 6, alors que les deux circuits ont peu de choses en commun, si ce n'est la grande boucle sur le bas. Les deux utilisateurs ayant répondu 2 à la place de 5 ont cette fois-ci répondu 4. Il est intéressant de noter que ces deux utilisateurs sont ceux qui ont parcouru le plus les circuits avec une moyenne de 9,7 et 11 lectures par fil, alors que mis à part un utilisateur qui a une moyenne de 7,5, les autres restent entre 1,1 et 5,6 lectures. Parcourir le schéma trop souvent n'a donc fait qu'apporter de la confusion.

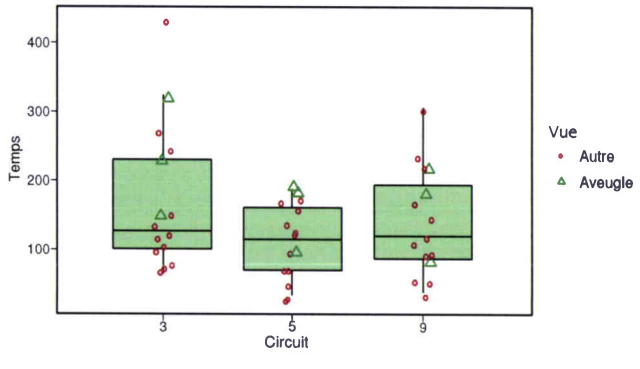


Figure 28 – Temps d'exploration par circuit

Les données des temps médians par circuit de la table 5 sont complétés par la figure 28. Les utilisateurs ont exploré le circuit 3 avec une médiane de $127s$, contre $115s$ pour le circuit 5 et $120s$ pour le circuit 9. Les temps d'explorations de chaque circuit ont pu être comparés avec un tests statistique non paramétrique de Kruskall-Wallis. Il n'y a pas de différences significatives entre les temps d'exploration des trois circuits ($\chi^2 = 2{,}19$, $p = 0{,}33$).

Les utilisateurs aveugles ont effectué une médiane de $186s$ sur l'ensemble des circuits, et les utilisateurs à vision résiduelle $115s$. Nous avons comparé les résultats de ces deux groupes d'utilisateurs à l'aide d'un test de Mann Whitney, et la différence est significative ($W = 89{,}5$, $p = 0{,}04$). Sur le circuit 3, les utilisateurs aveugles ont mis plus de temps que les trois quarts des utilisateurs à vision résiduelle. Sur le circuit 5, deux des utilisateurs aveugles sont ceux qui ont utilisé le plus de temps et le troisième se situe entre les deux moitiés des utilisateurs à vision résiduelle. Enfin sur le circuit 9, deux des utilisateurs aveugles ont mis plus de temps que les trois quarts des utilisateurs à vision résiduelle, alors que le troisième a été plus rapide que les trois quarts des utilisateurs aveugles.

Discussion.

Les résultats montrent que la tâche confiée aux utilisateur s'est avérée plus compliquée que prévu. La navigation a posé plusieurs problèmes aux utilisateurs. Premièrement nous attirons le PHANToM vers le point le plus proche du schéma. Or en forçant il est possible de se téléporter dans le circuit comme le montre la figure 29. Si le PHANToM passe la moitié de la distance entre deux fils, il est soudainement attiré vers l'autre fil. Ce problème est très gênant pour la navigation non visuelle car l'utilisateur perd tous ses repères. Dans les versions ultérieures de l'application nous veillerons à ce que l'utilisateur ne puisse se déplacer que vers des objets adjacents.

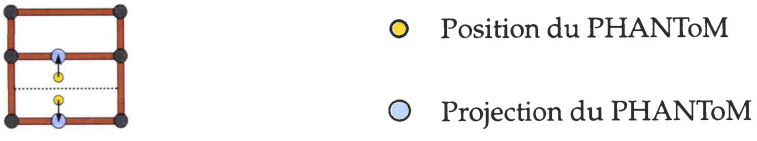


Figure 29 – Problème de téléportation dans un circuit

Le second problème majeur est reporté par les utilisateurs aveugles. Ils ont l'habitude d'explorer leur environnement par tâtonnements, et donc les séquences de PICOB guidés automatiques que nous avons placées dans les nœuds les a perturbés car ils n'étaient pas maîtres de leur exploration. Dans les versions ultérieures du logiciel nous avons désactivé la lecture automatique de ces informations, en laissant cependant la possibilité à l'utilisateur de les lire manuellement lorsqu'il le souhaite.

Ces problèmes d'exploration ont perturbé les utilisateurs et ne leur ont pas permis d'explorer tous les circuits en détails. L'utilisateur qui a répondu 1 pour le circuit 3 a exploré deux fois la boucle du dessus, mais jamais la boucle du dessous. Il pensait donc avoir terminé l'exploration. Celui qui a répondu 7 à la place de 3 a probablement perdu ses repères, ce qui explique pourquoi il a répondu un circuit ayant plus de fils et de nœuds. Les erreurs sur le circuit 9 s'expliquent par le fait que certains utilisateurs on très peu parcouru la boucle du dessus en comparaison du reste du circuit, comme

le montre la figure 30. Ces constatations nous forcent à rejeter la première hypothèse selon quoi les utilisateurs ont parcouru les circuits électriques en entier.



Figure 30 – Exploration typique du circuit 9

L'analyse des taux d'erreurs confirme que les utilisateurs ont eu des difficultés à réaliser les tâches à réaliser. 19 réponses données sur 45 sont erronées, soit 42% d'erreurs. 7 des 9 réponses des utilisateurs aveugles sont erronées. De plus selon l'aveu de ces utilisateurs, ils ont reconnu le circuit 9 car il était plus compliqué que les autres, donc ils ont pu répondre par élimination. Par contre parmi les utilisateurs à vision partielle, la moitié n'a fait aucune erreur, ce qui est un résultat encourageant. Cela peut suggérer que les problèmes reconnaissances sont en majorité dûs aux problèmes d'exploration mentionnés ci-dessus. Les taux d'erreurs qui résultent de cette expérience nous conduisent à rejeter la seconde hypothèse qui prévoyait que les utilisateurs réussiraient à parcourir les schémas en faisant peu d'erreurs.

Les utilisateurs ayant une vision résiduelle n'ont pas pu se servir de leurs vue pour réaliser la tâche car nous avons voulu savoir si le retour haptique était suffisant. Cependant en conditions réelles ils pourront se servir de leur vue, et ainsi nous pensons que l'exploration ne sera pas un problème pour eux.

Le temps médian d'exploration des trois circuits est respectivement de $127s$, $115s$ et $120s$ pour les circuits 3, 5 et 9. De manière générale les utilisateurs aveugles ont pris significativement plus de temps pour explorer les schémas que les utilisateurs à vision partielle. Cependant la quantité de données récoltées ne permet pas de faire de statistiques robustes entre catégories d'utilisateurs et par circuit. Cependant la médiane des trois utilisateurs aveugles est toujours largement supérieure à celle des autres utilisateurs : $225s$, $186s$ et $178s$ contre $116s$, $106s$ et $115s$ pour les circuits 3, 5 et 9. Sans pouvoir le confirmer par des statistiques solides, nous pouvons soupçonner que l'exploration est plus longue pour un utilisateur aveugle que pour un utilisateur ayant une vision résiduelle. Par contre ces utilisateurs n'ont pas beaucoup plus parcouru chaque fil en moyenne : 5 fois contre 4,5 fois pour les autres utilisateurs. Ils ont donc passé plus de temps à explorer chaque fil. Par contre l'analyse des temps d'exploration par circuit révèle qu'il n'y a pas de différences significatives de temps d'exploration entre les circuits. Nous devons donc rejeter la troisième hypothèse.

3.3. Application finale

Une nouvelle version du logiciel a été développée en prenant en compte les enseignements tirés des évaluations des deux premiers prototypes. La nouveauté est que nous avons utilisé l'architecture

multimodale décrite dans la section 2. Grâce à cette architecture nous avons pu ajouter aisément d'autres techniques d'interaction telles que les Tactons décrits dans le chapitre II.

3.3.1. Modifications

L'évaluation du second prototype nous a beaucoup appris sur l'exploration d'un environnement par des utilisateurs déficients visuels, et particulièrement par des aveugles. Nos observations, ainsi que nos discussions avec les utilisateurs nous ont poussés à faire quelques modifications sur l'application.

Dans un premier temps nous nous sommes penchés sur le problème de projection. En effet le tort du premier prototype était que l'utilisateur n'était qu'aimanté sur le circuit. Au fur et à mesure que l'utilisateur s'éloignait du circuit, l'attraction diminuait, et ainsi l'utilisateur pouvait se perdre aisément dans la scène. Nous avons donc décidé pour le deuxième prototype de projeter le PHANToM vers le circuit, avec une force proportionnelle à la distance. Cependant ceci n'est pas suffisant. En effet comme nous l'avons vu précédemment l'utilisateur pouvait être téléporté dans le circuit, car nous projetions vers l'élément le plus proche du circuit. Nous avons donc modifié la fonction de projection, qui ne projette que sur l'élément le plus proche parmi l'élément courant et ses voisins. Ainsi il n'est plus possible de sauter d'un endroit à l'autre du circuit.

Dans un second temps nous avons étudié la présentation de nos informations directionnelles. En effet dès le premier prototype nous avions placé dans chaque nœud des séquences de PICOB guidés afin d'indiquer les directions adjacentes. Ces PICOB étaient lus une première fois automatiquement, et pouvaient être lues à volonté par la suite par un clic de la part de l'utilisateur. Les tests du second prototype avec les utilisateurs aveugles ont révélé que les PICOB automatiques perturbent la navigation. En effet les aveugles ont l'habitude de parcourir leur environnement pour en construire une représentation mentale. Cette information imposée les empêche de parcourir librement avec une méthode qui leur est propre. Nous avons donc supprimé cette lecture automatique. Par contre il est toujours possible de recevoir ces informations à l'aide du clic. De plus les séquences de PICOB sont disponibles en tout point du circuit désormais.

3.3.2. Utilisation de l'architecture MICOLE

L'architecture décrite dans la section 2 permet de développer rapidement et aisément des applications multimodales et collaboratives pour enfants déficients visuels. Or notre application est destinée aux enfants déficients visuels. Nous utilisions principalement le retour de force, mais nous comptons utiliser d'autres modalités. Ce fût donc l'occasion d'utiliser cette architecture [142]. D'un point de vue conceptuel, si nous observons notre application en utilisant le modèle PAC [38], alors l'architecture MICOLE nous fournit les agents de présentation. Nous devons donc développer la partie abstraction, ainsi que les agents de contrôle.

La partie abstraction est très réduite : elle contient principalement la liste des fils, composants et nœuds du circuit. Cette partie est donc étroitement liée avec les agents PAC de contrôle s'occupant des entrées du PHANToM et de la souris. Enfin les agents PAC de contrôle pour les périphériques de sortie peuvent se diviser en deux catégories. En reprenant les propriétés CARE [42], Les premiers

servent à afficher de manière redondante les informations comme le type de composant par exemple. Nous les appellerons les *agents de rendu*. Les autres agents PAC de contrôle pour les périphériques de sortie permettent de guider l'utilisateur sur une trajectoire de manière assignée. Nous appellerons ces agents des *agents de guidage*.

Abstraction.

L'abstraction est distribuée dans plusieurs classes qui gèrent la représentation interne du circuit électrique. Un circuit est enregistré sous forme de fichiers VRML contenant la description du schéma, c'est-à-dire la liste de ses nœuds, ses fils et ses composants. Pour cela nous avons créé des nœuds Reachin pour chaque type d'objet. Ces nœuds Reachin permettent au passage de réaliser le rendu visuel du schéma. Nous avons choisi de laisser l'API Reachin dessiner les composants car cette représentation ne change pas pendant l'exécution de l'application, mis à part quelques détails que nous discuterons plus loin. La liste des nœuds, des composants et des fils est créée lors du chargement du fichier VRML. Cette liste est manipulée par les agents input qui gèrent l'exploration du circuit par le PHANToM et par la souris.

Agents input.

Les schémas peuvent être explorés de deux manières différentes en parallèle : soit avec la souris, soit avec le PHANToM. Le but est de permettre à deux utilisateurs de travailler en collaboration grâce à ce système.

L'agent input de la souris calcule la projection du curseur de la souris sur le circuit. Ensuite il envoie sur le bus cette position grâce au premier message de la figure 31.

```
EC MSE CURSOR : pos=(1, 2, 3);
EC PHTM CURSOR : pos=(1, 2, 3);
```

Figure 31 – Messages pour les curseurs de l'application de circuits électriques

L'agent input du PHANToM a un rôle similaire : il calcule la projection du PHANToM sur le circuit et l'envoie par message sur le bus comme le montre la figure 31. Par contre l'exploration avec le PHANToM est responsable du déclenchement du retour haptique du circuit. Pour cela cet agent retient le dernier objet parcouru afin d'envoyer un message lorsque le PHANToM passe par un nouvel objet. Ce message permet aux agents de rendu d'afficher les informations correspondantes sur les différents périphériques grâce aux messages de la figure 32.

Ces messages permettent de signaler que l'utilisateur qui manipule le PHANToM passe respectivement sur un fil, une résistance, une lampe, un condensateur, une pile ou un nœud. Le dernier est un cas particulier qui signale que l'utilisateur n'est sur aucun composant. Ceci peut être utile si on souhaite désactiver temporairement la représentation du circuit. Ces messages sont destinés aux agents de rendu qui vont s'occuper de la représentation de ces objets sur les différents périphériques.

```
EC LINE : seg=((0, 1, 1),(1, 1, 1));
EC RES : seg=((0, 1, 1),(1, 1, 1));
EC LAMP : seg=((0, 1, 1),(1, 1, 1));
EC CAP : seg=((0, 1, 1),(1, 1, 1));
EC BAT : seg=((0, 1, 1),(1, 1, 1));
EC NODE : pos=(0, 2, 1);
EC NOTHING;
```

Figure 32 – Messages pour chaque objet de l'application de circuits électriques

Agents de rendu.

Les messages envoyés par les agents input sont gérés par trois agents de rendu, qui eux communiquent avec les agents output. Le premier agent de rendu permet de gérer le PHANToM. Son rôle est de créer le modèle de force correspondant au composant actuellement parcouru. En effet à chaque objet correspond un modèle de force. Ces modèles de forces sont modifiés par un simple envoi de message sur le bus, à destination de l'agent output du PHANToM. Par exemple dans le cas d'un fil entre deux points, nous envoyons un message ordonnant de contraindre sur la ligne passant par les deux points en question. Lorsque l'utilisateur arrive au bout du fil il entre dans un nœud. À ce moment là nous le contraignons sur un disque centré sur la position du nœud. Dans les prototypes précédents nous utilisions une sphère, mais nous avons remarqué que les utilisateurs avaient des difficultés à sortir des sphères. Nous avons donc opté pour le disque afin de rester dans le plan. Pour les composants nous utilisons le modèle de force de PICOB semi-guidés, qui contraint sur une ligne passant par les deux extrémités et dont la séquence d'impulsions se trouve au milieu. Les séquences utilisées sont représentées sur la figure 33. Contrairement aux prototypes, nous avons choisi d'orienter les impulsions perpendiculairement au plan d'exploration afin de bien différencier les composants du circuit lui-même. Enfin cet agent gère aussi la lecture des PICOB guidés. Il lit les listes d'impulsions à envoyer, et crée le message destiné à l'agent output en utilisant les paramètres définis pour l'amplitude des impulsions.

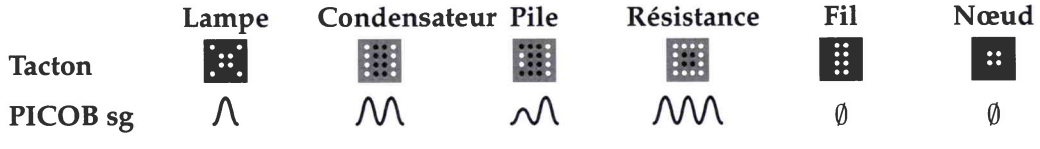


Figure 33 – Représentations haptiques des objets

Le second agent de rendu gère la souris VTPlayer. Ce rendu n'était pas disponible dans les prototypes. Cet agent lit les messages de l'agent input du PHANToM, et en réaction il crée un Tacton représentant l'objet parcouru sur la cellule se trouvant sous l'index de la main non-dominante. Le basculement en mode gaucher se fait en pressant la touche G. La réalisation de cette fonctionnalité

se fait en lisant les messages sur le bus disant que la touche G a été pressée. La figure 33 montre la représentation de chaque objet. Les lampes sont représentées par une croix, les condensateurs par deux lignes, les piles par une petite ligne et une grande ligne, les résistances par un carré, les fils par une grosse ligne et les nœuds par un petit carré. La VTPlayer affiche aussi les aides directionnelles en complément des PICOB guidés. La séquence de directions est affichée en utilisant les Tactons directionnels statiques expérimentés dans le chapitre II. Chaque direction est affichée $400ms$, en intercalant une image sans picot levés entre chaque image à motif directionnel. Ce temps est inférieur au temps médian utilisé par les utilisateurs du chapitre II pour reconnaître la direction des Tactons statiques, mais plus long que celui séparant deux images de Tactons dynamiques. Nous pensons que l'alternance de plusieurs directions permet de comparer les directions, et ainsi de les reconnaître plus facilement. De plus en pratique nous n'utilisons que 4 directions alors que dans les tests précédents nous en utilisions 8. L'évaluation nous montrera si cette valeur est adaptée ou non.

Le troisième agent de rendu gère quelques indicateurs visuels. Il reçoit à la fois la position de la souris et la position du PHANToM pour afficher les projections de ces deux périphériques sur le circuit. La projection du PHANToM est affichée en jaune, et celle de la souris est affichée en rouge. La différence majeure entre cet agent et les deux agents de rendu précédents est que celui-ci agit directement sur le scene graph, alors que les autres envoient des ordres à des agents output sur le bus.

Agents de guidage.

Dans cette version du logiciel, nous avons ajouté des services permettant à des utilisateurs de collaborer pour réaliser des tâches avec ce logiciel. Pour cela nous avons utilisé les fonctionnalités de guidage sur des trajectoires offertes par l'architecture. Deux modes de guidage ont été implémentés.

Le premier mode de guidage permet à un utilisateur d'enregistrer une trajectoire avec un PHANToM, et de la rejouer à volonté. Les deux phases importantes sont l'enregistrement et la lecture. Pour commencer l'enregistrement l'utilisateur presse la touche R, et pour l'arrêter il presse la touche S. Le début de l'enregistrement coupe l'éventuel guidage en cours et ouvre le fichier où sera enregistré la trajectoire. Pendant l'enregistrement, l'agent enregistre toutes les positions de curseur envoyées par l'agent input du PHANToM. L'arrêt de l'enregistrement ferme le fichier et sort du mode enregistrement. Pour rejouer une trajectoire l'utilisateur presse la touche R, et pour l'arrêter il presse la touche S. Le début de la lecture désactive le rendu du circuit, et guide l'utilisateur vers le point de départ. Une fois ce point atteint, le ressenti du circuit est réactivé. L'utilisateur est attiré vers le point suivant de la trajectoire. À chaque fois que l'utilisateur passe suffisamment près d'un point[2], le guidage passe au point suivant. Le guidage est arrêté si l'utilisateur arrive à la fin de la liste de points.

Le second mode de guidage permet à un utilisateur de guider un autre utilisateur avec la souris. Ce mode est activé par une pression sur la touche F, et désactivé par une pression sur la touche S. Le mode de fonctionnement est analogue au mode précédent : l'agent maintient une liste de points de

[2]la distance est fixée par un paramètre

passage. À chaque déplacement de la souris, l'agent enregistre la position comme point de passage si la distance dépasse un seuil fixé. De manière symétrique le PHANToM est attiré vers le point en tête de liste. Une fois qu'il atteint ce point, il est attiré vers le suivant.

Ces deux systèmes ont quelques points communs. Dans les deux cas les PICOB sont désactivés pendant la lecture. En effet le but de ce guidage est de faire ressentir la forme du circuit. Les composants sont donc désactivés pour éviter de perturber l'utilisateur. Nous avons aussi ajouté un système permettant à l'utilisateur de retrouver son chemin s'il force et navigue sur d'autres parties du circuit (figure 34). En effet si l'utilisateur se trouve sur un brin opposé du circuit, la force de guidage $\overrightarrow{G_1}$ serait opposée à la force $\overrightarrow{F}$ de la contrainte sur le circuit. Du coup la résultante $\overrightarrow{R}$ ne permettrait ni de ressentir correctement le circuit, ni d'être guidé correctement. Nous enregistrons donc le passage dans chaque nœud et ajoutons ces points en tête de liste de guidage, qui généreront les forces $\overrightarrow{G_2}$, $\overrightarrow{G_3}$ et $\overrightarrow{G_4}$ sur notre exemple. Enfin un curseur bleu est affiché à la position du point de guidage.



Figure 34 – Ajout de positions intermédiaires

Enfin le troisième système de guidage est visuel. Nous avons gardé le code de couleur des fils des prototypes, avec les couleurs pour les fils déjà parcourus, les fils à parcourir et le fil actuellement exploré (figure 22, page 148). Les autres indications visuelles sont les différents curseurs. Un curseur rouge indique la projection de la souris sur le circuit, un curseur jaune indique la projection du PHANToM sur le circuit et un curseur bleu indique la position du point de guidage lorsque le guidage est actif. Ces indicateurs visuels sont utiles pour les utilisateurs à vision résiduelle, ainsi que pour les utilisateurs voyants qui travaillent en collaboration avec un utilisateur déficient visuel.

3.3.3. Techniques d'interaction

Les techniques d'interaction proposées par les agents développés dans l'application d'exploration de schémas de circuits électriques sont résumées dans la table 6. Nous reprenons les définitions de Nigay et Coutaz [123], où une technique d'interaction est définie comme un couple formé d'un périphérique physique et d'un langage d'interaction. Alors que dans la section précédente nous avons évoqué les interactions du point de vue des agents et de leur fonctionnement, nous allons évoquer ici les tâches associées à ces techniques d'interaction.

Tâche	Techniques d'interaction	
	Périphérique	**Langage d'interaction**
Reconnaître un composant	écran PHANToM VTPlayer	symbole PICOB semi-guidés Tactons statiques
Reconnaître la forme du circuit	écran PHANToM écran PHANToM PHANToM	schéma 2D attraction vers le schéma curseur bleu guidage par la souris suivi d'un enregistrement
Reconnaître les directions possibles	écran PHANToM VTPlayer	schéma 2D + curseur jaune PICOB guidés Tactons directionnels
Reconnaître les directions déjà parcourues	écran PHANToM	couleur des fils amplitude des PICOB guidés
Connaître la position de la souris et du PHANToM	écran	curseurs rouge et jaune

Table 6 – Techniques d'interaction

La première tâche se situe au niveau local d'exploration : il s'agit de la reconnaissance des composants. Cette tâche peut être complétée de manière redondante grâce à tous les périphériques disponibles. Premièrement, le symbole du composant peut être reconnu visuellement. La représentation choisie est basée sur les représentations officielles des composants électroniques. Deuxièmement, les PICOB semi-guidés de la section 5 du chapitre III sont utilisés pour coder les principaux composants. Enfin troisièmement, nous avons créé des Tactons statiques basés sur les définitions du chapitre II pour représenter chaque composant à l'aide de la souris VTPlayer.

