THÈSE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

École Doctorale N°601
*Mathématique et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Signal, Image, Vision*

Par

Thibaud TOULLIER

**Simultaneous characterization of objects temperature and
radiative properties through multispectral infrared thermography**

Thèse présentée et soutenue à L'IFSTTAR - CENTRE DE NANTES , le 6 Novembre 2019
Unité de recherche : Inria
Thèse N° :

Rapporteurs avant soutenance :

Xavier MALDAGUE	Full Professor	Université Laval - Québec
Christophe PRADÈRE	Directeur de Recherche	Universite de Bordeaux

Composition du jury :

Président :	Guillaume MOREAU	Professeur des Universités	École Centrale de Nantes
Examinateurs :	Xavier MALDAGUE	Full Professor	Université Laval - Québec
	Christophe PRADÈRE	Directeur de Recherche	Université de Bordeaux
	Alexia GORECKI	Ingénieure R&D	LYNRED
Dir. de thèse :	Laurent MEVEL	Directeur de Recherche	Inria
Co-dir. de thèse :	Jean DUMOULIN	Chargé de Recherche	IFSTTAR

Invité(s) :

Laurent IBOS	Maître de Conférence	Université Paris-Est Créteil

There is no real ending. It's just the place where you stop the story.

Franck Herbert

À Mélina et Naomi

Remerciements

Je souhaite remercier avant tout mes encadrants qui m'ont accompagné durant ces trois années. En premier lieu, merci à Jean DUMOULIN pour l'ensemble des remarques scientifiques qui ont concrétisé cette thèse. Au-delà des échanges professionnels, merci pour ta compréhension, ta patience ainsi que tes conseils avisés.

Je remercie également Laurent MEVEL pour nos discussions très pertinentes qui ont beaucoup apporté dans la thèse et ce jusqu'aux derniers instants. Mes différentes visites à Rennes ont à chaque fois permis une grande progression dans mes travaux.

Ainsi, l'implication des différents membres d'I4S fût très bénéfique. En particulier, je tiens à remercier Qinghua ZHANG pour les échanges que nous avons eu qui ont fait mûrir ma thèse. Même si je n'ai pas toujours compris l'ensemble des remarques immédiatement, celles-ci se sont avérées pertinentes et m'ont fait de prendre du recul sur mes travaux. De la même manière je souhaite remercier Frédéric GILLOT pour le temps passé à réfléchir à mes équations. Merci notamment pour ta pédagogie et pour m'avoir présenté le Krigeage qui fût finalement un point-clé de ces travaux.

Je remercie l'IFSTTAR, l'Inria ainsi que la région Bretagne pour avoir financé cette thèse.

Je remercie Guillaume MOREAU pour avoir bien voulu présider mon jury. Des remerciements aussi aux rapporteurs Xavier MALDAGUE et Christophe PRADÈRE pour leur examen minutieux du manuscrit ainsi que leurs remarques qui ont permis de finaliser et d'éclaircir certains points. J'adresse également des remerciements à Alexia GORECKI qui a accepté de faire partie du jury et Laurent IBOS pour avoir assisté à la soutenance et m'avoir suivi lors du comité de suivi de thèse.

Je souhaite aussi remercier Sébastien BOURGUIGNON de l'École Centrale Nantes pour m'avoir offert l'opportunité de donner des cours mais aussi de m'avoir guidé dans ce processus.

D'une manière moins académique, ces trois années de thèses furent une expérience personnelle épanouissante, notamment grâce à l'ambiance qui règne au sein du laboratoire SII et de l'équipe I4S. Les différents collègues (permanents, stagiaires, doctorants et post-docs) rencontrés ont été une source d'enrichissement. Merci ainsi à Louis-Marie COTTINEAU à qui je souhaite une bonne retraite et à Vincent LE CAM (aérobic) les directeurs successifs du laboratoire. Des remerciements aussi à l'ensemble des chercheurs avec qui j'ai pu intéragir, Vincent BALTAZAR, David BÉTAILLE, Xavier CHAPELEAU et Alex COIRET.