La deuxième tâche que nous avons identifiée est la reconnaissance la forme du circuit. C'est la tâche principale au niveau d'exploration global des schémas. Elle peut naturellement se faire grâce à la représentation visuelle, mais elle peut aussi être réalisée en faisant attention à l'attraction du PHANToM sur le schéma. En cas de difficultés, l'utilisateur peut bénéficier des deux techniques de guidage du PHANToM : par la souris ou le suivi d'une trajectoire préalablement enregistrée. La VTPlayer n'est pas utilisée pour cette tâche. Nous envisageons cependant d'utiliser les résultats des expériences de suivi de formes géométriques. Ces travaux n'étant pas assez avancés, nous n'avons pas encore intégré cette technique d'interaction à ce stade.

La troisième tâche consiste à reconnaître les directions possible lors de l'exploration. Elle se situe aussi au niveau global d'exploration. Il s'agit de savoir dans quelle direction se trouvent les parties du circuits autour de la position actuelle. Ceci est particulièrement utilisé aux nœuds car c'est

l'endroit où plusieurs fils et composants se croisent, et aussi là où ils changent de direction. Cette tâche peut naturellement être exécutée en observant la portion de circuit autour du curseur jaune. Il est aussi possible de demander de l'aide en cliquant sur le bouton du PHANToM. À ce moment deux types d'informations sont données à l'utilisateur de manière redondante : des Tactons directionnels sur la VTPlayer tels que ceux développés dans le chapitre II, et des PICOB guidés décrits dans le chapitre III.

La quatrième tâche a pour but de déterminer si un fil ou un composant a déjà été parcouru ou non. Cette tâche peut être vue comme une sous-tâche de la troisième tâche. Les techniques d'interaction sont donc liées. Deux techniques d'interaction permettent de remplir cette tâche du niveau global d'exploration. La première consiste à observer la couleur des fils et des composants. Elle est sombre pour ceux qui ont déjà été parcourus, brillante pour celui actuellement parcouru, et intermédiaire pour les autres. La deuxième utilise les PICOB guidés de la tâche précédente. Alors que l'information pour la troisième tâche était codée par la direction des impulsions, l'information pour cette tâche est codée par l'amplitude des impulsions. Nous envisageons d'utiliser un paramètre supplémentaire, comme la vitesse de clignotement, sur les Tactons de la troisième tâche pour coder cette information avec la souris VTPlayer. Cependant les travaux de la section 4 du chapitre II n'étaient pas aboutis au moment de la conception de l'application.

La cinquième tâche a pour objectif de connaître la position du PHANToM et de la souris sur le schéma. Elle ne peut se réaliser directement que de manière visuelle, en observant les curseurs rouge et jaune pour la souris et le PHANToM respectivement. Nous n'avons pas développé de technique d'interaction haptique car d'une part ce canal est déjà chargé. De plus seule la position de la souris est nécessaire car le PHANToM est contrôlé par l'utilisateur utilisant le retour haptique, et celle-ci peut être indiquée de manière indirecte grâce au guidage par la souris de la deuxième tâche.

3.3.4. Évaluation

Cette application a fait partie des trois applications retenues pour l'évaluation finale du projet européen MICOLE [113]. Nous avons pu à cette occasion évaluer son utilisation par des enfants déficients visuels, en collaboration avec un enseignant. Le but de cette évaluation est de tester l'application en conditions réelles d'utilisations. Ceci implique notamment que l'utilisateur déficient visuel puisse être aidé par un autre utilisateur, et que les utilisateurs ayant une vision résiduelle puissent s'aider de leur vision partielle. L'aspect collaboratif est donc mis en avant. Nous nous sommes focalisés sur les deux premières tâches définies dans la section précédente, à savoir la reconnaissance des composants et la reconnaissance de la forme du circuit. En effet les autres tâches sont des tâches secondaires permettant de réaliser ces deux tâches principales.

Méthodologie.

13 utilisateurs déficients visuels âgés de 9 à 17 ans (médiane : 15 ans) ont participé à cette évaluation. Trois d'entre eux sont aveugles (utilisateurs 11 à 13 sur la table 7), dont deux sont aveugles de naissance (utilisateurs 11 et 12). Deux des utilisateurs aveugles possèdent aussi une

légère déficience mentale (utilisateurs 12 et 13). Les utilisateurs venaient de trois établissements accueillant des enfants déficients visuels. Les quatre utilisateurs les plus jeunes (utilisateurs 1 à 4) viennent de l'école primaire Saint Eucaire à Metz. Deux des plus vieux (utilisateurs 10 et 11) étaient scolarisés au lycée Robert Schuman à Metz, et enfin les autres font partie du centre Santifontaine à Nancy. Parmi ces derniers, quatre sont scolarisés dans des établissements avec des enfants voyants. Pour ces utilisateurs nous nous sommes rendus à leur domicile pour réaliser les tests, hors jours scolaires (samedi, jour fériés, et vacances). Les autres utilisateurs ont exécuté leurs tests dans leur établissement.

Utilisateur	Âge	Sexe	Vision	Connaissances haptiques
1	9	M	5	joypad vibrant
2	11	M	0,05 / 0,2	PHANToM, VTPlayer, joypad vibrant, joystick à retour de force
3	11	M	2	PHANToM, VTPlayer, joypad vibrant, joystick à retour de force
4	11	M	2	PHANToM, VTPlayer
5	13	M	0,05	joypad vibrant, joystick à retour de force
6	14	M	1 / 4,5	joypad vibrant, joystick à retour de force
7	15	M	0 / 0,1	joypad vibrant
8	16	M	1,4 / 2,4	joypad vibrant
9	16	M	2 / 0,5	joypad vibrant, joystick à retour de force
10	17	F	0,5 / 0,1	PHANToM, VTPlayer, lis le Braille
11	17	F	Aveugle de naissance	PHANToM, clavier braille, lis et écris le braille
12	17	M	Aveugle de naissance	lis le Braille, papier en relief, machine Perkins
13	17	M	Aveugle	lis le Braille, papier en relief, machine Perkins, joypad vibrant, joystick à retour de force

Table 7 – Utilisateurs de l'application finale

Les utilisateurs 2, 3, 4, 10 et 11 ont déjà utilisé le PHANToM et la souris VTPlayer lors de tests précédents avec notre équipe. La plupart des autres utilisateurs avaient déjà utilisé des contrôleurs de jeux à retour haptique tels que les joypads vibrants ou les joysticks à retour de force. Quatre utilisateurs (utilisateurs 10, 11, 12 et 13) savent lire le braille, dont trois savent l'écrire : l'utilisateur 11 avec un clavier braille, et les utilisateurs 12 et 13 avec une machine à écrire braille Perkins [133]. Au niveau des connaissances en circuits électriques, tous les utilisateurs avaient déjà étudié cette notion à l'école. Cependant les utilisateurs 1 à 4, 11 et 12 ne connaissaient que la pile et la lampe.

Les expériences passées nous ont montré que les tâches que nous allons demander aux utilisateurs ne sont pas simples à réaliser. Notre but est d'obtenir un système utilisable, donc nous ne cherchons pas à tout prix à comparer cette version avec les précédentes. Nous n'avons dont pas hésité à simplifier les formes de circuits et réduire le nombre de composants afin de rendre les tâches réalisables par tous les utilisateurs. Nous n'avons donc utilisé que trois formes de circuit : un carré, un rectangle et deux rectangles superposés (figure 35). Cependant nous n'avons pas demandé aux utilisateurs de distinguer le carré du rectangle, car la différence entre la longueur et la largeur est faible, et est difficile à apprécier avec un PHANToM d'après notre expérience personnelle. De plus cette distinction n'a pas de pertinence dans le cas des circuits électriques.

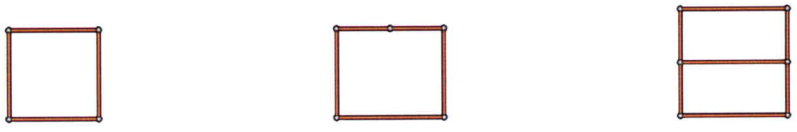


Figure 35 – Formes de circuits de l'application finale

Certains utilisateurs sont relativement jeunes et ont peu de connaissances en circuits électriques. Nous ne souhaitons pas mesurer les connaissances des utilisateurs en matière de circuits électriques, mais mesurer la pertinence de la représentation des schémas de circuits électriques avec notre application. Nous avons donc restreint les composants possibles à trois : la pile et la lampe qui sont connues de tous les utilisateurs, et la résistance qui est inconnue par certains utilisateurs. Nous avons donc conçu des circuits avec les formes ci-dessus, et les trois composants retenus. De plus les tâches à réaliser ne demandent pas de connaissances qui pourraient départager les utilisateurs *a priori*. Nous avons gardé comme contrainte de n'orienter les composants qu'horizontalement car les expériences précédentes ont montré dans la section 5 du chapitre III que l'orientation pouvait perturber la reconnaissance des composants. Nous avons aussi conçu un circuit d'apprentissage (circuit 0 sur la figure 36) n'utilisant pas une forme de circuit réservée aux circuits de tests, et sur lequel figure les trois composants retenus.

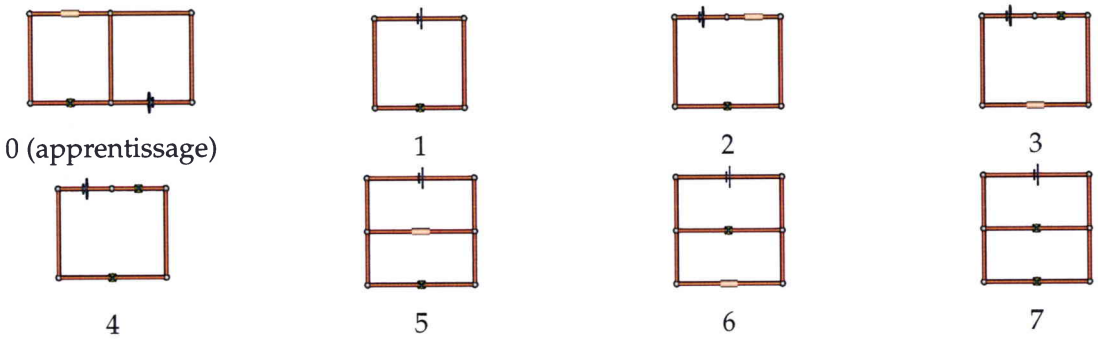


Figure 36 – Circuits utilisés dans les tests finaux

Chaque utilisateur commence par remplir un pré-questionnaire (voir annexe A), dont les informations recueillies sont réunies dans la table 7. L'expérimentateur joue le rôle de l'enseignant lors de ces expériences afin que tous les utilisateurs travaillent avec le même enseignant. Après avoir expliqué le but de l'expérimentation et décrit le PHANToM et la souris VTPlayer, l'expérimentateur exécute quelques démonstrations afin que l'utilisateur puisse se familiariser avec ces périphériques. Les applications de démonstration pour la souris VTPlayer sont des démonstrations des Tactons utilisées lors d'exhibitions telles que la Fête de la Science. Les démonstrations utilisées pour le PHANToM sont les exemples fournis par le fabriquant, ainsi que les exemples de l'API Reachin. Après cette phase durant environ 10 minutes, l'utilisateur passe à l'application de circuits électriques.

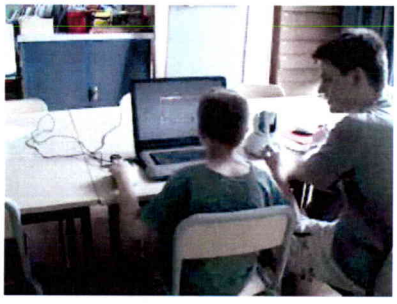


Figure 37 – Un utilisateur et un enseignant

L'expérimentateur charge l'application d'exploration de schémas de circuits électriques avec le circuit de démonstration. Comme le montre la photographie de la figure 37, l'utilisateur tient le PHANToM avec sa main dominante, et pose les doigts de sa main non-dominante sur les matrices de la souris VTPlayer. Les fonctionnalités de l'application sont expliquées : représentations des composants, informations directionnelles et méthodes de guidage. L'expérimentateur illustre chaque fonctionnalité par un exemple, notamment pour le guidage du PHANToM par la souris. L'utilisateur peut parcourir ce schéma à volonté afin de s'habituer à l'exploration d'un circuit, et apprenne les représentations des composants. Typiquement cette phase dure entre 5 et 10 minutes.

Avant de commencer les tests, les instructions sont données à l'utilisateur. Le but est d'explorer 6 schémas un après l'autre. Les circuits sont parcourus en ordre balancé parmi les utilisateurs afin d'éliminer un éventuel phénomène d'apprentissage (table 8). Pour chaque circuit, nous alternons deux conditions :

1. l'enseignant ne donne aucune indication à l'utilisateur, et ne fait que recueillir les réponses.
2. l'enseignant peut donner des indications à l'utilisateur, soit à son initiative, soit à la demande de l'utilisateur, mais toujours après consentement de celui-ci.

Ces deux conditions ont pour but de mesurer l'impact de la collaboration sur l'efficacité d'utilisation de l'application. La moitié des utilisateurs commence par la condition 1 (utilisateurs 1, 2, 5, 6, 9, 11 et 12), et l'autre moitié par la condition 2 (utilisateurs 3, 4, 7, 8, 10 et 13). Ensuite les conditions sont alternées à chaque circuit présenté. L'utilisateur a pour instruction de reconnaître, et d'annoncer

Utilisateur	Session (circuit) 1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						

Table 8 – Circuits explorés par les utilisateurs

les composants qu'il parcourt. Il doit aussi reconnaître et annoncer la forme du circuit dès qu'il l'a reconnue. Le nombre de formes différentes n'est pas précisé. L'utilisateur doit décrire verbalement la forme qu'il a reconnue. À la fin des 6 circuits, l'utilisateur remplit un post-questionnaire (voir annexe A). Ce questionnaire est constitué d'énoncés auxquelles l'utilisateur doit adhérer ou non. Les réponses à donner vont de 1 à 4, 1 signifiant « pas d'accord » et 4 signifiant « d'accord ». Il n'y a pas de réponse intermédiaire afin de forcer l'utilisateur à prendre position. La formulation des énoncés alterne des formes positives et négatives pour éviter que les utilisateurs répondent toujours « d'accord » ou toujours « pas d'accord ». Les énoncés sont les suivants :

- Les composants sont difficiles à identifier.
- Il est difficile de parcourir les circuits en entier.
- Le fait d'être contraint sur le circuit aide à parcourir les circuits.
- Le guidage du PHANToM par la souris n'aide pas à parcourir les circuits.
- Les Tactons directionnels aident à parcourir les circuits.
- Les Tactons sont difficiles à reconnaître.

- J'ai réussi à identifier tous les composants.
- Les Tactons sont plus utiles que les PICOB pour identifier les composants.
- La présence de l'enseignant aide à mieux comprendre le circuit.
- La méthode traditionnelle permet de mieux comprendre des schémas de circuits électriques.

Concernant les données enregistrées, nous avons filmé les tests de manière à ce que les deux mains et au moins une partie de l'écran soit visible. Nous avons aussi conservé une trace des messages transitant sur le bus. Grâce à ces messages nous pouvons reconstruire la trajectoire des utilisateurs, savoir les composants parcourus, les techniques d'interaction utilisées, etc.

Nous nous attendons à plus d'erreurs de reconnaissance de forme de circuit que d'erreurs de reconnaissance de composants. En effet contrairement à la vision, l'haptique ne permet pas d'avoir une vision globale d'un environnement.

Résultats.

Les réponses de chaque utilisateur pour la forme des circuits sont reportées dans la table 9. Seuls les utilisateurs ayant commis des erreurs sont représentés. Pour rappel, les détails des utilisateurs figurent sur la table 7. Ils sont classés par âge, et les trois aveugles sont les utilisateurs 11, 12 et 13. Ils sont aussi les plus âgés avec 17 ans. Nous constatons que les neuf premiers utilisateurs n'ont fait aucune erreur de reconnaissance de forme. Ces trois utilisateurs et un utilisateur à faible vision résiduelle (0,5 pour l'œil gauche et 0,1 pour l'œil droit) ont cependant donné de mauvaises réponses. Bien que les deux derniers utilisateurs aient toujours répondu « carré », les deux autres utilisateurs ont donné d'autres réponses.

L'utilisateur 10 a commis une erreur avec le circuit 5 en répondant « deux rectangles côte à côte ». Le problème vient de l'orientation, comme nous l'avions déjà remarqué avec des utilisateurs aveugles dans l'expérience de la section 3.2.2. L'utilisateur 10 a une vision résiduelle mais n'a pas souhaité s'en servir pour cette expérience. Notons que cet utilisateur a parcouru deux autres circuits avec la même forme par la suite, et a répondu correctement. De plus sa réponse était « deux rectangles un devant l'autre » à la place de « deux rectangle un au-dessus l'autre », ce qui a un sens que du point de vue du retour de force, car le circuit est placé sur le plan horizontal.

Enfin l'utilisateur 11 a répondu « rectangle » pour le circuit 6 alors qu'il s'agissait de deux rectangles un au-dessus de l'autre. Or cet utilisateur a reconnu les trois composants de ce schéma et donc a parcouru le circuit en entier. Cet utilisateur a aussi répondu « deux rectangles l'un dans l'autre » pour le circuit 5 alors que les rectangles sont l'un au-dessus de l'autre. Il y a donc un problème de repère entre l'exploration des deux rectangles. D'ailleurs à plusieurs reprises cet utilisateur a tenté de se repérer par rapport à un point en bas à gauche.

Les utilisateurs 12 et 13 ont fait de nombreuses erreurs d'exploration. En particulier, ils ont toujours répondu « carré » pour la forme des circuits. Contrairement aux autres utilisateurs il y a des composants qu'ils n'ont pas parcouru. Dans certains cas cela pouvait influer sur la perception du circuit, comme l'utilisateur 13 qui n'a pas parcouru la pile sur le circuit 5. Dans d'autres cas il s'agissait des circuits avec deux composants alignés, comme lorsque l'utilisateur 12 n'a pas exploré la résistance sur le circuit 2.

Utilisateur	Session (circuit) 1	2	3	4	5	6
10		5/deux rectangles côte à côte				
11		6/rectangle			5/deux rectangles l'un dans l'autre	
12		6/carré, pas exploré la lampe	2/pas exploré la résistance	5/carré, pas exploré la lampe		4/pas exploré la lampe
13	5/carré, pas exploré la pile		4/pas exploré la lampe du dessus	7/carré sans explorer la pile	2	6/carré

Table 9 – Erreurs de reconnaissance de la forme des circuits

Nous observons au passage que l'utilisateur 7 a différencié les circuits rectangles des circuits carrés, alors que ce n'était pas demandé. En effet il a répondu « rectangle » pour les circuits 4 et 3, et « carré » pour le circuit 1. Nous acceptions les deux réponses pour les circuits carrés et les circuits rectangles car la forme était similaire.

Les réponses des utilisateurs concernant les composants sont reportées dans la table 10. La réponse considérée est la dernière donnée par l'utilisateur. Le nombre de réponses données sont aussi notées car il dépend des circuits explorés, ainsi que du nombre de composants explorés. De manière globale les utilisateurs ont fait peu d'erreurs d'identification de composants. En particulier nous pouvons observer que les plus jeunes utilisateurs (utilisateurs 1 à 4) qui ne connaissaient pas les résistances n'ont eu aucune difficulté à les distinguer. En effet quand des utilisateurs ne se souvenaient plus du nom « résistance » mais qu'ils savaient que c'était ce composant, ils disaient « l'autre composant ».

Utilisateur	1	2	3	4	5	6	7	8	9	10	11	12	13
Nombre de réponses	17	18	17	17	17	18	17	17	17	17	17	13	14
Mauvaises réponses	0	0	0	0	3	0	1	0	0	0	0	3	3

Table 10 – Erreurs de reconnaissance de composants

Le détail des confusions est reporté dans la table 11, où chaque erreur pour chaque utilisateur et chaque session est détaillée. Ainsi on remarque que les utilisateurs ont, à deux exceptions près,

toujours confondu la lampe et la résistance. Les utilisateurs 5 et 7 ont une vision résiduelle, alors que les utilisateurs 12 et 13 sont aveugles, et ont une légère déficience mentale.

	Session (confusions)					
Utilisateur	1	2	3	4	5	6
5	R/L L/R	R/L				
7		P/L				
12	L/R		R/L		R/L	
13	L/R P/L				L/R	

L = lampe
R = résistance
P = pile

Table 11 – Confusions entre composants

L'utilisateur 5 a commencé par utiliser le PHANToM pour reconnaître les composants. Il a inversé la résistance et l'ampoule sur le premier circuit, ce qui lui a valu 2 erreurs. Sur le circuit suivant il a commis la même erreur en répondant résistance à la place de lampe. Cependant comme cette session était en condition 2, c'est-à-dire avec aide possible de l'enseignant, l'erreur lui a été signalée. À partir du troisième circuit exploré cet utilisateur a utilisé la souris VTPlayer pour reconnaître les composants, et n'a plus fait d'erreurs. L'utilisateur 7 a répondu « pile » à la place de lampe sur le circuit 6. Cependant il a dit n'avoir senti qu'une seule pulsation. Il a donc manifestement mal retenu la correspondance entre les codes et les composants. L'enseignant n'a pas pu faire la remarque à ce moment là car ce circuit était présenté en condition 1. L'utilisateur 12 a eu des difficultés pour reconnaître les composants. Premièrement il n'a pas exploré 4 des 17 composants qu'il avait à parcourir sur les 6 circuits. Deuxièmement parmi les 13 réponses données, 3 sont erronées. Il a répondu une fois lampe à la place de résistance, et deux fois résistance à la place de lampe. Il nous a confié avoir du mal à reconnaître les Tactons désignant la lampe et la résistance. Cet utilisateur étant brailliste, il a pu être perturb à la fois par la taille de la matrice à laquelle il n'est pas habitué, et à des motifs qu'il n'est pas habitué à ressentir et donc à interpréter. L'utilisateur 13 a aussi eu des difficultés pour reconnaître les composants. Il n'a parcouru que 14 composants sur les 17 possibles, et a donné 3 mauvaises réponses. 2 des réponses ont été commises sur le premier circuit exploré. Il a répondu lampe à la place de résistance alors qu'il a ressenti un carré sur la VTPlayer et deux bosses sur le PHANToM. Pour la deuxième erreur il a répondu pile à la place de lampe, en ayant senti une croix sur la VTPlayer et deux bosses sur le PHANToM. Nous en déduisons dans un premier temps qu'il a oublié la signification des codes. D'ailleurs l'enseignant les lui a rappelés après qu'il ait validé ses réponses car c'était en condition 2. Nous déduisons aussi qu'il a bien reconnu les Tactons de la VTPlayer, mais pas les impulsions sur le PHANToM. Il a beaucoup hésité à propos d'une lampe sur le troisième circuit exploré. Comme ce circuit était exploré en condition 2, l'enseignant a pu l'aider en lui suggérant de compter les bosses. Il s'est ainsi décidé pour la bonne réponse. Enfin il a répondu ampoule à la place de résistance, en ayant reconnu un carré sur le VTPlayer et deux bosses sur le PHANToM. L'erreur est la même que sur la résistance du premier circuit parcouru. Il n'y a pas eu

de circuit avec une résistance entre le premier et le cinquième circuit parcouru. Par contre il y avait à nouveau une résistance sur le sixième et dernier circuit. Cette fois-ci l'utilisateur l'a reconnu après quelques hésitations.

Au niveau de la collaboration, les utilisateurs 1, 5 et 13 y ont eu recours pour les composants. L'enseignant leur a rappelé les noms et les codages des composants. Par contre deux utilisateurs aveugles (11 et 12) ont refusé l'aide de l'enseignant pour les composants, préférant chercher par eux-mêmes. Les trois utilisateurs aveugles (10, 11 et 12) ont éprouvé des difficultés à explorer les circuits, et ont accepté l'aide de l'enseignant à quelques reprises. Cependant les résultats montrent que cela n'a pas suffi. Trois utilisateurs ont eu recours au guidage du PHANToM par la souris (7, 12 et 13) mais seul l'utilisateur 7 l'a jugé utile. Alors que plusieurs utilisateurs ont rempli leurs tâches sans difficultés, 7 d'entre eux révèlent dans le post-questionnaire avoir apprécié l'aide de l'enseignant (1, 5, 6, 7, 9, 12 et 13). Trois d'entre eux sont ceux ayant eu recours à l'aide de l'enseignant pour les composants, deux d'entre eux sont aveugles et les trois utilisateurs ayant utilisé le guidage en font partie.

Discussion.

Dans un premier temps nous pouvons souligner la satisfaction d'utilisation de l'application. Seul l'utilisateur 7 préfère utiliser des schémas agrandis ou en relief. Le fait d'utiliser un outil pouvant être aussi utilisé par des voyants les a mis à l'aise. De plus l'aspect ludique d'une application informatique n'est pas à exclure. Comme nous l'avons vu, chaque utilisateur possède ses propres particularités. Ils ont des déficiences différentes, de différentes approches face aux tâches demandées et la tranche d'âge est large. Il est par conséquent difficile voire impossible de généraliser. Du coup cette évaluation, qui s'apparente plus à une étude de cas, nous permet de tirer beaucoup d'enseignements.

Deux tâches principales étaient demandées aux utilisateurs : au niveau global ils devaient reconnaître la forme du circuit, et au niveau local ils devaient reconnaître les composants. Ainsi que nous l'avions prévu, la reconnaissance de la forme des schémas a posé plus de problèmes que la reconnaissance de composants. Deux des trois utilisateurs aveugles ont toujours répondu « carré » pour la forme de circuit. Ces deux utilisateurs ont une légère déficience mentale, ce qui a pu affecter leurs résultats. L'autre utilisateur aveugle a aussi eu des difficultés à reconnaître les formes de circuits. Il a reconnu tous les circuits rectangles ou carrés, mais par contre il a eu du mal à mentaliser les circuits plus compliqués. Il a tenté à plusieurs reprises de partir d'un point de départ qu'il s'était fixé. Dans un cas il a répondu rectangle en ayant parcouru les deux rectangles superposés. Dans un autre cas il pensait qu'un rectangle se trouvait à l'intérieur de l'autre. Il est évident que notre représentation manque des points de repère. Les informations directionnelles que nous avons placées ne sont que locales et ne permettent pas de situer les objets explorés. Pour améliorer ce point nous pourrions placer des étiquettes à chaque nœud afin que l'utilisateur puisse reconstruire mentalement le circuit. L'avantage des étiquettes est que l'utilisateur pourrait savoir quel nœud il parcourt et se référer à ce qu'il a déjà parcouru. Wall et Brewster ont utilisé une technique similaire pour aider des utilisateurs à parcourir des histogrammes [169]. Cependant les marque-pages qu'ils

ont conçus sont placés par l'utilisateur. Cette idée serait aussi à exploiter. Enfin l'utilisateur a commis une erreur de reconnaissance. Il a une vision résiduelle, mais n'a pas souhaité s'en servir. Il a confondu l'orientation des circuits, ce qui confirme les résultats que nous avions obtenus avec des utilisateurs aveugles du second prototype (section 3.2.2). Il est le seul à avoir utilisé les PICOB guidés. Il est possible que les utilisateurs ne l'ont pas utilisé car le nombre de techniques d'interaction proposées est trop élevé, et ils n'ont pas su assimiler la quantité d'instructions en si peu de temps.

La qualité de l'exploration locale reposait principalement sur l'assimilation des codages des composants. En effet contrairement à l'exploration globale, il était plus facile de recevoir l'information, mais le problème était de bien la lire et de bien l'interpréter. Nous avons facilité la lecture en simplifiant les codes : seuls trois composants étaient utilisés, ce qui donnait trois PICOB semi-guidés, qui ne variaient que par le nombre d'impulsions. Il n'y avait que 5 Tactons : 3 pour les composants, 1 pour les fils et 1 pour les nœuds. L'affichage de ces informations de manière redondante a permis à la plupart des utilisateurs de reconnaître les composants. 7 utilisateurs ont préféré utiliser les PICOB semi-guidés pour identifier les composants, alors que les 6 autres préféraient les Tactons. Aucune méthode n'est donc préférable à l'autre : c'est une affaire de goût. Ceci rejoint un des 10 mythes sur les systèmes multimodaux de Sharon Oviatt [131] : « Les utilisateurs n'organisent pas leur multimodalité de la même manière ». Certains ont changé de méthode pendant l'expérience, comme l'utilisateur 5 qui a fait 3 erreurs sur les deux premiers circuits en utilisant le PHANToM, avant d'utiliser la souris VTPlayer et ne plus faire d'erreur. Plusieurs utilisateurs ont eu des difficultés à se souvenir des codes des composants. L'utilisateur 7 a répondu pile à la place de lampe alors qu'il avoue avoir reconnu une impulsion et l'utilisateur 13 n'a pas répondu correctement par rapport aux impulsions qu'il a senties à plusieurs reprises. Il a aussi répondu faux en ayant reconnu le Tacton correspondant sur chacune de ses erreurs. La présence de l'enseignant a eu un impact positif sur cet utilisateur car il n'a fait des erreurs qu'en condition 1 (sans aide de l'enseignant) et a fait 2 de ses 3 erreurs sur le premier circuit qu'il a exploré. La présence de l'enseignant facilite l'apprentissage. Un apprentissage plus long pourrait diminuer sérieusement les erreurs. D'une part parce que les utilisateurs seraient habitués aux techniques d'interaction et seraient plus efficaces sur la phase de lecture. D'autre part parce qu'ils mémoriseraient mieux les codes et seraient plus efficaces sur la phase d'interprétation. L'utilisateur 12 a avoué avoir eu des difficultés à comprendre les Tactons. Cet utilisateur lit couramment le Braille, et était tenté de lire les Tactons comme des lettres Braille. Nous pensons donc que les Tactons peuvent déstabiliser les braillistes, du moins lors de leurs premières utilisations. À l'inverse une partie des utilisateurs non braillistes avaient des réticences à utiliser des matrices Braille. Cependant le fait que nous utilisons des icônes et non l'alphabet Braille a dissipé ces réticences. Cette technique d'interaction pourrait être une bonne introduction au Braille en permettant aux utilisateurs de passer le cap psychologique de l'utilisation de matrices à picots.

La collaboration a fonctionné dans certains cas. Lors de la reconnaissance de composants, elle a été nécessaire pour certains utilisateurs car ils avaient du mal à retenir la correspondance entre les représentations haptiques et les noms des composants. Les utilisateurs aveugles ont refusé l'aide pour les composants mais l'ont accepté pour l'exploration de la forme du circuit. C'est un indice

révélant qu'il est plus difficile de reconnaître la forme que les composants. Le refus de l'aide pour les composants s'explique par le fait que ces utilisateurs pensaient pouvoir se débrouiller seuls. Le guidage du PHANToM par la souris a été peu et mal utilisé. Nous songeons à étudier plus en détails une manière d'utiliser plus efficacement ce système.

4. Conclusion

Nous avons étudié dans ce chapitre une nouvelle architecture permettant de faciliter le développement d'applications multimodales pour enfants déficients visuels. D'une part le développement est facilité car la communication avec les périphériques supportés se fait de manière uniforme : il n'y a pas besoin d'utiliser une API par périphérique. D'autre part, des techniques d'interaction telles que celles présentées dans les chapitres II et III sont déjà implémentées. Cette architecture est une collection d'agents communiquant par un bus logiciel. Le fonctionnement du bus a été calqué sur Ivy [33] : les messages envoyés sur le bus sont récupérés par des motifs sous forme d'expressions régulières, et déclenchent des callbacks. Cependant nous n'avons retenu que la partie qui nous intéresse afin de pouvoir supporter une charge de travail lourde.

La partie la plus importante de l'architecture est le framework multimodal. Il s'agit de règles de conception d'agents pour les périphériques d'entrée et de sortie. Grâce à ces règles la communication avec ces agents est normalisée, et le développement d'applications est facilité. Ce framework définit trois types d'agents : les agents sender, les agents input et les agents output. Les agents sender et input sont utilisés par les périphériques d'entrée tandis que les agents output sont utilisés par les périphériques de sortie. L'agent sender interroge le périphérique et envoie les entrées sur le bus. L'agent input récupère les données de l'agent sender. Lorsqu'on développe un agent sender il est conseillé de développer un agent input abstrait correspondant, que le développeur d'applications n'a qu'à compléter pour lui donner le comportement souhaité. Les agents output proposent des services liées aux sorties offertes par le périphérique en question. Il intègre aussi des techniques d'interaction qu'il est possible de déclencher d'un simple message. L'architecture contient des agents pour gérer des périphériques courants comme la clavier ou la souris, ainsi que des périphériques tels que le PHANToM ou la souris VTPlayer. Les agents output de ces derniers permettent d'utiliser les icônes développées dans les chapitres II et III

La seconde partie de ce chapitre traite de l'application d'exploration de schémas de circuits électriques. Le développement de cette application s'est déroulé en trois phases. Le premier prototype avait pour but de définir l'allure générale de l'application. Les interactions proposées sont limitées : l'exploration se fait à l'aide du PHANToM, et en retour l'utilisateur est attiré vers le circuit par une force magnétique.L'évaluation de ce premier prototype était informelle, et par des utilisateurs voyants et non spécialistes. Les utilisateurs ont émis quelques remarques intéressantes. Premièrement la force magnétique n'est pas adaptée car elle décroît lorsque la distance augmente. Deuxièmement il était possible de sortir du circuit, ce qui fait que l'utilisateur était facilement perdu.

Le second prototype est une amélioration du premier, suite aux remarques des utilisateurs. Nous avons placé des indications directionnelles avec des PICOB guidés aux nœuds pour aider les utilisateurs à se déplacer dans le circuit, et connaître les directions possibles. Ces indications sont lues automatiquement à l'entrée d'un nœud et peuvent ensuite être déclenchées à l'initiative de l'utilisateur par un clic. L'exploration d'un circuit avec ce prototype a été testé par des utilisateurs déficients visuels. Le but était de tester les indications ajoutées, et d'avoir des remarques afin d'améliorer l'application. Les résultats montrent que les utilisateurs aveugles ont eu de grandes difficultés à parcourir les circuits. Il y a deux raisons principales à cela. Premièrement ils ont l'habitude d'être maître de l'exploration de leur environnement. Cependant nous avons ajouté des informations qui se lisent de manière automatique et qui empêchent l'exploration tant que la lecture n'est pas finie. Cela a perturbé les utilisateurs aveugles. Nous avons donc décidé dans la version suivante de ne plus jouer ces indications automatiquement. Deuxièmement pour contraindre l'utilisateur sur le schéma nous projetions le PHANToM sur la position la plus proche du circuit. Or dans certains cas les utilisateurs pouvaient être téléportés d'un endroit à l'autre du circuit. Ce faisant ils perdaient tout repère dans le circuit. Les utilisateurs à vision résiduelle avaient les yeux bandés afin de ne mesurer que l'apport des techniques d'interaction haptiques. Ils ont moins été gêné par les PICOB guidés, et au contraire ont eu plutôt tendance à s'en servir. Ils ont cependant été aussi perturbés par les téléportations. La reconnaissance de circuits s'est avéré être une tâche très difficile avec le retour de force comme seule méthode d'interaction.

Nous avons encore une fois apporté des améliorations motivées par les remarques des utilisateurs. De plus nous avons ajouté des techniques d'interaction. En premier lieu nous avons enlevé la lecture automatique des PICOB guidés dans les nœuds. Les utilisateurs peuvent toujours le demander par un clic lorsqu'ils désirent recevoir de l'aide. En second lieu nous avons changé la projection du PHANToM vers le circuit, en n'autorisant les déplacements qu'entres parties connexes. Ainsi il n'est plus possible de se téléporter dans le circuit. La grande modification de cette version est qu'elle a été portée sur l'architecture MICOLE. La partie présentation de l'application est assurée par les agents du framework multimodal. L'abstraction est constituée des classes qui stockent et manipulent directement le modèle interne du schéma. Et enfin la partie contrôle est construite à partir d'une collection d'agents dont certains analysent l'exploration des utilisateurs, et d'autres génèrent les ordres à destination des agents de la partie présentation. Désormais il est possible d'explorer le circuit avec la souris. Le seul retour en échange est un curseur de couleur. Cette exploration est réservée à des utilisateurs voyants, comme un enseignant ou un autre élève, qui peut guider le PHANToM sur une trajectoire. Cela permet de renforcer l'aspect collaboratif de l'application. L'utilisation de l'architecture permet de développer facilement des agents permettant d'afficher la même information que les autres, mais avec une autre modalité. Ainsi nous avons ajouté des Tactons sur la souris VTPlayer pour représenter les composants, les fils et les nœuds. De plus les indications données par des PICOB guidées ont été complétées par des Tactons directionnels se jouant en parallèle.

Cette application a été testée par des utilisateurs déficients visuels Ces tests avaient pour but d'évaluer l'utilisation de ce logiciel de manière collaborative, et en situation réelle. Les utilisateurs ayant une vision résiduelle ont donc pu s'en servir. L'enfant déficient visuel était toujours accompagné d'un expérimentateur jouant le rôle d'enseignant. Les deux tâches principales des utilisateurs étaient d'identifier la forme des circuits, et d'identifier les composants. Les résultats ont montré que la reconnaissance de la forme des circuits est problématique sans la vision. En effet les seuls utilisateurs ayant commis des erreurs à l'exploration sont les trois utilisateurs aveugles et un utilisateur n'ayant pas souhaité utiliser sa vision résiduelle. Une partie des autres utilisateurs a même reconnu la forme des circuits avant même de les avoir parcourus en entier. Par contre la reconnaissance des composants n'était pas directement liée au niveau de vision. Les problèmes de reconnaissance des composants sont liés à l'association des codes PICOB et Tactons. Certains utilisateurs ont eu du mal à retenir les représentations : ils ont reconnu le bon code mais n'ont pas donné la bonne réponse. En complément nous soulignons que la moitié des utilisateurs préfèrent utiliser les PICOB semi-guidés alors que l'autre moitié préfère utiliser les Tactons pour reconnaître les composants.

Pour conclure l'architecture que nous avons développée a été utilisée avec succès. En plus de notre application d'exploration de schémas de circuits électriques elle a été utilisée par une nouvelle version de l'application d'exploration du système solaire développée à l'université de Tampere [153]. L'application d'exploration de schémas de circuits électriques a un potentiel pour aider des enseignants à expliquer des schémas de circuits électriques. Plusieurs améliorations de cette application sont envisageables. Nous pouvons ajouter des informations sonores, que ce soit une synthèse vocale ou des sons. Ces informations peuvent être utiles pour les composants, en aidant à retenir les représentations haptiques. Il serait intéressant d'ajouter des étiquettes aux nœuds afin d'aider les utilisateurs à se repérer dans le circuit. Ces étiquettes pourraient utiliser différentes techniques d'interaction : Braille, Tactons, synthèse vocale, etc. Ainsi les utilisateurs pourraient naviguer plus facilement dans les schémas. Enfin il serait intéressant d'observer l'utilisation de ce logiciel en classe. L'apprentissage des codes pourrait permettre de réduire les erreurs de reconnaissance. De plus l'utilisation de l'application avec un vrai enseignant permettrait à un utilisateur de mieux appréhender les circuits car l'enseignant y apporterait un sens, ce qui peut faciliter la compréhension.

Conclusion

> La bataille contre l'ignorance se gagne tous les jours, et elle finit par ouvrir sur des perspectives insoupçonnées.
>
> Tenzin Gyatso

Dans cette thèse nous avons étudié des techniques d'interaction permettant de coder des informations en utilisant le retour haptique. Le premier objectif visait à trouver une structure permettant de concevoir des icônes tactiles en utilisant une matrice de picots. Une telle structure a été définie par Brown et Brewster pour des icônes vibrotactiles appelées Tactons [26, 25, 30, 31, 32, 29]. Bien que plusieurs études ont été conduites sur des icônes tactiles utilisant des matrices de picots [152, 19, 90, 112, 118], aucune ne s'appuie sur une telle structure. Nous avons donc cherché des paramètres et des valeurs adéquats pour la création de telles icônes, et ainsi trouver une structure permettant de les caractériser.

Le deuxième objectif était de créer des icônes utilisant le retour de force pour transmettre des informations à l'utilisateur. Les études sur de telles icônes sont plus limitées [53, 102, 54], et n'utilisent qu'une dimension d'espace. Nous nous étions fixés comme but d'exploiter les possibilités d'un bras articulé à retour de force. Nous avons donc utilisé d'autres degrés de liberté. En ce qui concerne la structure, nous avons gardé l'idée des icônes tactiles définies précédemment.

Nous avions fixé comme troisième objectif d'utiliser les techniques d'interaction précédentes dans des applications multimodales. Pour cela nous avons créé une architecture multimodale et collaborative supportant ces techniques d'interaction. Cette architecture a été conçue en collaboration avec l'université de Tampere. Elle est une amélioration d'une architecture plus ancienne et plus rigide [153]. Nous y avons apporté des améliorations, et ajouté plusieurs agents permettant de gérer des périphériques comme le PHANToM ou la souris VTPlayer. Nous avons utilisé cette architecture pour développer une application d'exploration de schémas de circuits électriques. Cette application utilise les techniques d'interaction haptique développées précédemment pour aider des utilisateurs déficients visuels à explorer ces schémas.

Résumé

Le premier chapitre consitue un état de l'art partant de l'interaction en générale, et aboutissant à l'interaction haptique dans des systèmes pour déficients visuels. Dans les chapitres suivants nous avons conçu et utilisé des techniques d'interaction utilisant un retour haptique. Nous allons rappeler plus en détails les problèmes soulevés, les solutions et les résultats obtenus.

Chapitre I

Dans ce premier chapitre nous avons commencé par introduire la notion d'interface à travers les modèles d'interfaces. Les interfaces permettent de faire le lien entre l'utilisateur qui a une tâche à accomplir, et la machine qui doit l'accomplir . Nous avons évoqué deux types de modèles : les modèles à couches et les modèles à agents. Les modèles à couches fonctionnent par analyse lexicale, syntaxique et sémantique, alors que les modèles par agents fonctionnent à la manière des langages objet. Nous avons ensuite étudié les espaces de conception et de caractérisation pour les interactions multimodales. Ceci nous a permis d'avoir un aperçu de la problématique de la saisie et l'affichage d'informations à travers une interface. Certains de ces espaces adoptent le point de vue de l'utilisateur, d'autres adoptent le point de vue du système, et enfin une partie d'entre eux adoptent les deux points de vue.

Par la suite nous nous sommes concentrés sur les modalités qui nous intéressent, c'est-à-dire les modalités d'haptique en sortie. En premier lieu nous avons abordé l'haptique du point de vue somesthésique. Après avoir étudié le point de vue humain, nous avons étudié le point de vue système. En effet au niveau du système, le rendu haptique est délivré par des périphériques dits haptiques. Nous avons présenté quelques catégories de périphériques haptiques, dont des périphériques tactiles et des périphériques à retour de force. Enfin nous avons étudié des systèmes utilisant le retour haptique.

Pour finir nous nous sommes intéressés aux systèmes haptiques pour personnes ayant une déficience visuelle. À titre de préambule nous avons évoqué des études présentant des recommandations sur la manière de présenter des informations à des utilisateurs déficients visuels à l'aide d'un bras articulés ou de matrices de picots. Par la ensuite nous avons observé des études sur l'exploration de manière non visuelle. Nous avons étudié l'exploration d'objets en trois dimensions à l'aide d'un bras articulé, puis l'exploration de différents types de graphiques, l'exploration d'interfaces et enfin l'exploration de scènes 3D. Enfin nous avons évoqué quelques systèmes utilisant l'haptique dans l'enseignement.

Chapitre II

Dans ce deuxième chapitre, nous avons défini une structure pour des Tactons à matrices de picots. Nous définissons à la fois des Tactons statiques et des Tactons dynamiques.

Nous avons commencé par concevoir des Tactons mono-dimensionnels représentant des directions, par développement incrémental au cours de 5 expériences. Nous avons classé ces Tactons dans trois catégories : les Tactons statiques, les Tactons en vagues et les Tactons clignotants, les deux dernières étant des sous-ensembles des Tactons dynamiques. Le but de ces expériences était à la fois de trouver des ensembles de Tactons satisfaisants pour ces trois catégories, et de trouver des paramètres optimaux pour la conception des Tactons de chaque catégorie. Nous avons rapidement obtenu deux ensembles de Tactons statiques satisfaisant. Par contre, nous avons eu plus de difficulté à trouver des Tactons dynamiques satisfaisants pour tous les utilisateurs. Nous avons découvert que différencier les Tactons dynamiques radiaux et les Tactons dynamiques diagonaux permet de mieux les distinguer, et ainsi nous avons obtenu des Tactons dynamiques satisfaisants. Lors d'une expérience en conditions réelles, nous avons pu vérifier que les utilisateurs ont plus de difficulté en conditions réelles qu'en laboratoire. Enfin nous avons expérimenté avec succès une technique d'amélioration des Tactons clignotants, basée sur un point de repère.

L'expérimentation suivante portait sur des Tactons multi-dimensionnels. Nous avons décidé d'utiliser plusieurs paramètres pour construire des Tactons clignotants, exploitant 3 dimensions. Les résultats de ces expériences ont été analysées selon trois facteurs : le taux d'erreurs, le temps de réponse et la quantité d'information transmise. La quantité d'information transmise permet de mesurer l'apport de l'ajout d'informations supplémentaires par Tacton. L'analyse par taux d'erreurs n'aboutit pas aux mêmes conclusions que l'analyse par information transmise. Si on n'observe que le taux d'erreurs, l'utilisation d'une troisième valeur de vitesse provoque significativement plus d'erreurs. Il serait judicieux au vu de ces résultats de n'utiliser que deux valeurs de vitesse. Cependant l'analyse par information transmise montre que les utilisateurs ont reçu significativement plus d'informations avec les Tactons à trois vitesses qu'avec ceux à deux vitesses. Cette constatation laisserait conclure que l'utilisation de trois valeurs de vitesse est plus intéressante. Au final il s'agit d'un compromis à faire lors de l'utilisation de ces Tactons dans un système, selon que l'on ait besoin de plus d'informations à transmettre ou si l'on veut garantir un taux de reconnaissance élevé.

Ensuite nous avons utilisé des Tactons multi-dimensionnels dans un système de guidage destiné à aider des utilisateurs à explorer des formes géométriques de manière non visuelle. Nous avons pour cela utilisé des Tactons à deux dimensions. Afin de mesurer l'efficacité de cette méthode, nous avons comparé l'exploration de formes géométriques à l'aide de cette techniques, et à l'aide d'une méthode plus traditionnelle consistant à traduire les pixels de la forme en picots levés. Nous avons aussi comparé deux modes d'exploration de la forme : soit à l'aide d'une tablette graphique, soit à l'aide de la souris VTPlayer directement. Les résultats nous ont montré que la souris VTPlayer était plus pratique à utiliser que la tablette graphique pour explorer les formes. Par contre, les résultats ne permettent pas de trancher entre les deux méthodes de présentation. Nous avons donc proposé des améliorations à notre technique afin d'essayer d'obtenir de meilleurs résultats.