Je me dois de saluer aussi l'ensemble des collègues qui contribuent à cette bonne entente : Yveline pour son travail de l'ombre, Ivan pour l'animation des débats autour du café, Quentin pour ses goûts musicaux, Jean-Luc pour ses connaissances en champignons et ses connaissances tout court, Jean-Marc pour nos échanges sur le traitement d'images et Jean-Philippe pour ses conseils de bricoleur. Je pense

aussi à ceux qui ont été de passage et qui l'ont marqué, Ludo pour ne citer personne. Une pensée à tous les autres doctorants qui vont bientôt soutenir. Dernière ligne droite Shreedhar ! Courage Antoine, tu es le prochain sur la liste, bonne chance pour tes futures fonctions.

Salut à toi Arthur à qui je n'aurais jamais dû dire que je voulais faire un triathlon, quelle idée ! Merci au futur docteur David pour tous les échanges que nous avons pu avoir et que nous aurons sur Kalman et l'informatique en général (enfin quelqu'un qui comprend mes délires informatiques !). Courage pour la suite !

Merci à tous mes collègues de bureau, notamment Guillaume qui a eu le courage de se reconvertir en architecte Java. Une grande pensée pour Nico qui m'a emmené dans des drôles d'aventures (encore désolé pour la boule noire) et qui a répondu à mes nombreuses questions mathématiques. Profite bien de la campagne, je publierais une version du jeu python un jour. Enfin, merci à Martin maître-pneu investisseur, le petit dernier du bureau qui regorge d'idées innovantes. Accroche-toi, la route est longue mais le jeu en vaut la chandelle !

Merci à Emacs pour me simplifier la vie.

Mes remerciements les plus chaleureux s'adressent à toute ma famille et à ma belle-famille qui n'ont cessé de me soutenir et de m'aider durant ces années, notamment pour que la maison avance.

Je termine ces pensées avec toi Mélina. Tu as été mon soutien le plus fort grâce à ta présence à mes côtés au jour le jour et tous les sacrifices que cela a représenté. Il y a une part de toi dans l'aboutissement de ces travaux. En cette fin de thèse, tu m'as offert la plus belle rencontre qui soit et il est ainsi temps de démarrer notre nouvelle vie à trois.

Résumé étendu en Français

Introduction

L E travail réalisé dans cette thèse s'inscrit dans un contexte de développement croissant de l'utilisation de nouvelles technologies pour la surveillance de structures. Le contrôle de santé des infrastructures est primordial afin d'assurer la pérennité des réseaux de transports et des constructions. Au regard des récents évènements en Italie, la prévention et la réparation des infrastructures est un enjeu à la fois social et économique. Pour pallier la vétusté des structures, des moyens de mesures afin de quantifier leur état sont nécessaires. En particulier, l'utilisation de caméras infrarouges bas-coûts pour la surveillance long-terme d'infrastructures est prometteuse grâce aux récentes avancées technologiques du domaine. En effet, les derniers développements technologiques ainsi que la miniaturisation des détecteurs infrarouges non-refroidis permettent d'avoir des caméras bas-coûts et faciles à mettre en œuvre. La thermographie infrarouge vise alors à fournir une mesure sans-contact et plein champs de la température. Cependant, une mesure précise de la température des surfaces observées *in-situ* se heurte au manque de connaissance des propriétés radiatives de la scène. En effet, la thermographie infrarouge permet de mesurer un flux radiatif arrivant à la caméra. Cependant, ce flux est la résultante de différentes contributions radiatives : l'émission propre de l'objet, permettant de remonter - en théorie - à sa température, mais aussi l'atmosphère, le soleil, les autres objets environnants de la scène, l'optique de la caméra, *etc.*

Une étude bibliographique de la mesure par thermographie infrarouge ainsi que l'écriture des équations du transfert radiatif met en exergue ce phénomène complexe multi-varié. De cette étude se dégage trois voies d'amélioration de la précision de la mesure par thermographie infrarouge.