Pour terminer nous avons établi une analogie entre la structure des Tactons et la notation musicale. Le but de cette analogie est de permettre de trouver des paramètres communs, et à l'avenir de tester

la correspondance entre ces paramètres afin de concevoir des icônes crossmodales. L'intérêt de ces icônes est de pouvoir afficher la même information de la même manière, mais avec des modalités différentes. Ainsi l'utilisateur n'a qu'un seul langage à connaître pour l'interprétation des messages. Ces travaux sont encore à l'état embryonnaires et feront l'objet de recherches futures.

Chapitre III

Le sujet de ce troisième chapitre était la conception et l'évaluation d'une technique d'interaction permettant d'utiliser le retour de force pour coder de l'information. L'information peut ainsi être codée par une impulsion, ou une séquence d'impulsions appelée PICOB. Nous avons créé deux modes de lecture : guidée si la séquence d'impulsions est totalement contrôlée par le système, semi-guidée si elle est partiellement contrainte par le système mais que l'utilisateur garde un contrôle sur la lecture. Nous avons mené trois séries d'expériences : une sur les impulsions guidées, une sur les impulsions semi-guidées et enfin une sur les séquences d'impulsions semi-guidées.

L'étude sur les impulsions guidées avait pour objectif de trouver des paramètres et des valeurs adéquates pour la conception de cette technique d'interaction. Nous avons choisi de fixer la longueur de ces impulsions et de ne faire varier que la direction et l'amplitude. Les résultats ont montré que les utilisateurs n'ont eu aucune difficulté à déterminer la direction, que ce soit lorsqu'elle était présentée seule ou avec plusieurs amplitudes possibles. Par contre la distinction d'amplitude était plus problématique. Malgré les améliorations que nous avons essayées, il semble que les utilisateurs ont eu du mal à distinguer trois valeurs d'amplitude. Ils arrivaient souvent à reconnaître la plus petite amplitude, mais ont eu beaucoup d'hésitations avec les deux plus grandes. De plus l'analyse de la quantité d'information transmise révèle que les utilisateurs ne perçoivent pas plus d'informations avec trois valeurs. Nous recommandons donc de n'utiliser que deux valeurs d'amplitude au maximum.

La seconde expérimentation portait sur les impulsions semi-guidées. Alors que les impulsions guidées variaient selon le temps, ces impulsions varient selon la position du curseur sur un des axes. Les résultats montrent que les utilisateurs arrivent aussi bien à distinguer 4 directions que 2. De plus nous avons pu vérifier que les utilisateurs ont capté deux fois plus d'informations avec 4 directions possibles qu'avec seulement 2 directions. Le passage à 4 directions n'induit donc pas de perte d'informations, et donc il est est tout à fait possible d'utiliser 4 valeurs de directions dans un système de codage utilisant cette technique d'interaction. Les utilisateurs parviennent toujours à distinguer les direction lorsque plusieurs amplitudes sont proposées. Par contre, ils ont eu plus de difficultés à distinguer 3 amplitudes qu'à distinguer 2 amplitudes. De plus la différence d'information transmise n'est pas significative entre ces deux conditions. Nous concluons donc qu'il est inutile d'utiliser 3 valeurs d'amplitude.

En guise de troisième expérience nous avons étudié la reconnaissance de séquences d'impulsions. Les séquences se construisent par concaténation des impulsions dans le sens de la lecture. Il est possible d'espacer les impulsions afin de bien les distinguer. Les tests menés dans cette expériences ont

été conduits avec des utilisateurs déficients visuels. Cela nous a permis de recueillir le sentiment de ces utilisateurs sur cette technique d'interaction. Les résultats montrent que les utilisateurs parviennent à reconnaître les séquences, mais avec des hésitations. La tâche s'est avérée plus difficile lorsque l'orientation différait. Les utilisateurs ont parcouru les séquences plus de fois, et ont fait plus d'erreurs. Nous notons enfin qu'il y a des problèmes d'ambiguïtés entre certaines séquences, telles que *haut − haut* et *haut − bas − haut*. Dans cet exemple l'impulsion vers le bas entre les deux impulsions vers le haut peut être interprétée comme un prolongement du retour du premier haut et de l'aller du second. Nous proposons d'espacer les impulsions pour remédier à ce problème.

Chapitre IV

Dans le quatrième et dernier chapitre, nous avons détaillé la conception d'une architecture multimodale et collaborative, puis la conception et l'évaluation d'une application d'exploration de schémas de circuits électriques utilisant cette architecture.

L'architecture est basée sur un système multi-agents communiquant par le biais d'un bus logiciel. Le bus permet aux agents d'envoyer des informations sur le bus, et de réagir lorsque des informations les intéressant transitent sur le bus. Pour réagir à des messages transitant sur le bus, les agents spécifient les motifs de messages les intéressant et la fonction à appeler lorsqu'un tel message apparaît sur le bus. Nous avons conçu un framework multimodal permettant d'uniformiser la gestion des périphériques d'entrée et de sortie. Les agents sender sont responsable de l'acquisition des données venant des périphériques d'entrée et de les transmettre sur le bus selon une syntaxe normalisée. Les agents input sont des agents abstraits servant de patrons à la conception d'agents qui utilisent les données venant des périphérique d'entrée. Enfin les agents output sont responsable du rendu sur les périphériques de sortie. Les messages à leur envoyer sont aussi normalisés. Ainsi nous avons implémenté des agents pour le PHANToM, la souris VTPlayer, le clavier et la souris ordinaire, et pour le son.

Nous avons conçu une application permettant de naviguer dans un schéma de circuit électrique grâce à un guidage et des codages haptiques. Nous avons décrit les différentes phases de conception de cette application. Le premier prototype aimantait le circuit afin que l'utilisateur soit attiré dessus lorsqu'il passe à proximité. Les composants étaient codées par des séquences d'impulsions. Il s'est avéré que l'aimantation n'était pas suffisante et qu'il était donc difficile de parcourir les schémas de manière non-visuelle. Nous avons donc amélioré ce prototype en contraignant le stylet du bras articulé à l'intérieur du schéma. Nous avons aussi ajouté des informations de guidage aux intersections des fils et des composants afin d'aider l'utilisateur à trouver son chemin au sein du schéma. Le point critique selon certains utilisateurs, et particulièrement selon les utilisateurs aveugles, est que les aides placées aux intersections étaient jouées automatiquement, ce qui a perturbé leur navigation. Nous avons donc supprimé ces aides automatiques dans l'application finale, et n'avons laissé que l'aide à la demande de l'utilisateur. Cette application finale a été développée en utilisant l'architecture. L'exploration peut se faire désormais soit à l'aide d'un PHANToM soit à l'aide de la souris. En effet la souris peut être utilisée par un second utilisateur afin de guider le premier sur le circuit. Les

composants sont maintenant aussi représentés par des Tactons statiques. En effet il suffit d'ajouter un agent à l'application pour ajouter une représentation d'une information. Il est envisageable d'ajouter facilement une représentation par synthèse vocale ou par icône sonore. Cette application finale a été testée avec des utilisateurs déficients visuels. Les résultats montrent que les utilisateurs ayant le plus de difficultés à reconnaître la forme des circuits sont les utilisateurs aveugles. Par contre la plupart des utilisateurs ayant une vision résiduelle n'ont fait aucune erreur. Au niveau de la reconnaissance des composants, peu d'utilisateurs ont fait des erreurs, et généralement ils ont appris à corriger leurs erreurs au cours de l'expérience grâce à l'aspect collaboratif. La moitié des utilisateurs a préféré utiliser les informations tactiles alors que l'autre moitié a préféré utiliser les informations kinesthésiques. Nous en concluons que l'intégration de plusieurs techniques d'interaction est une bonne idée.

Résultats

Dans l'introduction de cette thèse nous avions énoncé trois objectifs, basés sur la création et l'utilisation de nouvelles techniques d'interaction. Ils ont été atteints grâce au travail présenté dans les chapitres précédents. Le premier objectif était de concevoir des icônes tactiles à l'aide d'une matrice de picots. La conception de ces icônes a été présentée dans le chapitre II. Elles possèdent une structure hiérarchique, et nous avons trouvé des paramètres adéquats permettant de coder plusieurs informations sur la même icône. Le deuxième objectif était de créer des icônes utilisant le retour de force. Nous avons décrit la conception et l'évaluation de telles icônes dans le chapitre III. Elles possèdent une structure similaire à celle des icônes tactiles. À l'instar de ces dernières, nous avons identifié des paramètres appropriés qui nous ont permis de coder plusieurs informations par icône. Enfin le troisième objectif était d'utiliser ces deux techniques d'interaction dans des applications multimodales. Pour parvenir à ce résultat, nous avons conçu une architecture multimodale, et l'avons utilisée pour construire une application d'exploration de schémas de circuits électriques. Cette application utilise entre autres les deux techniques d'interaction mentionnées ci-dessus. Les icônes tactiles ont aussi été utilisées dans une application d'exploration de formes géométriques (chapitre II), ainsi que dans une application d'exploration de labyrinthe [45].

Perspectives

La recherche possède la particularité que plus on progresse, plus il reste de travail à faire. Nous allons donc finir ce manuscrit en évoquant un condensé d'idées d'améliorations et de prolongements qui ont émergé au fil du temps.

Icônes tactiles

Nous avons défini les icônes tactiles de telle manière que chaque picot possède deux états : soit *levé* soit *baissé*. Cette contrainte a été dictée par les limitations des périphériques courants actuels. Cependant comme nous l'avons suggéré au moment où nous avons défini les Tactons, il serait intéressant d'étendre nos définitions afin de prendre en compte la hauteur du picot en définissant une valeur *min* et une valeur *max*. Les Tactons tels que nous les avons définis dans le chapitre II

formeraient un cas particulier où $min = 0$, $max = 1$ et la hauteur ne prendrait pas de valeurs dans $]0, 1[$.

Les Tactons multi-dimensionnels que nous avons conçus sont des Tactons clignotants utilisent comme dimension la direction, la taille et la vitesse. Nous songeons dans un premier temps à explorer d'autres paramètres tels que le rythme et la position spatiale. Dans un second temps, nous pensons aussi créer des Tactons multi-dimentionnels en utilisant d'autres types de Tactons dynamiques, tels que les vagues.

Nous avons laissé une question en suspens lors de ces études : la différence de reconnaissance des Tactons selon la position spatiale. En effet nous avons utilisé les Tactons sur l'index de la main dominante et de la main non-dominante. Il serait intéressant de vérifier qu'il n'y a pas de différence de perception entre les deux mains, et entre les différents doigts. Des tests à ce sujet on été initiés, mais sont restés au stade d'étude préliminaire. Bien que les premières évaluation ne semblent pas montrer de différences, nous comptons finir ce travail.

En ce qui concerne l'exploration de formes géométriques, beaucoup de travail reste à faire. En effet il s'agit du travail le plus récent de ce manuscrit, et d'autres expériences sont en cours. Nous allons dans un premier temps effectuer des modifications sur la technique de guidage. En effet dans la version présentée, chaque segment a une épaisseur, et le système applique un guidage différent selon si l'utilisateur est sur le segment ou en dehors. Le problème se pose si l'utilisateur oscille entre ces deux zones. Ce problème est récurrent car le guidage en dehors du segment consiste à diriger l'utilisateur vers le segment. Or à peine la limite franchie l'utilisateur est guidé vers le point suivant, mais il reste proche de la limite. Afin d'éviter ces oscillations nous allons ajouter un système d'hystérésis. Avec ce système lorsque l'utilisateur sera en dehors du segment, il ne sera plus guidé vers le bord de l'épaisseur du segment mais vers le milieu. Ainsi il ne sera plus à cheval sur la limite. La seconde amélioration que nous allons effectuer consiste à ajouter une information pour indiquer à l'utilisateur qu'il a effectué un tour complet sur la forme. Ensuite nous réfléchirons à une manière de représenter plus que 8 directions, ainsi qu'une manière de représenter les courbes. Ainsi nous pourrions représenter un panel plus large de formes géométriques.

Au niveau du protocole d'évaluation, nous ajusterons la taille des formes selon le mode de présentation afin qu'elles paraissent de la même taille dans les deux cas. Nous envisageons aussi d'ajouter d'autres formes afin de réduire les réponses par élimination. Enfin dans le mode de présentation par traduction des pixels en picots levés, il serait intéressant de comparer l'exploration de formes pleines et de formes creuses.

Pour terminer sur ce sujet, nous avons défini une analogie musicale afin de tenter de construire des icônes crossmodales. Notre analogie définit un espace théorique où les paramètres peuvent être mis en correspondance. Cependant il faut d'une part explorer tous ces paramètres pour chaque technique d'interaction, puis tester si les utilisateurs arrivent à faire le lien entre les paramètres de nos Tactons, ceux des Tactons vibrotactiles [26] et les earcons [22]. Un travail similaire a déjà été effectué

par Hoggan et Brewster afin de créer des icônes crossmodales à partir de Tactons vibrotactiles et d'earcons [70].

Icônes kinesthésiques

Nous avons exploré deux paramètres des PICOB : la direction et l'amplitude. Il nous reste à explorer d'autres paramètres tels que la longueur et la forme de signal. L'exploration d'autres paramètres permettrait d'élargir la palette de PICOB utilisables.

Cette technique d'interaction est extensible à d'autres périphériques que le PHANToM. Il serait intéressant de comparer la reconnaissance de ces PICOB sur d'autres périphériques tels que le Falcon ou la souris Wingman Force Feedback. En effet l'effet ressenti dépend fortement de la prise en main du périphérique car la force délivrée par le périphérique dépend de la résistance de la main et du poignet.

Nous nous efforcerons à trouver d'autres domaines d'application de cette technique d'interaction. En effet l'intégration de ces informations dans notre application d'exploration de schémas de circuits électriques est une réussite. Cela nous pousse à trouver d'autre cas où cette technique peut être utilisée.

La structure des ces icônes est volontairement proche de celle des Tactons. Nous n'avons pas encore défini d'analogie permettant de faire un lien entre les deux techniques d'interaction, mais il serait intéressant de s'y attacher. Ce travail est d'autant plus intéressant que nous avons déjà utilisé ces deux techniques d'interaction de manière crossmodale dans l'application d'exploration de schémas de circuits électriques.

Applications multimodales

Il reste du travail au niveau de l'architecture multimodale. En effet dans son état actuel, le contrôle de l'application se fait à travers l'API Reachin et son scene graph. Nous prévoyons d'abstraire l'architecture afin de pouvoir se passer de cette API. En effet l'utilisation de cette API était motivée par sa capacité à gérer de nombreux périphériques haptiques tels que le PHANToM, et des périphériques de pointage spécifiques tels que le Magellan. Or il existe d'autres API pour ces périphériques. En rendant l'architecture indépendante de toute API, l'utilisateur pourrait choisir l'API qu'il souhaite.

Nous n'avons pas implémenté toutes les fonctionnalités que nous avons souhaitées dans cette architecture. Tout d'abord il n'y a pas encore le support pour plusieurs PHANToM. Ensuite des agents gérant la synthèse vocale font partie d'autres applications utilisant l'architecture. Ils devraient être déplacés dans celle-ci afin de proposer leurs services à d'autres applications. Un agent gérant les tablettes braille était prévu mais n'a pas été implémenté. Cet agent pourrait être utile à de nombreuses applications. Il nous reste à inclure le support pour d'autres techniques d'interaction telles que les earcons et les Tactons vibrotactiles. Enfin, une fois l'API Reachin séparée de l'architecture, il faudra proposer des agents de rendu visuel permettant d'afficher des informations et des objets grâce au bus.

L'application d'exploration de circuits électriques est encore améliorable sur certains points. Dans un premier temps si nous extrayons l'API Reachin de l'architecture, nous devrons créer un agent se chargeant du rendu visuel. Ensuite nous pourrons créer un agent de synthèse vocale pour suppléer les indications tactiles. Cet agent pourra servir dans un premier temps à apprendre les indications tactiles. De manière générale il serait intéressant de supporter d'autres composants, dont des composants possédant plus de trois bornes, comme le transistor. Plusieurs utilisateurs auraient souhaité que le circuit soit modifiable dynamiquement dans l'application. Le problème est de pouvoir réaliser cette opération de manière non visuelle. Nous pourrions concevoir l'espace de travail comme une grille aimantée afin d'aider l'utilisateur à créer des circuits facilement. Nous pourrions ajouter une vibration en mode édition afin d'indiquer que le circuit est fermé. Pour finir, nous souhaitons ajouter le support pour un second PHANToM afin de renforcer l'aspect collaboratif de l'application.

Bibliographie

[1] *BraMaNet.* `http://handy.univ-lyon1.fr/projets/bramanet/`.

[2] *Reachin API 4, Reachin Technologies AB.* `http://www.reachin.se`.

[3] The design space of interfaces. multimedia systems, interaction and applications. *Dans Proceedings of the 1st Eurographics Workshop*, Stockholm, Sweden, avril 1991. Springer-Verlag GmbH.

[4] A metamodel for the runtime architecture of an interactive system : the uims tool developers workshop. *SIGCHI Bull.*, 24(1):32–37, 1992.

[5] Motoyuki AKAMATSU et Sigeru SATO : A multi-modal mouse with tactile and force feedback. *International Journal of Human-Computer Studies*, 40(3):443–453, 1994.

[6] Yusr AMAMOU et John STEWART : Analyse comparative de trajectoires d'exploration perceptive. *Dans RJC IHM 06*, Anglet, France, 2006.

[7] Yusr AMAMOU et John STEWART : Perceptive strategies with an enactive interface. *Dans Enactive /06*, pages 101–102, Montpellier, France, 2006.

[8] Dominique ARCHAMBAULT, Bernhard STÖGER, Mario BATUSIC, Claudia FAHRENGRUBER et Klaus MIESENBERGER : A software model to support collaborative mathematical work between braille and sighted users. *Dans Assets '07 : Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 115–122, Tempe, AZ, USA, 2007. ACM Press.

[9] Paul Bach-y RITA, Kurt KACZMAREK, Mitchell E. TYLER et Jorge GARCIA-LARA : Form perception with a 49-point electrotactile stimulus array on the tongue : a technical note. *Journal of Rehabilitation Research Development*, 35(4):427–430, 1998.

[10] Paul Bach-y RITA, Mitchell E. TYLER et Kurt KACZMAREK : Seeing with the Brain. *International Journal of Human-Computer Interaction*, 15(2):285–295, 2003.

[11] Sarah BAILLIE, Andrew CROSSAN, Stephen A. BREWSTER, Dominic MELLOR et Stuart REID : Validation of a Bovine Rectal Palpation Simulator for Training Veterinary Students. *Studies in Health Technology and Informatics*, 111:33–36, 2005.

[12] Gilles BAILLY, Laurence NIGAY et David AUBER : 2M : un espace de conception pour i'interaction bi-manuelle. *Dans UbiMob '05 : Proceedings of the 2nd French-speaking conference on Mobility and ubiquity computing*, pages 177–184. ACM Press, 2005.

[13] BASS audio library. *BASS audio library.* `http://www.un4seen.com`.

[14] Benjamin BAYART, Aurélien POCHEVILLE et Abderrahmane KHEDDAR : An adaptive haptic guidance software module for i-touch : example through an handwriting teaching simulation and a 3d maze. *Dans HAVE 2005, Haptic Audio Visual Environments and their Applications*, Ottawa, Canada, octobre 2005. IEEE Computer Society Press.

[15] Fabio BELLIFEMINE, Agostino POGGI et Giovanni RIMASSA : Jade - a fipa-compliant agent framework. *Dans Proceedings of the Practical Applications of Intelligent Agents*, pages 97–108, London, England, avril 1999.

[16] Mohamed BENALI-KHOUDJA et Moustapha HAFEZ : VITAL : A VIbroTActiLe Interface with Thermal Feedback. *Dans Journée Scientifique Internationale IRCICA*, pages 133–144, Lille, France, mars 2004.

[17] Mohamed BENALI-KHOUDJA, Moustapha HAFEZ, Jean-Marc ALEXANDRE et Abderrahmane KHEDDAR : Tactile interfaces : a state-of-the-art survey. *Dans International Symposium on Robotics*, Paris, France, mars 2004.

[18] Mohamed BENALI-KHOUDJA, Moustapha HAFEZ et Abderrahmane KHEDDAR : Vital : An electromagnetic integrated tactile display. *Displays*, 28(3):133–144, 2007.

[19] Mohamed BENALI-KHOUDJA, Moustapha HAFEZ et Amaury and SAUTOUR : Towards a new tactile language to communicate emotions. *Dans Proceedings of the International Conference on Mecatronics & Automation*, Niagara Falls, Canada, juillet 2005.

[20] Niels Ole BERNSEN : Modality theory : supporting multimodal interface design. *Dans ERCIM Workshop on Multimodal Human-Computer Interaction*, pages 37–44, 1993.

[21] Joseph P. BIGUS, Don A. SCHLOSNAGLE, Jeff R. PILGRIM, W. Nathaniel MILLS et Yixin DIAO : ABLE : A toolkit for building multiagent autonomic systems. *IBM Systems Journal*, 41(3):350–371, 2002.

[22] Meera M. BLATTNER, Denise A. SUMIKAWA et Robert M. GREENBERG : Earcons and Icons : Their Structure and Common Design Principles. *Human-Computer Interaction*, 4(1):11–44, 1989.

[23] Richard BOLT : Put-That-There : Voice and Gesture at the Graphics Interface. *Dans Computer Graphics Proceedings, SIGGRAPH 1980*, pages 262–270. ACM Press, 1980.

[24] Christoph W. BORST et Vijay B. BAIYYA : Enhancing VR-based Visualization with a 2D Vibrotactile Array. *Dans 2007 ACM symposium on Virtual Reality Software and Technology*, pages 183–186. ACM Press, 2007.

[25] Stephen A. BREWSTER et Lorna M. BROWN : Non-visual information display using tactons. *Dans CHI '04 : Extended abstracts on Human factors in computing systems*, pages 787–788, Vienna, Austria, avril 2004. ACM Press.

[26] Stephen A. BREWSTER et Lorna M. BROWN : Tactons : structured tactile messages for non-visual information display. *Dans CRPIT '28 : Proceedings of the fifth conference on Australasian user interface*, pages 15–23, Dunedin, New Zealand, 2004. Australian Computer Society, Inc.

[27] Stephen A. BREWSTER, Faraz CHOHAN et Lorna M. BROWN : Tactile feedback for mobile interactions. *Dans CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–162, San Jose, California, USA, avril 2007. ACM Press.

[28] Stephen A. BREWSTER, Peter C. WRIGHT et Alistair EDWARDS : An evaluation of earcons for use in auditory human-computer interfaces. *Dans CHI '93 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 222–227, Amsterdam, The Netherlands, avril 1993. ACM Press.

[29] Lorna M. BROWN : *Tactons : Structured Vibrotactile Messages for Non-Visual Information Display*. Thèse de doctorat, University of Glasgow, Glasgow, Scotland, 2007.

[30] Lorna M. BROWN, Stephen A. BREWSTER et Helen C. PURCHAISE : A first investigation into the effectiveness of tactons. *Dans Proceedings of the WorldHaptics 2005 Conference*, pages 167–176, Pisa, Italy, mars 2005. IEEE Computer Society Press.

[31] Lorna M. BROWN, Stephen A. BREWSTER et Helen C. PURCHAISE : Multidimensional tactons for non-visual information presentation in mobile devices. *Dans MobileHCI '06 : Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 231–238, Helsinki, Finland, 2006. ACM Press.

[32] Lorna M. BROWN, Stephen A. BREWSTER et Helen C. PURCHAISE : Tactile crescendos and sforzandos : applying musical techniques to tactile icon design. *Dans CHI '06 : Extended abstracts on Human factors in computing systems*, pages 610–615, Montréal, Québec, Canada, avril 2006. ACM Press.

[33] Marcellin BUISSON, Alexandre BUSTICO, Stéphane CHATTY, François-Régis COLIN, Yannick JESTIN, Sébastien MAURY, Christophe MERTZ et Philippe TRUILLET : Ivy : Un bus logiciel au service du développement de prototypes de systèmes interactifs. *Dans IHM 2002 : Proceedings of the 14th French-speaking conference of human-computer interaction*, Poitiers, France, novembre 2002. ACM Press.

[34] Jean CAELEN : 10 ans de recherche... en multimodalés. *Dans Colloque sur les Interfaces Multimodales*, IRIT, Toulouse, France, 2000.

[35] Gianni CAMPION, Qi WANG et Vincent HAYWARD : The Pantograph Mk-II : A Haptic Instrument. *Dans Proceedongs of IROS 2005, IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 723–728, Edmonton, Canada, août 2005.

[36] David CANFIELD-SMITH, Charles IRBY, Ralph KIMBALL, Bill VERPLANK et Eric HARSLEM : Designing the Star User Interface. *Byte Magazine*, 7(4):242–282, avril 1982.

[37] Angela CHANG, Sile O'MODHRAIN, Rob JACOB, Eric GUNTHER et Eric ISHII : Comtouch : design of a vibrotactile communication device. *Dans DIS '02 : Proceedings of the conference on Designing interactive systems*, pages 312–320, London, England, 2002. ACM Press.

[38] Joëlle COUTAZ : PAC, an Oriented Object Model for Dialog Design. *Dans Interact 1987*, pages 431–436, 1987.

[39] Joëlle COUTAZ et Laurence NIGAY : *Architecture logicielle conceptuelle des systèmes interactifs*, pages 207–246. 2001. Chapitre 7 Analyse et Conception de l'Interaction Homme-Machine dans les systèmes d'information, Kolski Ed., Hermes Publ.

[40] Joëlle COUTAZ, Laurence NIGAY et Daniel SALBER : The MSM framework : A Design Space for Multi-Sensori-Motor Systems. *Dans 3rd International Conference EWHCI'93, East/West Human Computer Interaction*, pages 431–436. Springer-Verlag GmbH, 1993.

[41] Joëlle COUTAZ, Laurence NIGAY et Daniel SALBER : *Agent-Based Architecture Modeling for Interactive Systems*, pages 191–209. Springer-Verlag GmbH, Critical Issues in User Interface Engineering, P. Palanque & D. Benyon édition, 1995.

[42] Joëlle COUTAZ, Laurence NIGAY, Daniel SALBER, Ann BLANDFORD, Jon MAY et Richard M. YOUNG : Four easy pieces for assessing the usability of multimodal interaction : the CARE properties. *Dans Interact 1995*, pages 115–120, 1995.

[43] Andrew CROSSAN : *The Design and Evaluation of a Haptic Veterinary Training Simulator*. Thèse de doctorat, University of Glasgow, Glasgow, Scotland, 2003.

[44] Andrew CROSSAN et Stephen A. BREWSTER : MICOLE — Inclusive Interaction for Data Creation, Visualisation and Collaboration. *Dans CHI '05 : Proceedings of the SIGCHI conference on Human factors in computing systems*, Portland, USA, 2005. ACM Press.

[45] Andrew CROSSAN et Stephen A. BREWSTER : Two-handed navigation in a haptic virtual environment. *Dans CHI '06 : Proceedings of the SIGCHI conference on Human factors in computing systems*, Montréal, Québec, Canada, avril 2006. ACM Press.

[46] Andrew CROSSAN, John WILLIAMSON et Stephen A. BREWSTER : A General Purpose control-Based Trajectory Playback for Force-Feedback System. *Dans Proceeding of the 6th International Conference EuroHaptics 2006*, Paris, France, juillet 2006.

[47] Adolphe DANHAUSER : *Théorie de la musique*. Éditions Henri Lemoine, 1929.

[48] Gintautas DAUNYS et Vidas LAURUSKA : Théorie de la musique. *Dans Proceedings of the 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007 Held as Part of HCI International 2007*, pages 349–352, Beijing, China, juillet 2007. Springer-Verlag GmbH.

[49] Jack Tigh DENNERLEIN, David B. MARTIN et Christopher HASSER : Force-feedback improves performance for steering and combined steering-targeting tasks. *Dans CHI '00 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 423–429, The Hague, The Netherlands, avril 2000. ACM Press.

[50] Thierry DUVAL et Laurence NIGAY : Implémentation d'une application de simulation selon le modèle PAC-Amodeus. *Dans IHM'99 : Proceedings of the 11th French-speaking conference of human-computer interaction*, pages 86–93, Montpellier, France, novembre 1999. Cépaduès-Editions.

[51] Alistair EDWARDS, Heather MCCARTNEY et Flavio FOGAROLO : Lambda : a multimodal approach to making mathematics accessible to blind students. *Dans Assets '06 : Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 48–54, Portland, OR, USA, octobre 2006.

[52] Benoît ENCELLE, Nadine JESSEL, Josiane MOTHE et Bachelin RALALASON : BMML : Braille Music Markup Language. *Dans International Conference on Internet Computing (ICOMP)*, Las Vegas, USA, juillet 2008.

[53] Mario ENRIQUEZ et Karon E. MACLEAN : The hapticon editor : A tool in support of haptic communication research. *Dans Proceedings of the 11th International Symposium on Haptic Interfaces*

for Virtual Environment and Teleoperator Systems (HAPTICS 2003), pages 356–362, Los Angeles, CA, USA, mars 2003. IEEE Computer Society Press.

[54] Mario ENRIQUEZ et Karon E. MACLEAN : Impact of haptic warning signal reliability in a time-and-safety-critical task. *Dans Proceedings of the 12th Annual Symposium on Haptic Interfaces for Virtual Environments and Tele-operator Systems*, pages 407–414, Chicago, IL, USA, 2004.

[55] Yvonne ERIKSSON : How to make tactile pictures understandable to the blind reader. *Dans Proceedings of the 65th IFLA Council and General Conference*, Bangkok, Thailand, août 1999.

[56] Grigori EVREINOV, Tatiana EVREINOVA et Roope RAISAMO : Manipulating vibro-tactile sequences on mobile pc. *Dans* Jörg Roth RÉMI BASTIDE, Philippe Palanque, éditeur : *Joint Working Conferences EHCI-DSVIS 2004, Revised Selected Papers, LNCS 3425*, pages 245–252, Hamburg, Germany, juillet 2004. Springer-Verlag GmbH.

[57] Grigori EVREINOV, Tatiana EVREINOVA et Roope RAISAMO : Mobile games for training tactile perception. *Dans* Matthias RAUTERBERG, éditeur : *Entertainment Computing - ICEC 2004, Third International Conference, LNCS 3166*, pages 468–475, Eindhoven, The Netherlands, septembre 2004. Springer-Verlag GmbH.

[58] FIPA, Fondation for Intelligent Physical Agents. *Specifications, 1997*. `http://www.fipa.org`.

[59] Force Dimension. *Force Dimension*. `http://www.forcedimension.com`.

[60] Masaaki FUKUMOTO et Toshiaki SUGIMURA : Active click : tactile feedback for touch panels. *Dans CHI '01 : Extended abstracts on Human factors in computing systems*, pages 121–122, Seattle, Washington, avril 2001. ACM Press.

[61] Blaithin GALLAGHER et Walter FRASCH : Tactile Acoustic Computer Interaction System : A new type of Graphic Access for the Blind. *Dans Technology for Inclusive Design and Equality Inproving the Quality of Life for the European Citizen, Proceedings of the 3rd TIDE Congress*, Helsinki, Finland, juin 1998.

[62] Philippe GARREC, Jean-Pierre FRICONNEAU et François LOUVEAU : Virtuose 6D : a new force-control master arm using innovative ball-screw actuators. *Dans 35th International Symposium on Robotics*, Paris-Nord-Villepinte, France, mars 2004.

[63] William W. GAVER : Synthesizing auditory icons. *Dans CHI '93 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 228–235, Amsterdam, The Netherlands, avril 1993. ACM Press.

[64] E. Bruce GOLDSTEIN : *Sensation and Perception*. Wadsworth Publishing, 2006.

[65] Florian GOSSELIN, Alain RIWAN et F LAUTURE : Virtuose 3d : a new input device for virtual reality. *Dans Proc. of the Virtual Reality International Conference*, LAval, France, 2001.

[66] Mark GREEN : Report on dialogue specification tools. *User Interface Management Systems, G. Pfaff, ed.*, pages 9–20, 1985.

[67] Sylvain HANNETON, Charles LENAY, Olivier GAPENNE, Sébastien VERMANDEL et Catherine MARQUE : Dynamique de la reconnaissance de caractères via une interface haptique. *Dans Actes du colloque ARC'98*, pages 343–347, Paris Saint Denis, France, 1998.

[68] Vincent HAYWARD : Transduction tactile par champs de contraintes latérales réparties : les dispositifs STReSS. *Dans Journée Conception de stimulateurs tactiles orientée utilisateur*, Lille, France, mars 2004.

[69] Vincent HAYWARD, Oliver R. ASTLEY, Juan Manuel CRUZ-HERNANDEZ, Danny GRANT et Gabriel ROBLES DE LA TORRE : Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.

[70] Eve HOGGAN et Stephen A. BREWSTER : Crossmodal Icons for Information Display. *Dans CHI '06 : Extended abstracts on Human factors in computing systems*, pages 857–862, Montréal, Québec, Canada, avril 2006. ACM Press.

[71] Eve HOGGAN et Stephen A. BREWSTER : Crossmodal spatial location : initial experiments. *Dans 4th Nordic Conference on Human-Computer Interaction*, pages 469–472, Oslo, Norway, octobre 2006. ACM Press.

[72] Eve HOGGAN et Stephen A. BREWSTER : Designing audio and tactile crossmodal icons for mobile devices. *Dans ICMI '07 : Proceedings of the 9th annual International Conference on Multimodal Interfaces*, pages 162–169, Nagoya, Japan, novembre 2007. ACM Press.

[73] Eve HOGGAN et Stephen A. BREWSTER : New parameters for tacton design. *Dans CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2417–2422, San Jose, California, USA, avril 2007. ACM Press.

[74] Eve HOGGAN, Stephen A. BREWSTER et Jody JOHNSTON : Investigating the effectiveness of tactile feedback for mobile touchscreens. *Dans CHI '08 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1573–1582, Florence, Italy, avril 2008. ACM Press.

[75] Mikro HORSTMANN, Cornelius HAGEN, Neil KING, Sijo DIJKSTRA, David CROMBIE, David Gareth EVANS, George T. IOANNIDIS, Paul BLENKHORN, Otthein HERZOG et Christoph SCHLIEDER : TeDUB : Automatic interpretation and presentation of technical diagrams for blind people. *Dans Conference and Workshop on Assistive Technologies for Vision and Hearing Impairment - CVHI'2004, EURO-ASSIST-VHI-2 : Accessibility, Mobility and Social Integration*, Granada, Spain, 2004.

[76] Immersion. *Immersion*. `http://www.immersion.com`.

[77] Christophe JACQUET : *Linux VTPlayer Driver (GPL)*. `http://vtplayer.sourceforge.net/`.

[78] Gunnar JANSSON : Can a haptic force feedback display provide visually impaired people with useful information about texture roughness and 3d form of virtual objects ? *Dans Proceedings of the 2nd European Conference Series on Disability, Virtual Reality and Associated Technologies*, pages 105–111, Skövde, Sweden, 1998.

[79] Gunnar JANSSON et Katarina BILLBERGER : The PHANToM used without visual guidance. *Dans The First PHANToM Users Research Symposium (PURS'99)*, pages 27–30, Heidelberg, Germany, May, Jon 1999.

[80] Gunnar JANSSON, Katarina BILLBERGER, Helen PETRIE, Chetz COLWELL, Diana KORNBROT, Jens FÄNGER, Henry KÖNIG, Andrew HARDWICK et Stephen FURNER : Haptic virtual envi-

ronments for blind people : exploratory experiments with two devices. *International Journal of Virtual Reality*, 4:10–20, 1999.

[81] Gunnar JANSSON, Jens FÄNGER, Henry KÖNIG et Katarina BILLBERGER : Visually impaired person's use of the phantom for information about texture and 3d form of virtual objects. *Dans Proceedings of the Third PHANToM User's Group Workshop*, pages 95–100, Cambridge, MA, USA, 1998.

[82] Gunnar JANSSON et Anna IVÅS : Can the efficiency of a haptic display be increased by short-time practice in exploration? *Dans Proceedings of the First International Workshop on Haptic Human-Computer Interaction*, pages 22–27. Springer-Verlag, 2001.

[83] Gunnar JANSSON et Karin LARSSON : Identification of haptic virtual objects with different degrees of complexity. *Dans Proceeding of the 2nd International Conference Eurohaptics 2002*, pages 57–60, Edinburgh, Scotland, juillet 2002. ACM Press.

[84] Gunnar JANSSON et Patrik PEDERSEN : Obtaining geographical information from a virtual map with a haptic mouse. *Dans International Cartographic Conference*, La Coruña, Spain, juillet 2005.

[85] Joel F. JENSEN et John HILL : Advanced Telepresence Surgery System Development. Rapport technique, Medical Technology Laboratory, SRI International, décembre 2005.

[86] Hiroyuki KAJIMOTO, Yonezo KANNO et Susumu TACHI : Forehead electro-tactile display for vision substitution. *Dans Proceeding of the 6th International Conference EuroHaptics 2006*, Paris, France, juillet 2006.

[87] Johan KILDAL et Stephen A. BREWSTER : Explore the matrix : Browsing numerical data tables using sound. *Dans Proceedings of the 11th International Conference on Auditory Display (ICAD2005)*, Limerick, Ireland, juillet 2005.

[88] Roberta L. KLATZKY et Susan J. LEDERMAN : The haptic glance : A route to rapid object identification and manipulation. *Attention and Performance XVII. Cognitive regulations of performance : Interaction of theory and application*, pages 165–196, 1999.

[89] Roberta L. KLATZKY et Susan J. LEDERMAN : *A.F. Healy & R. W. Proctor (Eds.), Experimental Psychology*, volume 4 de *Handbook of Psychology*, chapitre Touch, pages 147–176. John Wiley & Sons, New York, 2003.

[90] Eric LECOLINET et Gérard MOURET : Tactiball, tactipen, tactitab - ou comment toucher du doigt les donnéees de son ordinateur. *Dans IHM 2005 : Proceedings of the 17th French-speaking conference of human-computer interaction*, pages 227–230, Toulouse, France, septembre 2005. ACM Press.

[91] Anatole LECUYER : *Contribution à l'étude des retours haptiques et pseudo-haptiques et de leur impact sur les simulations d'opérations de montage/démontage en aéronautique.* Thèse de doctorat, Université Parix XI Orsay, Paris, France, 2001.

[92] Anatole LECUYER, Jean-Marie BURKHARDT et Laurent ETIENNE : Feeling Bumps and Holes without a Haptic Interface : the Perception of Pseudo-Haptic Textures. *Dans CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 239–246, Vienna, Austria, avril 2004. ACM Press.

[93] Anatole LECUYER, Pascal MOBUCHON, Christine MEGARD, Jérôme PERRET, Claude ANDRIOT et Jean-Pierre COLINOT : Homere : a multimodal system for visually impaired people to explore virtual environments. *Dans Proceedings of IEEE Virtual Reality*, 2003.

[94] Johnny Chung LEE, Paul H. DIETZ, Darren LEIGH, William S. YERAZUNIS et Scott E. HUDSCON : Haptic Pen : A Tactile Feedback Stylus For Touch Screens. *Dans* Steven FEINER et James A. LANDAY, éditeurs : *UIST '04 : Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 291–294, Santa Fe, New Mexico, USA, octobre 2004. ACM Press.

[95] Vincent LÉVESQUE : Blindness, technology and haptics. Rapport technique, McGill University, Montréal, Québec, Canada, 2005.

[96] Vincent LÉVESQUE et Vincent HAYWARD : Experimental Evidence of Lateral Skin Strain During Tactile Exploration. *Dans Proceeding of the 3rd International Conference Eurohaptics 2003*, pages 261–275, Dublin, UK, juillet 2003. ACM Press.

[97] Vincent LÉVESQUE, Jérôme PASQUERO, Vincent HAYWARD et Maryse LEGAULT : Display of virtual braille dots by lateral skin deformation : Feasibility study. *ACM Transactions on Applied Perception*, 2(2):132–149, 2005.

[98] LibUSB. *LibUSB*. `http://libusb.sourceforge.net/`.

[99] John LINVILL et Jim BLISS : A direct translation reading aid for the blind. *Proceedings of the IEEE*, 54(1):40–51, 1966.

[100] Logitech. *Logitech*. `http://www.logitech.com`.

[101] Jack M. LOOMIS et Susan J. LEDERMAN : *K. Boff, L. Kaufman, & J. Thomas (Eds.), Handbook of perception and human performance*, chapitre Tactual perception, pages 31–1–31–41. John Wiley & Sons, New York, 1986.

[102] Karon E. MACLEAN et Mario ENRIQUEZ : Perceptual design of haptic icons. *Dans Proceeding of the 3rd International Conference Eurohaptics 2003*, pages 351–363, Dublin, UK, juillet 2003. ACM Press.

[103] Charlotte MAGNUSSON, Henrik DANIELSSON et Kirsten RASSMUS-GRÖHN : Non visual haptic audio tools for virtual environments. *Dans Workshop on Haptic and Audio Interaction Design*, Glasgow, Scotland, septembre 2006.

[104] Charlotte MAGNUSSON, Håkan EFTRING, Kirsten RASSMUS-GRÖHN, Andrew CROSSAN, Stephen A. BREWSTER, Gunnar JANSSON, Imre JUHASZ, Apostolos STAMOU, Vidas LAURUSKA, Dominique ARCHAMBAULT, Eva-Lotta SALLNÄS, Arto HIPPULA, Roope RAISAMO, Erika TANUA-PIIROINEN, Thomas PIETRZAK, Fredrik WINBERG et Benoît MARTIN : Final user requirements and design recommendations. *Dans MICOLE deliverable D14*. 2007.

[105] Charlotte MAGNUSSON, Kirsten RASSMUS-GRÖHN et Håkan EFTRING : AHEAD - audio-haptic drawing editor and explorer for education. *Dans Proceedings of HAVE 2007 – IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pages 351–363, Ottawa, Canada, octobre 2007.

[106] Benoît MARTIN, Isabelle PECCI et Thomas PIETRZAK : Angle Recognition Cues using a new API dedicated to the VTPlayer Mouse. *Dans International Conference on Human-Machine iNteraction, HuMaN'07*, Timimoun, Algeria, mars 2007.

[107] Jean-Claude MARTIN : TYCOON : Theoretical Framework and Software Tools for Multimodal Interfaces. *Dans Intelligence and Multimodality in Multimedia interfaces*, pages 1–25. AAAI Press, 1998.

[108] Jean-Claude MARTIN : TYCOON : six primitive types of cooperation for observing, evaluating and specifying cooperations. *Dans Working notes of the AAAI Fall 1999 Symposium on Psychological Models of Communication in Collaborative Systems*, pages 1–25, North Falmouth, MA, USA, novembre 1999. AAAI Press.

[109] Jean-Claude MARTIN et Dominique BÉROULE : Types et Buts de Coopération entre Modalités. *Dans IHM'93 : 5èmes Journées sur l'Ingénierie des interfaces Homme-Machine*, pages 17–22, Lyon, France, octobre 1993. Cépaduès-Editions.

[110] Dominic MASSARO : Illusions and issues in bimodal speech perception. *Dans Auditory Visual Speech Perception '98*, pages 21–26, Terrigal-Sydney, Australia, décembre 1998.

[111] Thomas M. MASSIE et J. Kenneth SALISBURY : The phantom haptic interface : A device for probing virtual objects. *Dans Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 295–301, Chicago, novembre 1994. vol. 1.

[112] Christine MEGARD, Samuel ROSELIER et Jean-Marie BURKHARDT : Verbal association to tactile patterns : A step towards textured legends in multimodal maps. *Dans CHI '08 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 3579–3584, Florence, Italy, avril 2008. ACM Press.

[113] Micole. *European project MICOLE : Multimodal Collboration Environment for Inclusion of Visually Impaired Children - IST-2003-511592*. `http://micole.cs.uta.fi`.

[114] Microsoft. *Microsoft*. `http://www.microsoft.com`.

[115] George A. MILLER : The magical number seven, plus or minus two : Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

[116] Moog FCS. *Moog FCS*. `http://www.moog.com/`.

[117] Doublas B. MORAN, Adam J. CHEYER, Luc E. JULIA, David L. MARTIN et Sangkyu PARK : Multimodal User Interfaces in the Open Agent Architecture. *Dans Proc. of the 1997 International Conference on Intelligent User Interfaces (IUI97)*, pages 61–68, Orlando, FL, USA, janvier 1997. ACM Press.

[118] Tahir MUHAMMAD, Gilles BAILLY et Eric LECOLINET : Exploring the Impulsion and Vibration Effects of Tactile Patterns. *Dans British HCI 2008*, Liverpool, UK, septembre 2008.

[119] Andrew NASHEL et Sharif RAZZAQUE : Tactile virtual buttons for mobile devices. *Dans CHI '03 : Extended abstracts on Human factors in computing systems*, pages 854–855, Ft. Lauderdale, Florida, USA, avril 2003. ACM Press.

[120] Laurence NIGAY : *Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales*. Thèse de doctorat, Laboratoire de Génie Informatique (IMAG), Université Joseph Fourier, Grenoble, France, 1994.

[121] Laurence NIGAY : Design space for multimodal interaction. *Dans Proceedings of IFIP Congress'2004, Building the Information Society*, pages 403–408, Toulouse, France, 2004.

[122] Laurence NIGAY et Joëlle COUTAZ : A Design Space For Multimodal Systems : Concurrent Processing and Data Fusion. *Dans Proceedings of INTERCHI'93*, pages 172–178, Amsterdam, Netherlands, 1993. ACM Press.

[123] Laurence NIGAY et Joëlle COUTAZ : A Generic Platform for Addressing the Multimodal Challenge. *Dans CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 98–105, Denver, Colorado, United States, May, Jon 1995. ACM Press.

[124] Laurence NIGAY et Joëlle COUTAZ : Espaces conceptuels pour l'interaction multimédia et multimodale. *TSI, spécial Multimédia et Collecticiel, AFCET Hermes Publ., Vol 15(9)*, pages 1195–1225, 1996.

[125] Nintendo. *Nintendo*. `http://www.nintendo.com`.

[126] Nicolas NOBLE et Benoît MARTIN : Preliminary study of tactile feedback devices featuring quick discrimination. *Dans Enactive /05*, Genova, Italy, 2005. CD-ROM proceedings.

[127] Nicolas NOBLE et Benoît MARTIN : Shape discovering using tactile guidance. *Dans Proceeding of the 6th International Conference EuroHaptics 2006*, Paris, France, juillet 2006.

[128] Novint. *Novint*. `http://www.novint.com`.