Surveillance long-terme d'infrastructures par thermographie infrarouge

Dans un premier temps, il est montré que la correction des paramètres environnementaux inhérents à la mesure ainsi que l'exploitation de données issues d'une instrumentation multi-capteurs permettent d'accroître la précision de la mesure. Dans l'optique d'une mesure précise, les processus de calibration spatiale et thermique sont tout d'abord présentés. Afin d'exploiter l'ensemble des données provenant d'une instrumentation multi-capteurs, un logiciel de visualisation et de traitement est présenté. En particulier, un module YAML permet un traitement d'un large nombre de données par lots ; facilitant ainsi l'ensemble du processus. L'interface graphique du logiciel permet une vue d'ensemble des mesures et assiste les utilisateurs dans l'exploitation des données. Le modèle de conversion des im-
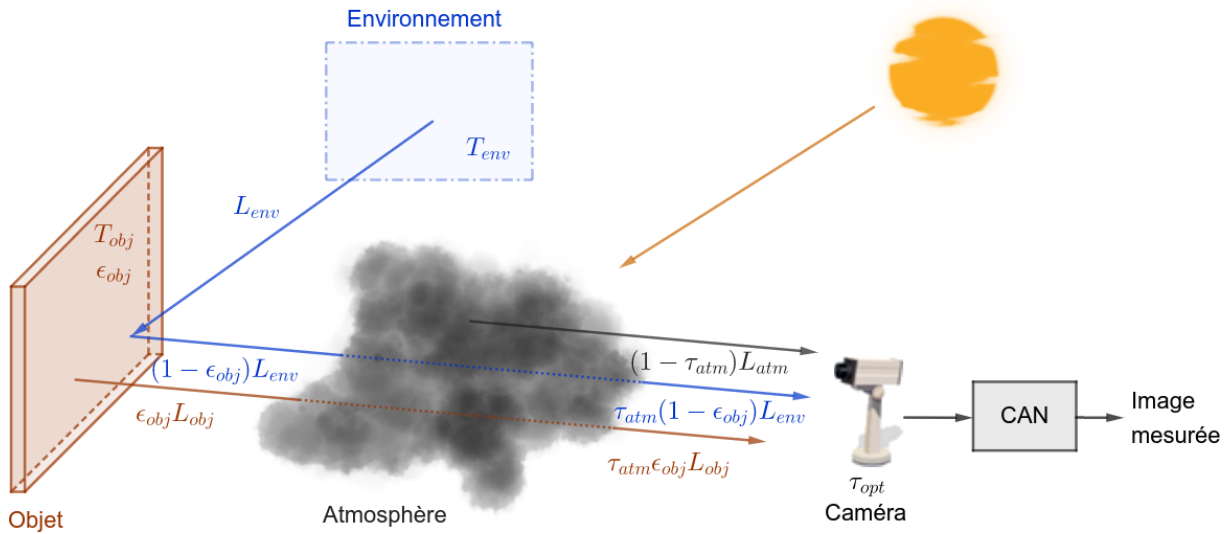
Figure 1: Principe de la mesure par thermographie infrarouge: le flux résultant dépend de l'environnement et des différentes réflections de la scène.

ages infrarouges en niveaux numériques vers la température est parallélisé par une approche GPGPU (*General-purpose processing on graphics processing units*), afin d'obtenir un logiciel de traitement efficace.

Ensuite, l'exploitation des données multi-capteurs montre qu'il est possible d'améliorer l'estimation de la température. De plus, des solutions faisant usage de bases de données climatiques en ligne et libres d'accès sont proposées pour pallier tout manque d'instrument de mesure.

Enfin, une étude de sensibilité des différents paramètres est menée. Il apparaît alors que l'émissivité, propriété radiative propre à tout matériaux, est un facteur prépondérant dans l'estimation de la température ; nonobstant l'usage de données multi-capteurs. Par conséquent, des méthodes d'estimation conjointe d'émissivité et de température sont proposées dans la suite.

Simulateur multi-spectral de scènes 3D

Dans un second temps et dans l'optique d'évaluer les méthodes d'estimation conjointe d'émissivité et de température, un simulateur de scènes 3D complexes dans l'infrarouge a été étudié et développé. Suite à une étude bibliographique, la méthode des radiosités progressives est sélectionnée pour son approche par éléments finis. Cet algorithme a été implémenté en utilisant l'accélération matérielle des machines ainsi que les dernières bibliothèques graphiques. Une approche GPGPU permet une parallélisation des calculs pour un rendu accéléré des radiosités.

L'importation de scènes 3D issues d'outils de CAO (Conception assistée par ordinateurs) ainsi que la visualisation et le rendu sont faits en temps réel via une interface graphique. L'intégration d'un interpréteur Python au sein du logiciel le rend programmable par l'utilisateur. Ainsi, des scripts Python

peuvent être utilisés pour interagir avec la scène 3D et le moteur de rendu. Des scènes statiques ou dynamiques peuvent alors être générées.

De plus, l'intégration d'un modèle de rayonnement solaire spectral et de transmission atmosphérique permet une simulation accrue de scènes *in-situ*. De même, l'introduction d'un premier modèle de caméra infrarouge équipée d'un détecteur quantique dont les performances métrologiques sont paramétrables permet d'optimiser de nouvelles solutions d'instrumentations pour des essais *in-situ*.
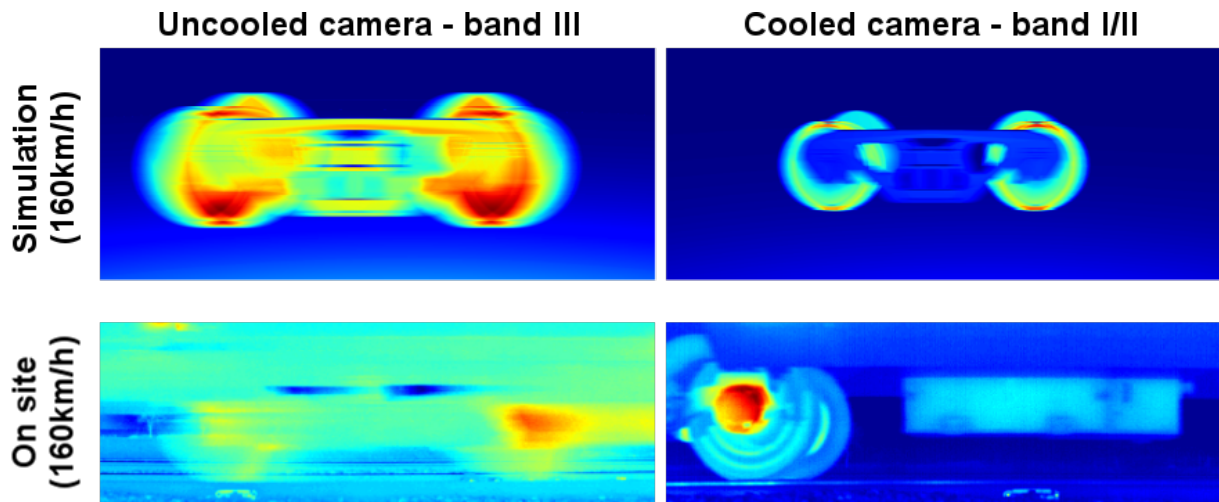
Figure 2: Exemple de rendu dynamique avec des boogies de trains, sur deux bandes spectrales différentes. Les données simulées sont comparées à des données terrain.

Méthodes d'estimation conjointe d'émissivité et de température

Enfin, quatre méthodes d'estimation conjointe d'émissivité et de température sont comparées, dont trois nouvelles. Un cas d'étude est réalisé à l'aide du simulateur précédemment introduit dans le but de tester ces méthodes. Ce cas d'étude est constitué d'une cible composée de quatre matériaux différents dont les propriétés radiatives sont connues. Les matériaux sont supposés être des couches minces, contrôlées en température. À l'aide d'une commande en température provenant de mesures *in-situ* au sol sur deux jours, des images sont simulées. Un parangonnage des différentes méthodes est alors réalisé.