[129] Ian OAKLEY, Marilyn Rose MCGEE, Stephen A. BREWSTER et Philip GRAY : Putting the feel in look and feel. *Dans CHI '00 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, The Hague, The Netherlands, avril 2000. ACM Press.

[130] Sile O'MODHRAIN et Brent GILLESPIE : The moose : A haptic user interface for blind persons. *Dans Proceedings of the World Wide Web Conference*, Santa Clara, CA, USA, 1997.

[131] Sharon OVIATT : Ten Myths of Multimodal Interaction. *Communications of THE ACM*, 42(11):74–81, 1999.

[132] Jérôme PASQUERO et Vincent HAYWARD : Stress : A practical tactile display system with one millimeter spatial resolution and 700hz fresh rate. *Dans Proceeding of the 3rd International Conference Eurohaptics 2003*, pages 94–110, Dublin, UK, juillet 2003. ACM Press.

[133] Perkins. *PErkins*. `http://www.perkins.org`.

[134] Grégory PETIT, Aude DUFRESNE, Vincent LÉVESQUE, Vincent HAYWARD et Nicole TRUDEAU : Graphisme tactile appliqué aux illustrations de manuels scolaires à l'usage d'enfants ayant une déficience visuelle. *Dans IHM 2008 : Proceedings of the 20th French-speaking conference of human-computer interaction*, Metz, France, septembre 2008. ACM Press.

[135] G. E. PFAFF, éditeur. *Workshop on User Interface Management Systems*, Seeheim, Germany, novembre 1985. Springer-Verlag GmbH.

[136] Thomas PIETRZAK, Andrew CROSSAN, Stephen A. BREWSTER, Benoît MARTIN et Isabelle PECCI : Creating usable pin array tactons for non-visual information. *IEEE Transactions on Haptics*, 2008. In revision.

[137] Thomas PIETRZAK, Benoît MARTIN et Isabelle PECCI : Affichage d'informations par des impulsions haptiques. *Dans IHM 2005 : Proceedings of the 17th French-speaking conference of human-computer interaction*, pages 223–226, Toulouse, France, septembre 2005. ACM Press.

[138] Thomas PIETRZAK, Benoît MARTIN et Isabelle PECCI : Information display by dragged haptic bumps. *Dans Enactive /05*, Genova, Italy, 2005. CD-ROM proceedings.

[139] Thomas PIETRZAK, Benoît MARTIN, Isabelle PECCI, Rami SAARINEN, Roope RAISAMO et Janne JÄRVI : The MICOLE Architecture : Multimodal Support for Inclusion of Visually Impaired Children. *Dans The Ninth International Conference on Multimodal Interfaces, ICMI2007*, Nagoya, Japan, novembre 2007.

[140] Thomas PIETRZAK, Nicolas NOBLE, Isabelle PECCI et Benoît MARTIN : Évaluation d'un logiciel d'exploration de circuits électriques pour déficients visuels. *Dans RJC IHM 06*, Anglet, France, 2006.

[141] Thomas PIETRZAK, Isabelle PECCI et Benoît MARTIN : Static and dynamic tactile directional cues experiments with vtplayer mouse. *Dans Proceeding of the 6th International Conference EuroHaptics 2006*, pages 63–68, Paris, France, juillet 2006.

[142] Thomas PIETRZAK, Isabelle PECCI et Benoît MARTIN : Un logiciel d'exploration de schémas de circuits électriques basé sur l'API MICOLE. *Dans IHM 2007 : Proceedings of the 19th French-speaking conference of human-computer interaction*, Paris, France, novembre 2007. ACM Press.

[143] Christophe RAMSTEIN : A multimodal user interface system with force feedback and physical models. *Dans Conference on Human-Computer Interaction INTERACT '95*, pages 157–162, Lillehammer, Norway, juin 1995.

[144] Christophe RAMSTEIN et Vincent HAYWARD : The pantograph : a large workspace haptic device for multimodal human computer interaction. *Dans CHI '94 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 57–58, Boston, Massachusetts, United States, avril 1994. ACM Press.

[145] Christophe RAMSTEIN, Odile MARTIAL, Aude DUFRESNE, Michel CARIGNAN, Patrick CHASSÉ et Philippe MABILLEAU : Touching and hearing GUI's : design issues for the PC-Access system. *Dans Assets '96 : Proceedings of the second annual ACM conference on Assistive technologies*, pages 2–9. ACM Press, 1996.

[146] Kirsten RASSMUS-GRÖHN, Charlotte MAGNUSSON et Håkan EFTRING : AHEAD - audio-haptic drawing editor and explorer for education. *Dans IEEE International Workshop on Haptic Audio Visual Environment and their Applications - HAVE 2007*. IEEE Computer Society Press, octobre 2007.

[147] Kirsten RASSMUS-GRÖHN, Charlotte MAGNUSSON et Håkan EFTRING : Iterative design of an audio-haptic drawing application. *Dans CHI '07 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2627–2632, San Jose, California, USA, avril 2007. ACM Press.

[148] Trygve M.H. REENSKAUG : Models - views - controllers. Rapport technique, Xerox PARC, décembre 1979. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

[149] Trygve M.H. REENSKAUG : THING-MODEL-VIEW-EDITOR - an Example from a planning-system. Rapport technique, Xerox PARC, May, Jon 1979. http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html.

[150] Jonathan C. ROBERTS et Keith FRANKLIN : Haptic Glyphs (Hlyphs) - structured haptic objects for haptic visualization. *Dans Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, WorldHaptics*, pages 369–374, Pisa, Italy, juillet 2005. IEEE Computer Society Press.

[151] Fabien ROBINEAU, Frédéric BOY, Jean-Pierre ORLIAGUET, Jacques DEMONGEOT et Yohan PAYAN : Guiding the surgical gesture using an electro-tactile stimulus array on the tongue : a feasibility study. *Transactions on Biomedical Engineering*, 54(4):711–749, 2007.

[152] A. F. ROVERS et Harm A. van ESSEN : Him : a framework for haptic instant messaging. *Dans CHI '04 : Extended abstracts on Human factors in computing systems*, pages 1313–1316, Vienna, Austria, avril 2004. ACM Press.

[153] Rami SAARINEN, Janne JÄRVI, Roope RAISAMO et Jouni SALO : Agent-based architecture for implementing mutimodal learning environments for visually impaired children. *Dans ICMI '05 : Proceedings of the 7th annual International Conference on Multimodal Interfaces*, pages 309–316, Trento, Italy, octobre 2005. ACM Press.

[154] Saitek. *Saitek*. http://www.saitek.com.

[155] J. Kenneth SALISBURY : Making graphics physically tangible. *Communications of the ACM*, 42(8):74–81, 1999.

[156] Eva-Lotta SALLNÄS, Fredrik WINBERG, Jonas MOLL, Andrew CROSSAN, Stephen A. BREWSTER, Dominique ARCHAMBAULT, Aurélie CASSON, Erika TANUA-PIIROINEN, Arto HIPPULA et Roope RAISAMO : Report on results from empirical studies of collaboration in cross-modal interfaces. *Dans MICOLE deliverable D9*. 2006.

[157] Sensable. *Sensable*. http://www.sensable.com.

[158] Calle SJÖSTRÖM : Using haptics in computer interfaces for blind people. *Dans CHI '01 : Extended abstracts on Human factors in computing systems*, pages 245–246, Seattle, Washington, avril 2001. ACM Press.

[159] Calle SJÖSTRÖM : *Non-Visual Haptic Interaction Design*. Thèse de doctorat, Certec, Lund Institute, Lund, Sweden, 2002.

[160] SRI. *SRI*. http://www.sri.com.

[161] Tactaid. *Tactaid*. http://www.tactaid.com.

[162] Hong Z. TAN, Robert GRAY, J. Jay YOUNG et Ryan TRAYLOR : A haptic back display for attentional and directional cueing. *Haptics-e*, 3(1), juin 2003.

[163] Thrustmaster. *Thrustmaster*. http://www.thrustmaster.com.

[164] Richard Q. van der LINDE, Piet LAMMERTSE, Erwin FREDERIKSEN et B. RUITER : The hapticmaster, a new high-performance haptic interface. *Dans Proceeding of the 2nd International Conference Eurohaptics 2002*, Edinburgh, Scotland, juillet 2002. ACM Press.

[165] Jan B. F. van ERP, A. H. C. van VEEN, Hendrik, Chris JANSEN et Trevor DOBBINS : Waypoint navigation with a vibrotactile waist belt. *ACM Transactions on Applied Perception*, 2(2):106–117, 2005.

[166] Steven A. WALL et Stephen A. BREWSTER : Assessing haptic properties for data representation. *Dans CHI '03 : Extended abstracts on Human factors in computing systems*, pages 858–859, Ft. Lauderdale, Florida, USA, avril 2003. ACM Press.

[167] Steven A. WALL et Stephen A. BREWSTER : Scratching the surface : Preliminary investigations of haptic properties for data representation. *Dans Proceeding of the 3rd International Conference Eurohaptics 2003*, pages 330–342, Dublin, UK, juillet 2003. ACM Press.

[168] Steven A. WALL et Stephen A. BREWSTER : Application based assessment of frictional properties for haptic data visualisation. *Dans Proceeding of the 4th International Conference EuroHaptics 2004*, pages 190–195, München, Germany, juillet 2004. ACM Press.

[169] Steven A. WALL et Stephen A. BREWSTER : Providing external memory aids in haptic visualisations for blind computer users. *Dans Proceedings of the 5th International Conference on Disability, Virtual Reality and Associated Technologies*, pages 157–164, New College, Oxford, UK, septembre 2004.

[170] Steven A. WALL et Stephen A. BREWSTER : Sensory substitution using tactile pin arrays : Human factors, technology and applications. *Special Issue of Signal Processing on multimodal interfaces*, 86:3674–3695, 2006.

[171] Steven A. WALL et Stephen A. BREWSTER : Tac-tiles : multimodal pie charts for visually impaired users. *Dans 4th Nordic Conference on Human-Computer Interaction*, pages 9–18, Oslo, Norway, octobre 2006. ACM Press.

[172] Qi WANG et Vincent HAYWARD : Compact, Portable, Modular, High-performance, Distributed Tactile Transducer Device Based on Lateral Skin Deformation. *Dans Proceedings of the 2006 Symposium on Haptic Interfaces For Virtual Environment And Teleoperator Systems*, pages 67–72, Arlington, VA, mars 2006. IEEE Computer Society Press.

[173] Jason WILLIAMS et Georg MICHELITSCH : Designing effective haptic interaction : inverted damping. *Dans CHI '03 : Extended abstracts on Human factors in computing systems*, pages 856–857, Ft. Lauderdale, Florida, USA, avril 2003. ACM Press.

[174] Robert L. WILLIAMS, Mayank SRIVASTAVA, Robert CONASTER et John N. HOWELL : Implementation and evaluation of a haptic playback system. *Haptics-e*, 3(3), May, Jon 2004.

[175] Wai YU, Rameshsharma RAMLOLL et Stephen A. BREWSTER : Haptic Graphs for Blind Computer Users. *Dans Proceedings of the First International Workshop on Haptic Human-Computer Interaction*, pages 102–107, 2000.

[176] Wai YU, Rameshsharma RAMLOLL et Stephen A. BREWSTER : Comparing Two Haptic Interfaces for Multimodal Graph Rendering. *Dans Proceedings of the 10th Symposium on Haptic In-*

terfaces for Virtual Environment and Teleoperator Systems (Haptics 2002), pages 3–9, Florida, USA, 2002.

[177] Wai YU, Rameshsharma RAMLOLL, Stephen A. BREWSTER et Beate RIEDEL : Exploring computer-generated line graphs through virtual touch. *Dans Sixth International Symposium on Signal Processing and its Applications 2001*, pages 72–75. IEEE Computer Society Press, 2001.

[178] Mounia ZIAT et Olivier GAPENNE : étude préliminaire visant la détermination de seuils de confort pour un zoom haptique. *Dans IHM 2005 : Proceedings of the 17th French-speaking conference of human-computer interaction*, pages 3–10, Toulouse, France, septembre 2005. ACM Press.

[179] Mounia ZIAT, Olivier GAPENNE, Charles LENAY et John STEWART : Acuité perceptive via une interface pseudo-haptique. *Dans IHM 2004 : Proceedings of the 16th French-speaking conference of human-computer interaction*, pages 263–266, Namur, Belgium, août 2004. ACM Press.

[180] Mounia ZIAT, Olivier GAPENNE, Marie-Odile ROUZE et Aymeric DELWARDE : Recognition of different scales by using a haptic sensory substitution device. *Dans Proceeding of the 6th International Conference EuroHaptics 2006*, Paris, France, juillet 2006.

[181] Mounia ZIAT, Olivier GAPENNE, John STEWART et Charles LENAY : Haptic recognition of shapes at different scales : A comparison of two methods of interaction. *Interacting with Computers*, 19(1):121–132, janvier 2007.

[182] Sophie ZIJP-ROUZIER et Éric PETIT : Teaching geometry to visually impaired pupils using haptics and sound. *Dans HCI 2005, Universal Access in Human-Computer Interaction conference*, pages 3–9, Las Vegas, Nevada, USA, juillet 2005.

Index

Symbols

épiderme 14
état 11

A

ABV 19
accord 90
agent 11, **130**
  de guidage 159
  de rendu 159
  input 136
  intelligent 3
  output 138
  PAC 5
  réactif 3
  sender 136
amplitude 98
API 129
API Reachin 131
architecture
  MICOLE 129, 143, 144, 146, 158, 175
auditory icons 29

B

beacons 40
boucle
  du scene graph 140
  temps réel 140
Braille 15, 17, 19, 20, 34, 35, **44**, 52, 165, 176
BraMaNet 52
bus 131
but 11

C

callback 132
canal 7
  système 7
  utilisateur 7
CARE 11, 158
champ 8
collider 140
Comtouch 21
cutané 13
CyberForce 24
CyberGrasp 26
CyberTouch 19

D

degré de liberté 21
Delta 24
derme 14
dimension 53
direction 98
durée 53

E

earcons 29, 42
Espace de Frohlich 7

F

Falcon 25
fenêtre 8
FIPA 130
Forhead Retina System 20
formalisme 8
forme 98
fuseau neuromusculaire 15

G

GUI 29, 34
guidage 141

H

haptic icons 32
Haptic Master 25
Haptic Workstation 26
haptique 12
HOMERE 43
homonculus sensitif 20
hypoderme 14

I

icône 8, 27, 29
  crossmodale 33, 89
  haptique 30
  non visuelle 29
  tactons 30
  visuelle 29
image 53
impulsion **98**
  guidée **98**
  séquence d' **99**
  semi-guidée **98**
information 54

J

joystick 22

K

kinesthésique 13

L

Lambda 52
langage
  de commandes 8
  de programmation 8
  naturel 8
longueur 53, 98

M

mécanorécepteur 15
média 7
matrice de picots 17
MaWEn 52
menu 8
mesure 91
modèle de force 140
mode 7
Moose 43
motif 52
multimodalité 129
muscle 14
MVC 4

N

nœud Reachin 159
nocicepteur 16
note 90
  figure de 90
  hauteur de 90
noyau fonctionnel 2

O

Omega 24
OpenGL 146
Optacon 19
organe tendineux de Golgi 15
os 14

P

périphérique
  de pointage 21
  haptique 16
  tactile 17
PAC 5, 158
PAC-Amodeus 5, 130
Pantographe 24
pantographe 24
paramètre 53
partition 92
pause 99
PC-Access 43
peau 14
Pencat/Pro 24
PHANToM **22**, 36, 39, 99, 128, 139

PICOB 98, 99, 141
point d'orgue 92
portée 92
propriocepteur 15
pseudo-haptique 12, 14
Put-That-There 7

R

récepteur polymodal 16

S

SALOME 46
scene graph 140, 146
silence 90
somatosensoriel 14
somesthésie 14
souris
  Logitech iFeel 24
  Virtouch 17
  VTPlayer 17
  Wingman Force Feedback 24
STReSS 19
STReSS^2 ..... 20, 47
style 8
système
  multi-agents 129

T

Tac-tiles 40
TACIS 41
TactiBall 18
tactile 13
TactiPen 18
TactiTab 18
Tacton 30, **52**
  à matrices de picots 52
  clignotant 54
  diagonal 54
  dynamique 53
  ensemble de 54
  mono-dimensionnel 54
  radial 54
  statique 52
  vagues 54
  vibrotactile 52, 177
Tactos 17
TeDub 42
Teletact 21
tempo 91
temps 91
  fort 91
tendon 14
thermorécepteur 16
Thing-Model-View-Editor 3
Tongue Display Unit 20
TYCOON 11

U

unité temporelle 53

V

VBW32 Tactor 19
vibreur 19
Virtouch 17
Virtuose 23
VITAL 18, 31
volant 22
VRML 146
VTPlayer **17**, 139, 142, 160

W

WIMP 29

Annexe A

Questionnaires

- Chapitre II : expérience 1 sur les Tactons mono-dimensionnels ..... page 206
- Chapitre II : expérience 2 sur les Tactons mono-dimensionnels ..... page 208
- Chapitre II : expérience 4 sur les Tactons mono-dimensionnels ..... page 209
- Chapitre II : expérience 5 sur les Tactons mono-dimensionnels ..... page 210
- Chapitre III : expériences sur les PICOB guidés ..... page 211
- Chapitre III : expériences sur les PICOB semi-guidés ..... page 212
- Chapitre IV : expériences sur le prototype de circuits électriques ..... page 213
- Chapitre IV : expériences sur l'application finale de circuits électriques ..... page 214

Questionnaire

Nom : _____ Âge : _____ Sexe : _____

Quel jeu d'icônes préférez-vous ?
- ◯ les petits clignotants
- ◯ les petits fixes
- ◯ les vagues
- ◯ les grands fixes

Quels sont les points forts et les points faibles des petits clignotants selon vous ? _____

Quels sont les points forts et les points faibles des petits fixes selon vous ? _____

Quels sont les points forts et les points faibles des vagues selon vous ? _____

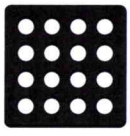
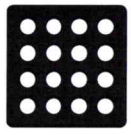
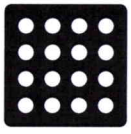
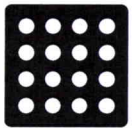
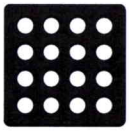
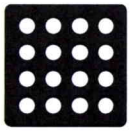
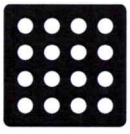
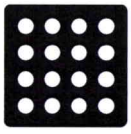
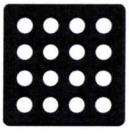
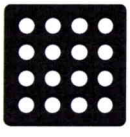
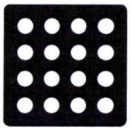
Quels sont les points forts et les points faibles des grands fixes selon vous ? _____

Trouvez-vous que certaines icônes sont trop semblables pour être reconnues ? Lesquelles ? _____

Comment trouvez-vous la vitesse de clignotement/mouvement des icônes dynamiques ? _____

Que pensez-vous de la souris ? _____

Avez-vous d'autres icônes à proposer ? Donnez un exemple d'horizontal (H) ou vertical (V) et de diagonal (D).

Direction	Étape 1	Étape 2	Étape 3	Étape 4	Étape 5	Étape 6	Étape 7	Étape 8
H / V								
D								
H / V								
D								
H / V								
D								
H / V								
D								

Questionnaire

Nom : _____ Âge : _____ Sexe : _____

Quel jeu d'icônes préférez-vous ?

- ◯ les icônes 4
- ◯ les icônes 5
- ◯ les icônes 6
- ◯ les icônes 7

Quels sont les points forts et les points faibles des icônes 4 selon vous ? _____

_____

_____

Quels sont les points forts et les points faibles des icônes 4 selon vous ? _____

_____

_____

Quels sont les points forts et les points faibles des icônes 5 selon vous ? _____

_____

_____

Quels sont les points forts et les points faibles des icônes 6 selon vous ? _____

_____

_____

Trouvez-vous que certaines icônes sont trop semblables pour être reconnues ? Lesquelles ? _____

_____

_____

Comment trouvez-vous la vitesse de clignotement/mouvement des icônes dynamiques ? _____

_____

_____

Que pensez-vous de la souris ? _____

_____

_____

Questionnaire

Nom : _____ Âge : _____ Sexe : _____

Quel jeu d'icônes préférez-vous ?
- ◯ les icônes 8
- ◯ les icônes 9

Quels sont les points forts et les points faibles des icônes 8 selon vous ? _____

Quels sont les points forts et les points faibles des icônes 9 selon vous ? _____

Trouvez-vous que certaines icônes sont trop semblables pour être reconnues ? Lesquelles ? _____

Comment trouvez-vous la vitesse de clignotement/mouvement des icônes 8 ? _____

Comment trouvez-vous la vitesse de clignotement/mouvement des icônes 9 ? _____

Avez-vous d'autres remarques ? _____

Questionnaire

Nom : _____ Âge : _____ Sexe : _____
Droitier / Gaucher : _____

Quel jeu d'icônes préférez-vous ?
- ◯ les icônes 10
- ◯ les icônes 11
- ◯ les icônes 10 inversées
- ◯ les icônes 11 inversées

Quel style de clignotement préférez-vous ?
- ◯ normal : l'indication de direction clignote
- ◯inversée : le repère clignote

Quels sont les points forts et les points faibles des icônes 10 selon vous ? _____

Quels sont les points forts et les points faibles des icônes 11 selon vous ? _____

Quels sont les points forts et les points faibles des icônes 10 inversées selon vous ? _____

Quels sont les points forts et les points faibles des icônes 11 inversées selon vous ? _____

Comment trouvez-vous la vitesse de clignotement/mouvement des icônes dynamiques ? _____

Autres remarques éventuelles _____

Questionnaire

Avez-vous eu des difficulés à reconnaître les directions ?

Avez-vous eu des hésitations sur certaines directions ? Lesquelles ?

Avez-vous senti la différence de hauteur entre les trois séries ?

Avez-vous des remarques ?

Questionnaire

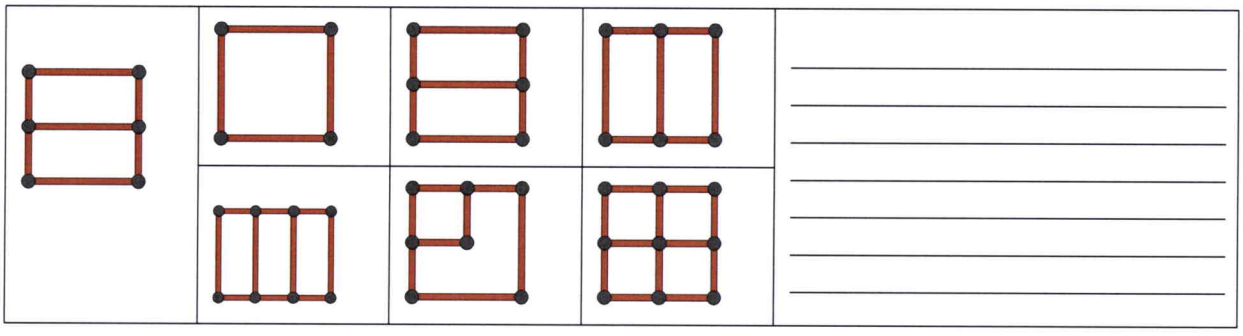
Avez-vous eu des difficulés à reconnaître les directions et les amplitudes ?