Le premier algorithme, inspiré de la littérature, se base sur une méthode de Monte-Carlo par chaînes de Markov. Par inférence statistique d'*a priori* sur les distributions d'émissivité et de température, l'estimation simultanée d'émissivité et de température est réalisée pixel par pixel à chaque instant de la mesure. Les résultats ainsi obtenus sont satisfaisants et permettent de remonter aux valeurs d'essai. Cependant, malgré ces résultats encourageants, le temps de calcul est important. Des améliorations de cette technique sont possibles en utilisant une parallélisation voire une méthode MCMC d'ensemble.
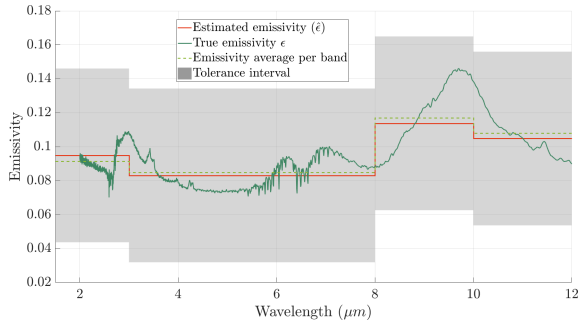
(a) Méthode MCMC: estimation de l'émissivité, ici avec un profile correspondant à l'or.
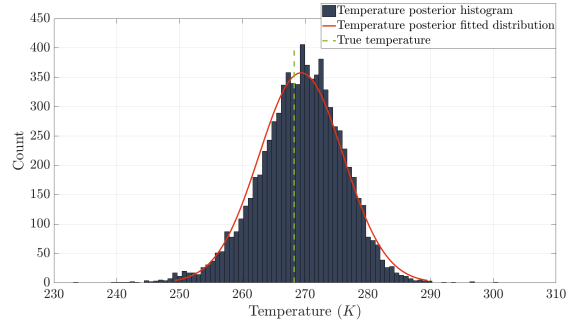
(b) Méthode MCMC: histogramme permettant d'obtenir l'estimé en température.

Figure 3: Exemple de résultat pour la méthode MCMC pour un pixel à un temps donné. (a) Estimation de l'émissivité. La valeur attendue (continue) est comparée à la valeur estimée (par bande). (b) Estimation de la température. La valeur attendue est représentée en pointillés.

La seconde méthode s'appuie sur un filtre de Kalman en interaction. Ce filtre permet de suivre l'évolution temporelle de la température via un filtre de Kalman ainsi qu'une estimation des paramètres du modèle grâce à un filtre particulier. La méthode montre de bons résultats sur le cas d'étude mais repose cependant sur une hypothèse de bandes spectrales fines, dont le cas d'application se limite à des données hyper-spectrales et non multi-spectrales.
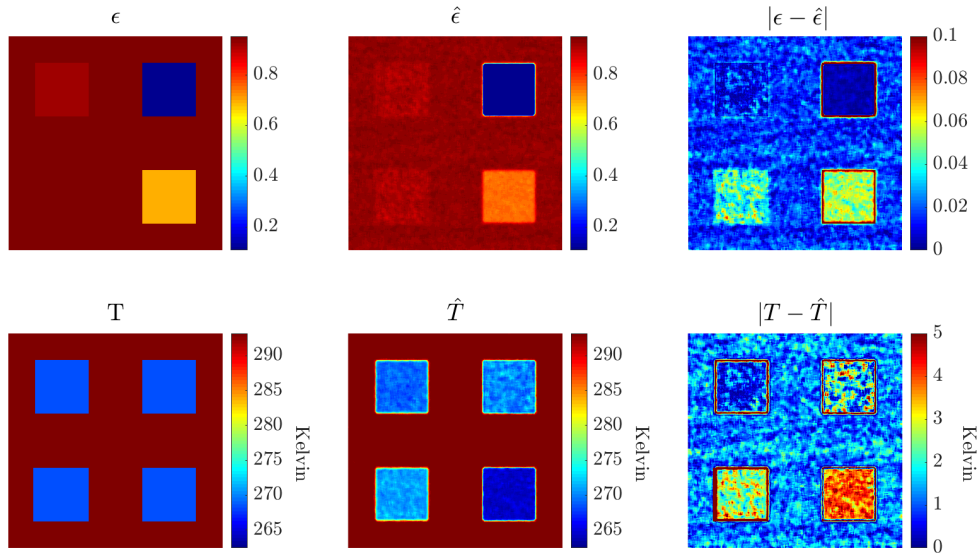