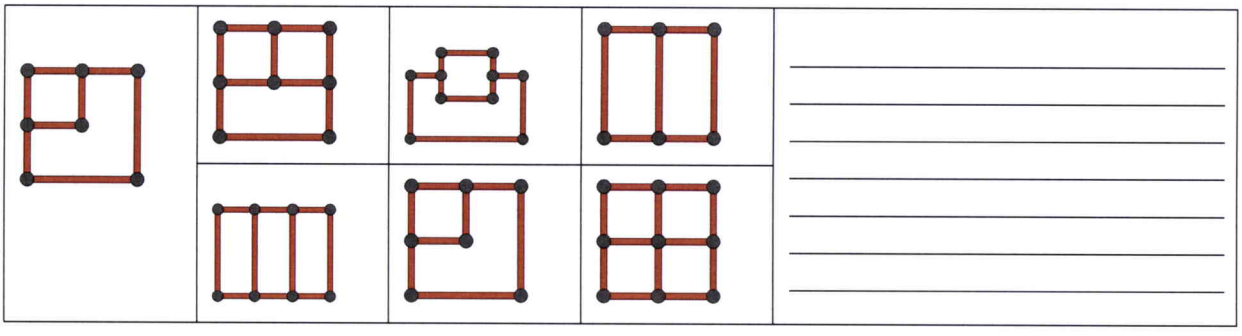
Avez-vous eu des hésitations ?

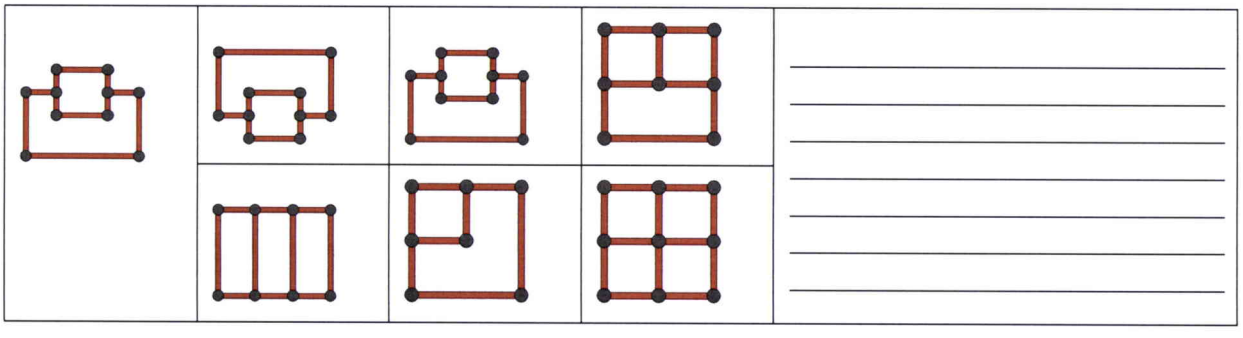
Avez-vous des remarques ?

Questionnaire

Nom : _____ Prénom : _____ Âge : _____ Sexe : _____







Questionnaire

Nom : .....
Prénom : .....
Âge : .....
Sexe : .....
Main dominante : .....
Déficience : .....
Connaissance du Braille : .....

Pré-questionnaire

Avez-vous déjà utilisé un périphérique haptique ? Si oui le(s)quel(s) ?

Quel matériel utilisez-vous d'ordinaire pour apprendre les circuits électriques ?

Vous faites vous aider par quelqu'un ? Sous quelle forme ?

Post-questionnaire

Êtes-vous d'accord avec ces assertions ?

	Pas d'accord		D'accord	
	1	2	3	4
Les composants sont difficiles à identifier.				
Il est difficile de parcourir les schémas en entier.				
Le fait d'être contraint sur le circuit aide à parcourir les schémas.				
Le guidage du PHANToM par la souris n'aide pas à parcourir les schémas.				
Les icônes sur la VTPlayer aident à parcourir les schémas.				
Les icônes tactiles sont difficiles à reconnaître.				
J'ai réussi à identifier tous les composants.				
Les icones tactiles sont plus utiles que les vibrations du Phantom pour identifier les composants.				
La présence de l'enseignant aide à mieux comprendre le circuit.				
La méthode traditionnelle permet de mieux comprendre des schémas de circuits électriques.				

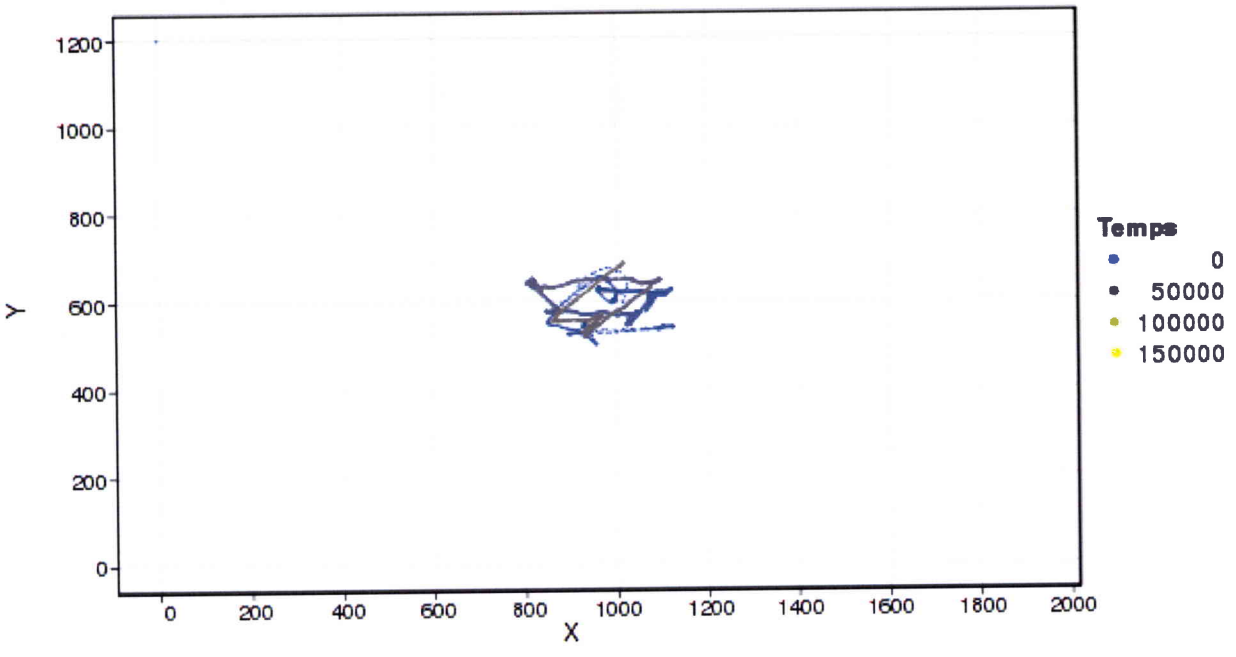
Annexe B

Traces d'exploration

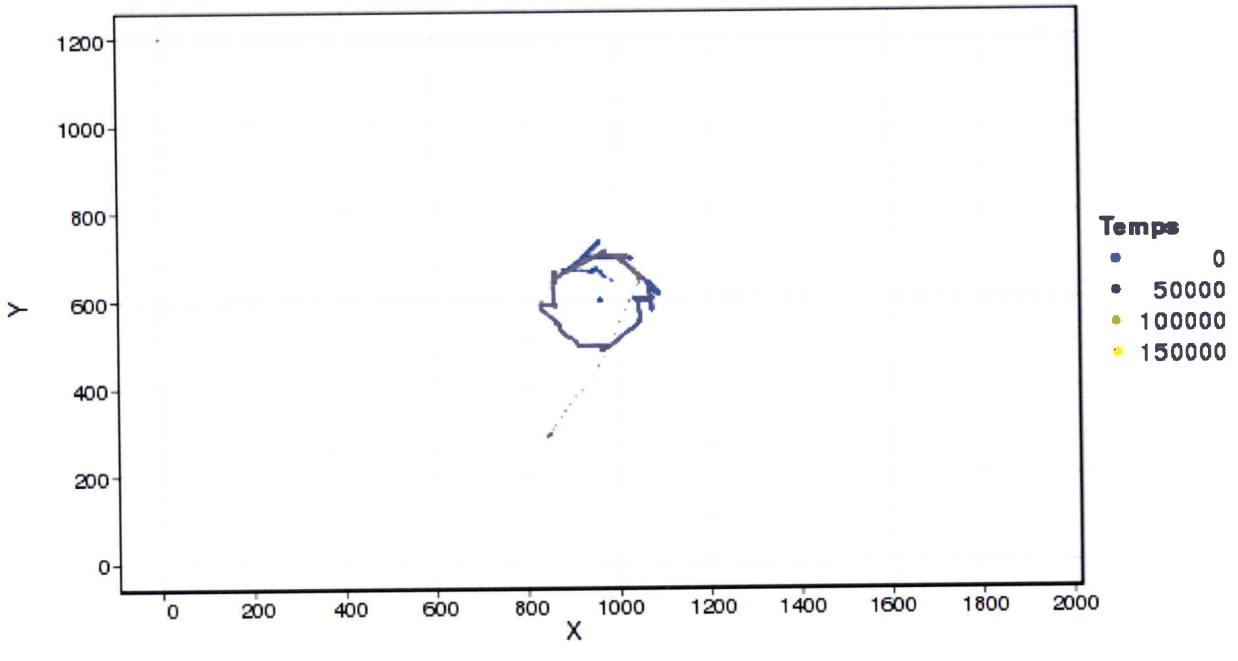
Ces traces représentent les trajectoires effectuées par les utilisateurs pour les réponses incorrectes.

- Utilisateur 1 ..... page 218
- Utilisateur 2 ..... page 219
- Utilisateur 3 ..... page 221
- Utilisateur 4 ..... page 225
- Utilisateur 5 ..... page 229
- Utilisateur 6 ..... page 230
- Utilisateur 7 ..... page 231

Utilisateur 1

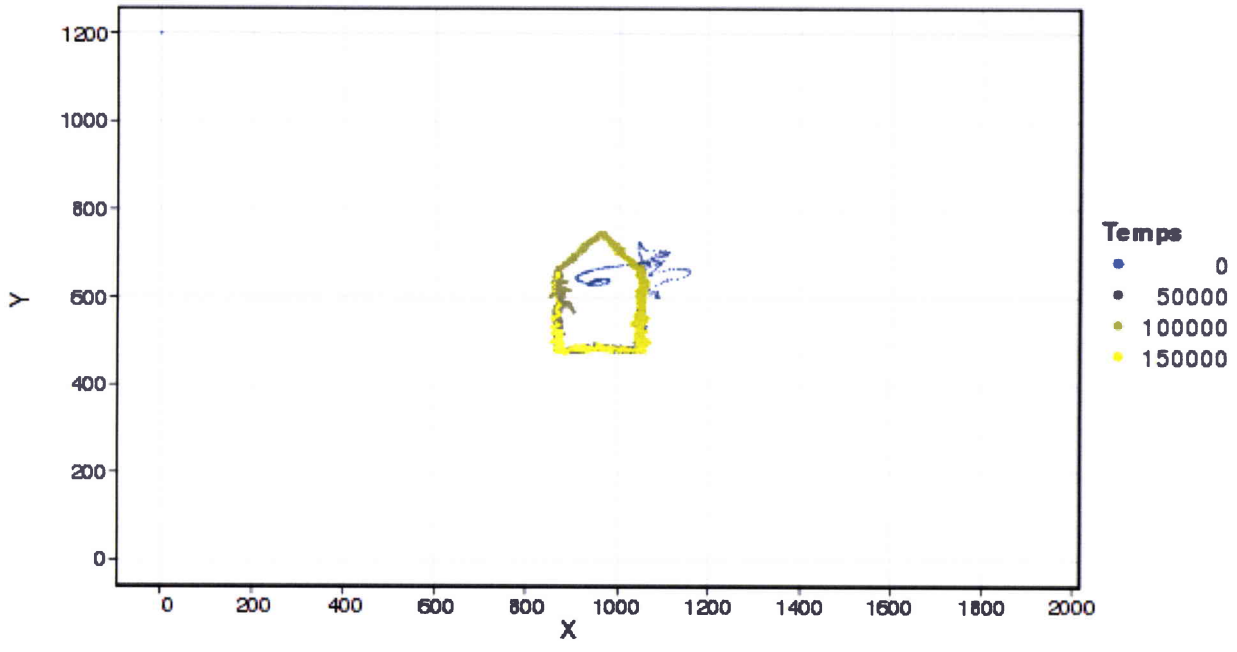


Forme 21, condition Pt

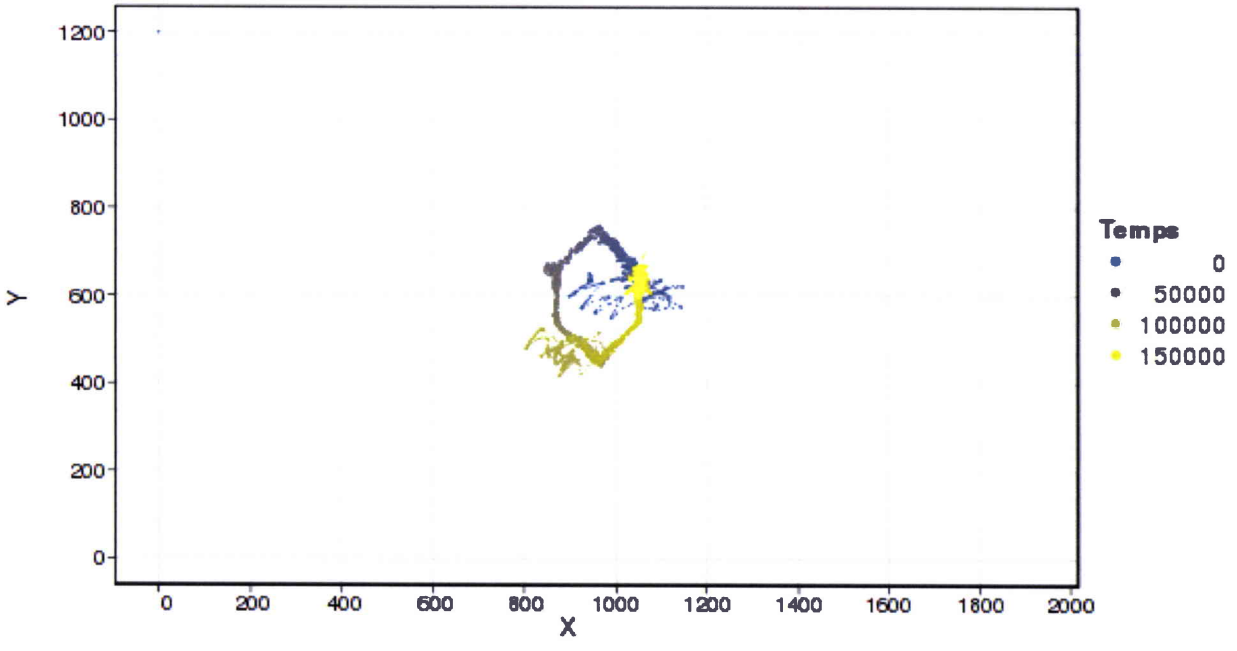


Forme 25, condition Pt

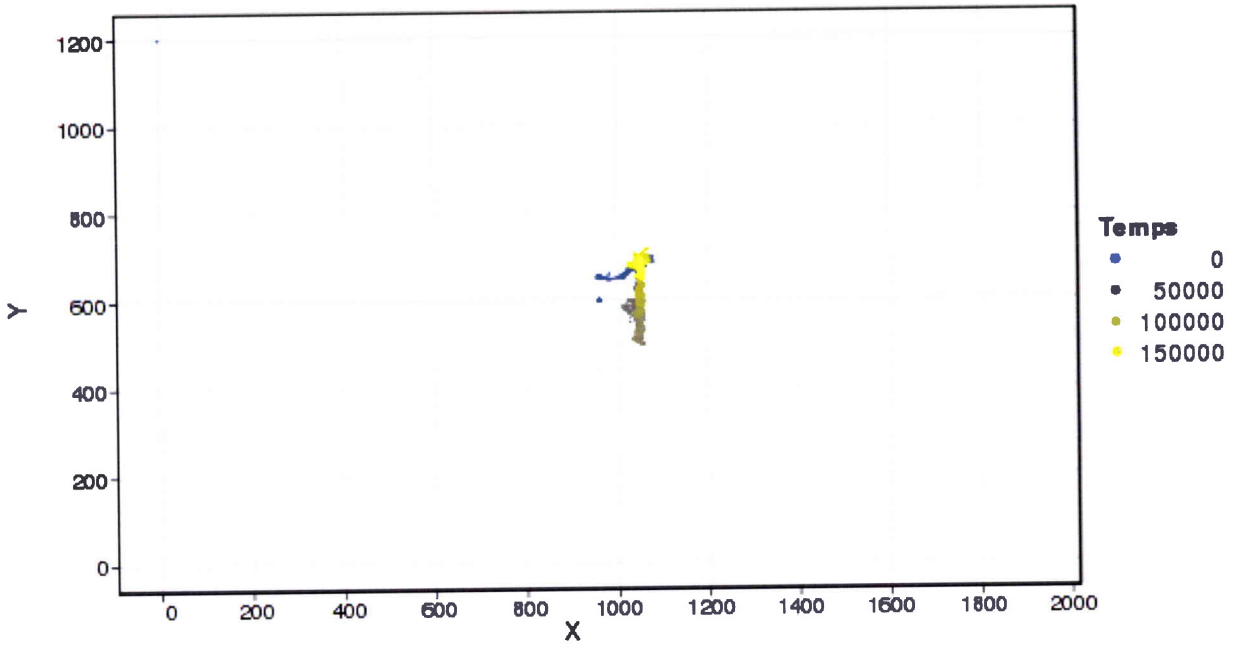
Utilisateur 2



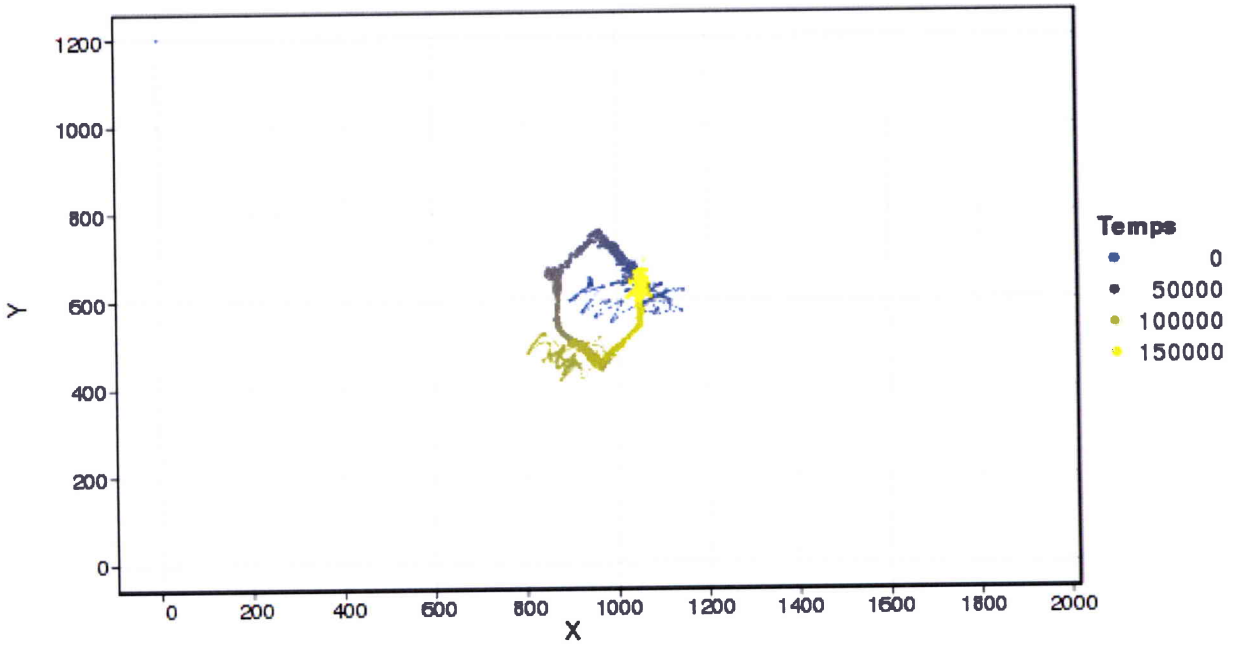
Forme 4, condition Pv



Forme 9, condition Pv

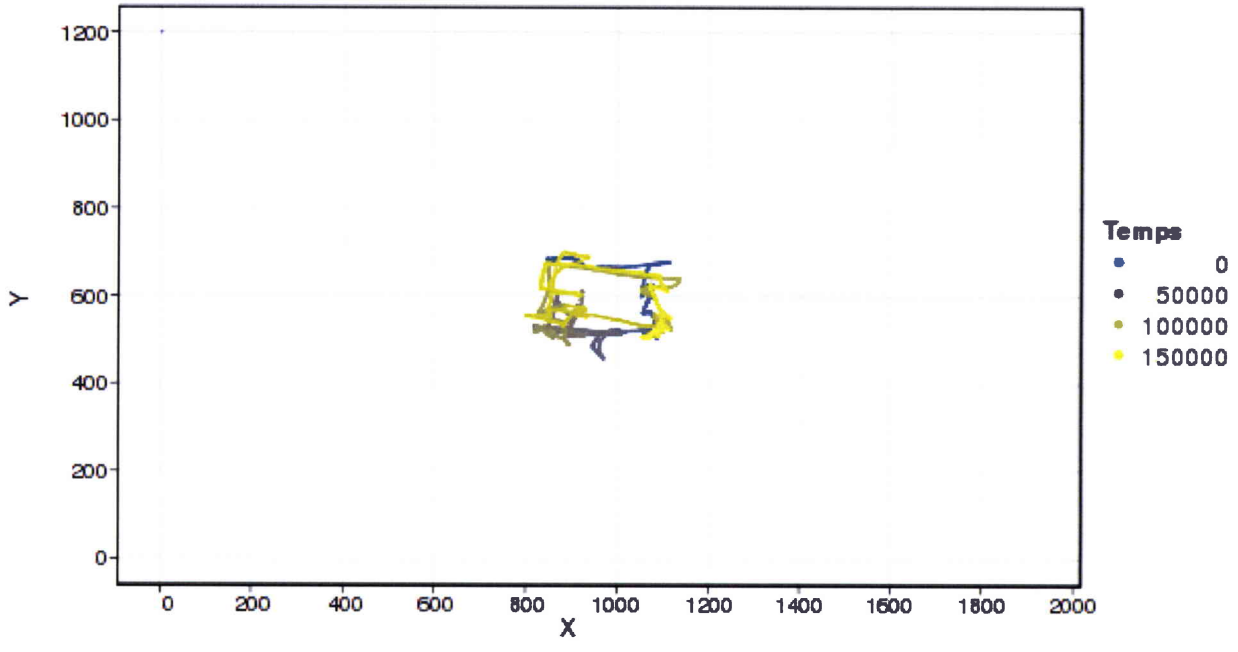


Forme 16, condition Pv

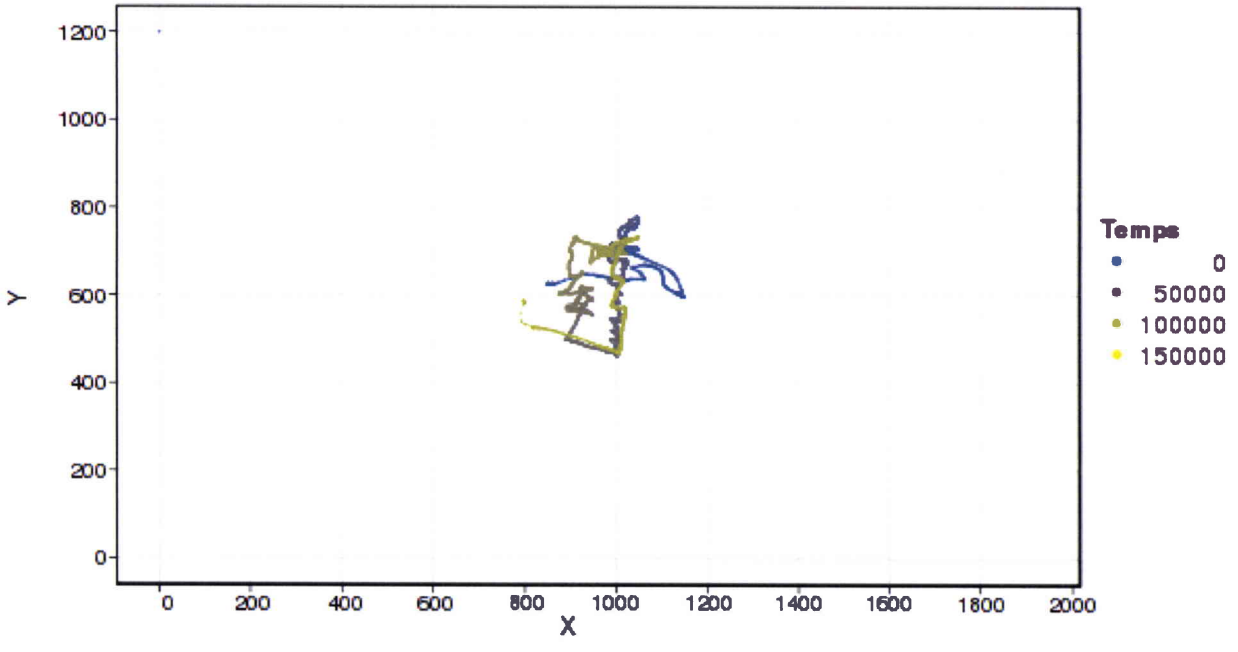


Forme 22, condition Pv

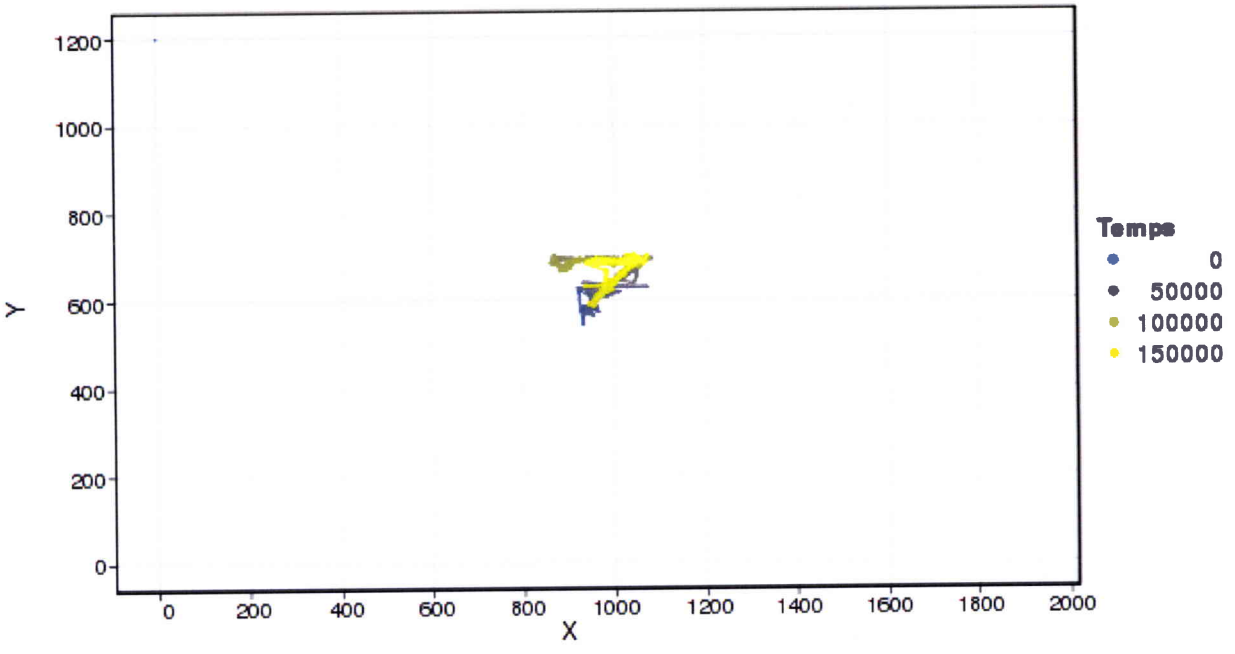
Utilisateur 3



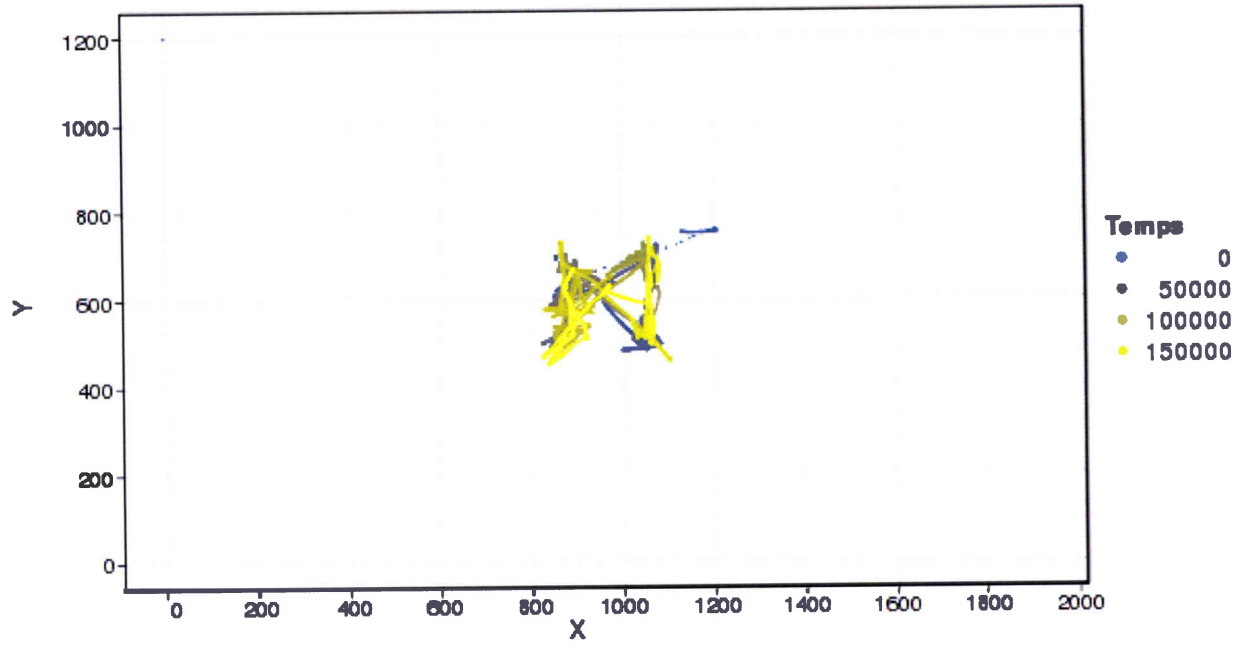
Forme 6, condition Pt



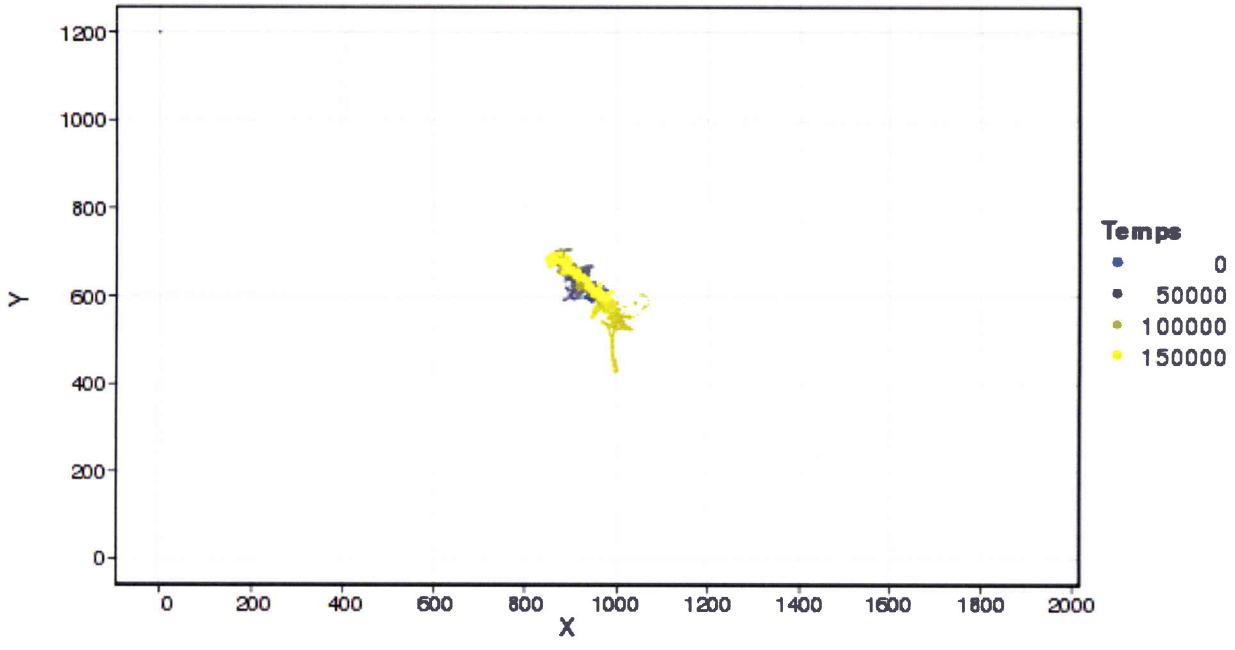
Forme 7, condition Pt



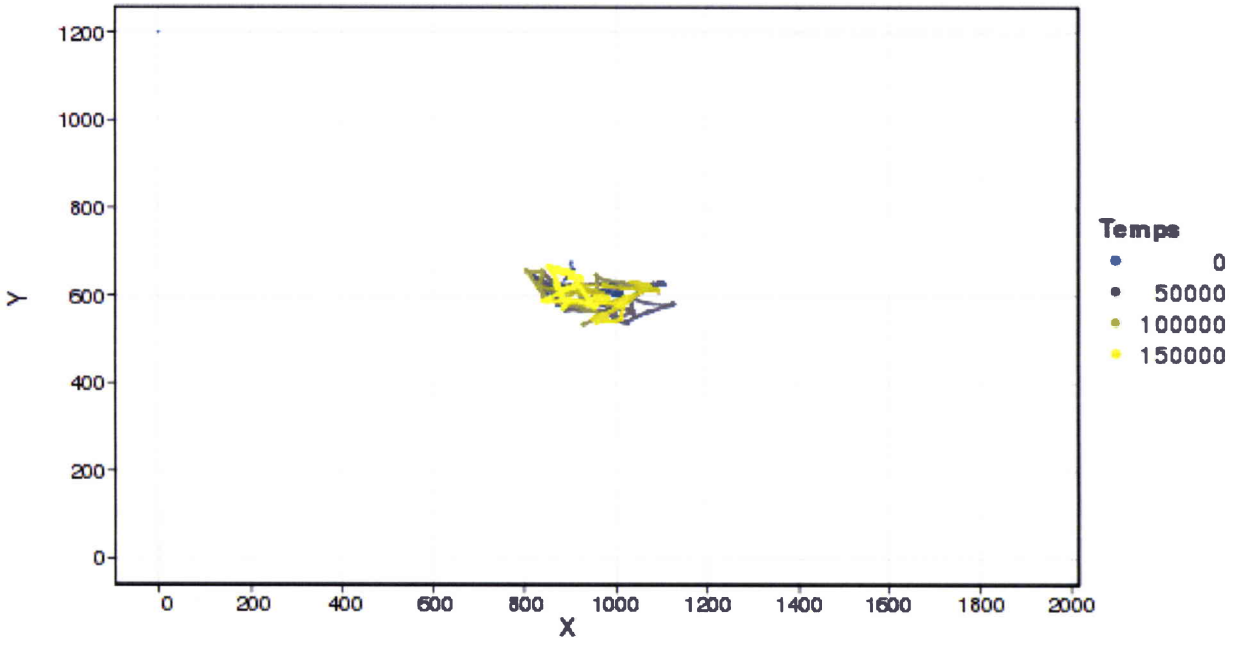
Forme 11, condition Pv



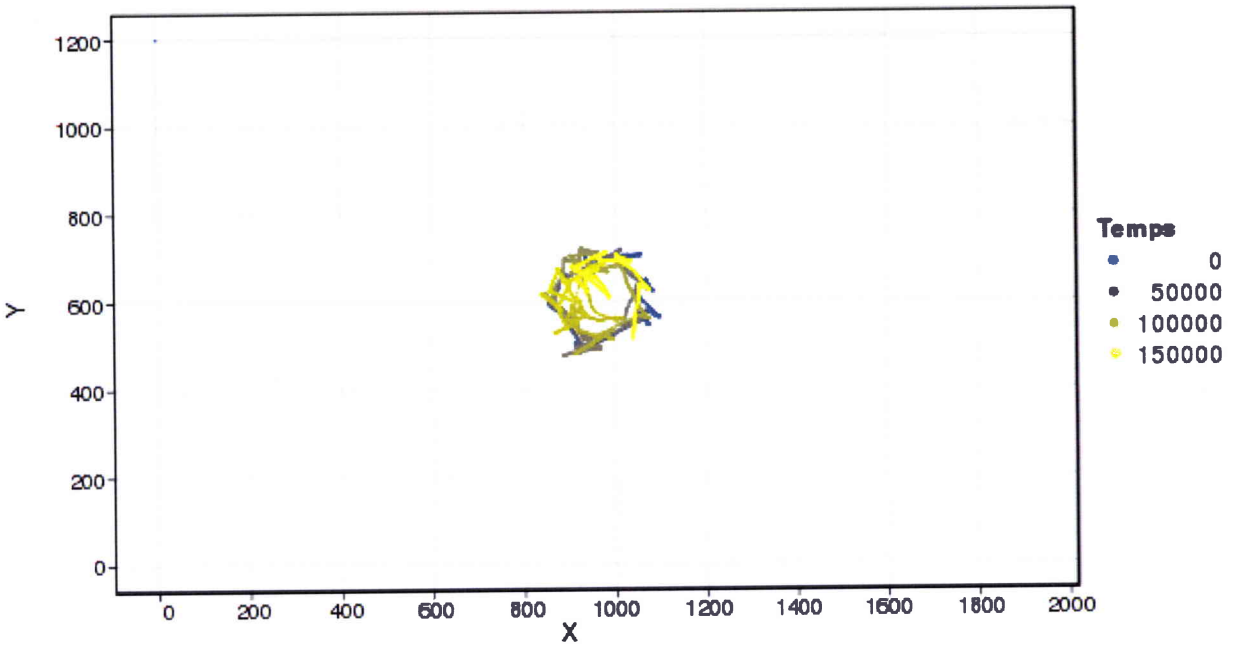
Forme 16, condition Pt



Forme 19, condition Pv

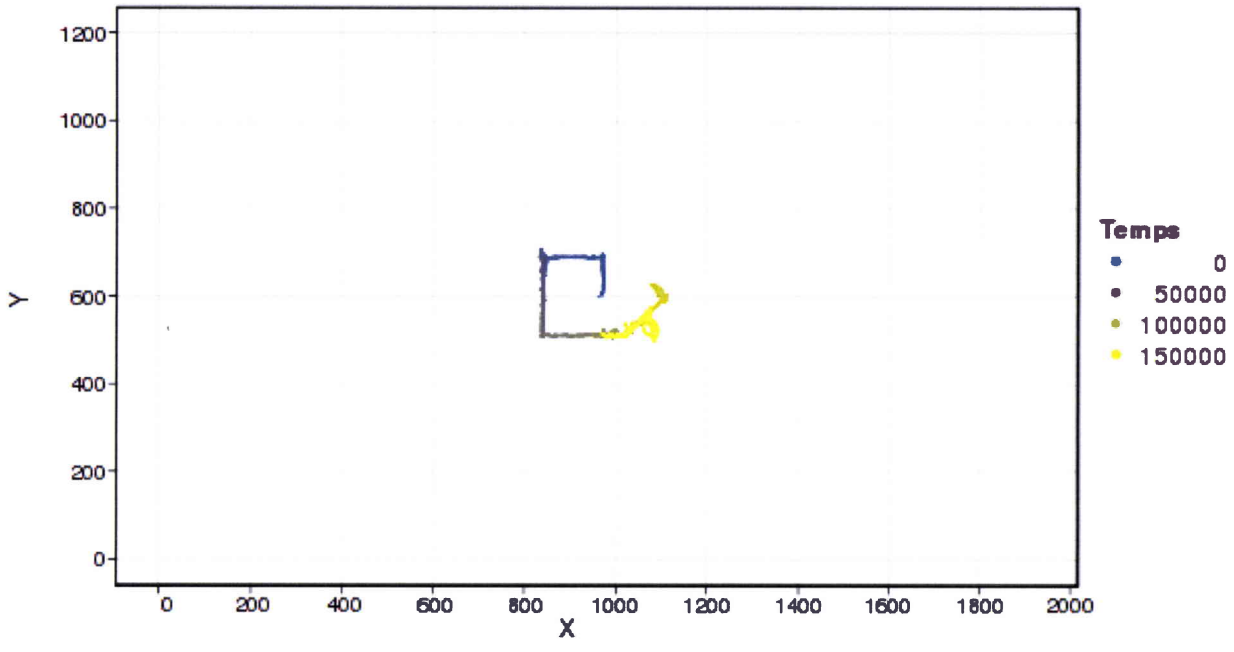


Forme 21, condition Pt

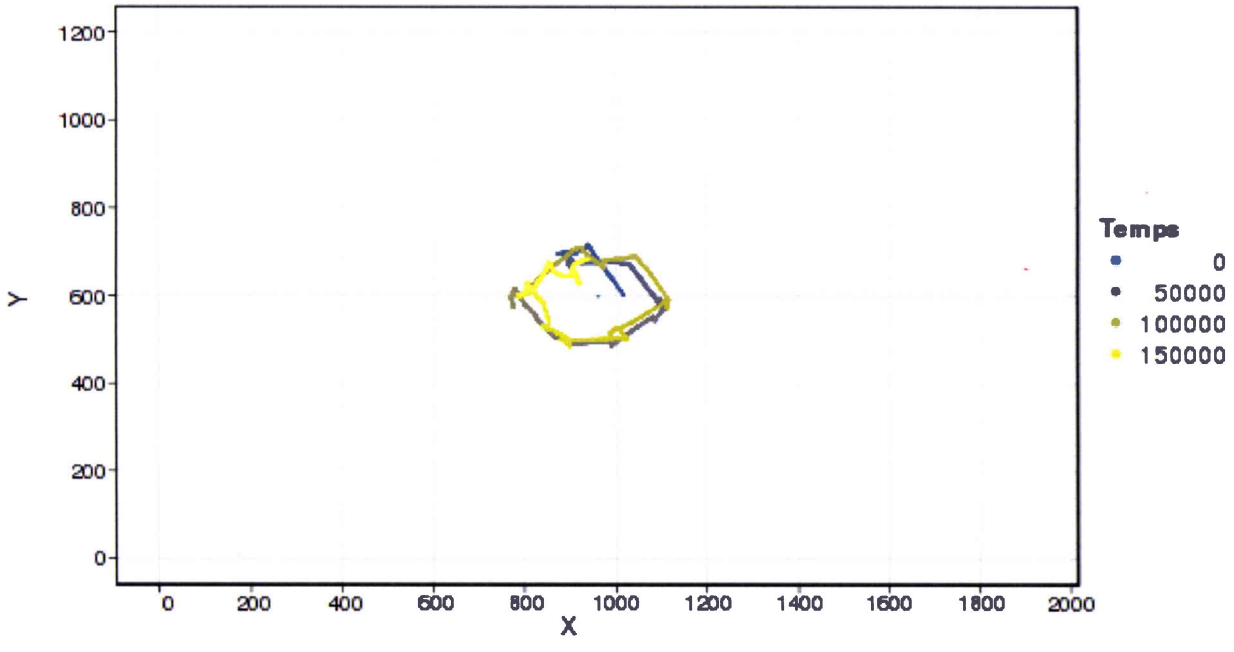


Forme 25, condition Pt

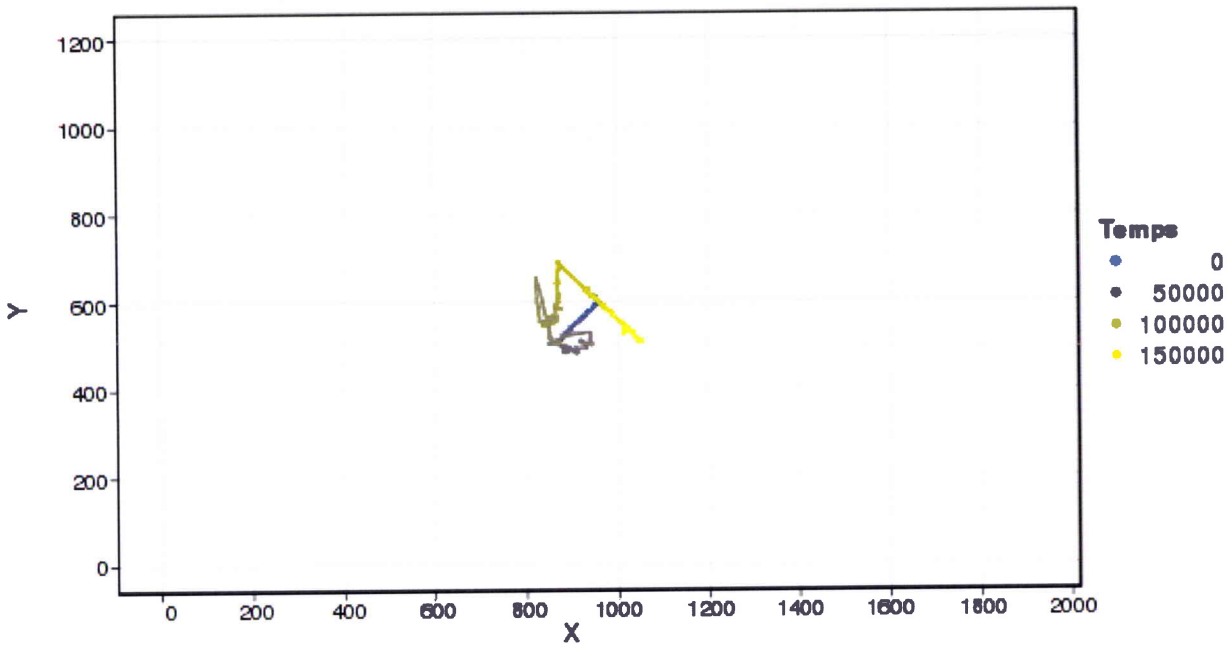
Utilisateur 7



Forme 5, condition Mv



Forme 8, condition Mt



Forme 16, condition Mv