Figure 4: Exemple de résultat pour la méthode CMA-ES. Une erreur relativement faible sur une valeur faible d'émissivité peut induire une erreur importante dans l'estimation de la température

Ensuite, la métaheuristique d'optimisation CMA-ES (*covariance matrix adaptation evolution strategy*) a été appliquée au problème en utilisant l'hypothèse de surfaces homogènes localement. En supposant que des pixels adjacents ont la même température à l'équilibre thermodynamique local, il devient pos-

sible de résoudre le système d'équations et d'estimer ainsi l'émissivité et la température. À nouveau, les résultats sont satisfaisants et l'application de la méthode est aisée. Cependant, comme pour la méthode MCMC, le temps d'exécution est important. Bien qu'une parallélisation soit possible, une application à des données multi-spectrales long-terme avec déploiement *in-situ* entièrement basé sur cette méthode semble difficile.

Afin de pallier ces précédentes limitations, une dernière méthode reposant sur un filtre de Kalman en interaction krigé (KIKF) est proposé. Cette méthode combine la précédente méthode du filtre de Kalman en intéraction avec une méthode de Krigeage dans le but d'inférer un modèle de covariance spatiale à l'émissivité. L'hypothèse de température homogène locale est aussi considérée. L'ensemble des solutions possibles est alors réduit permettant de résoudre le problème en mono-spectral. L'avantage de cette méthode est sa capacité à suivre l'évolution des paramètres et de la température en même temps. Ainsi, lorsque le krigeage dégénère, il est possible de remarquer une modification des propriétés spatiales. De plus, par comparaison aux précédentes métaheuristiques, l'exécution est réalisée en un temps raisonnable qui est potentiellement parallélisable (moins d'une minute d'exécution entre deux mesures sans parallélisation).
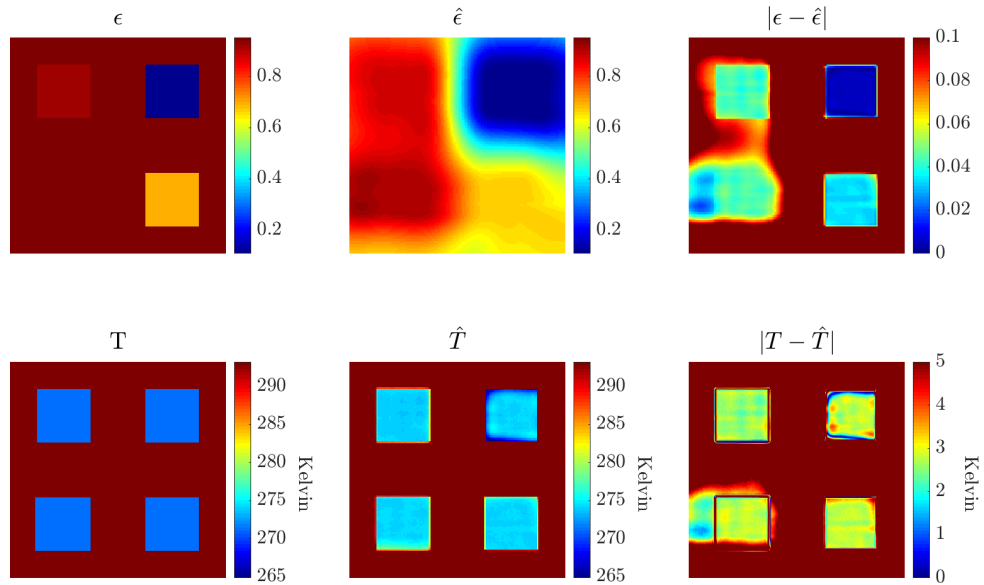

Figure 5: Estimé final après traitement de l'ensemble des mesures sur une bande spectrale avec la méthode KIKF, pour deux jours de mesures simulées.
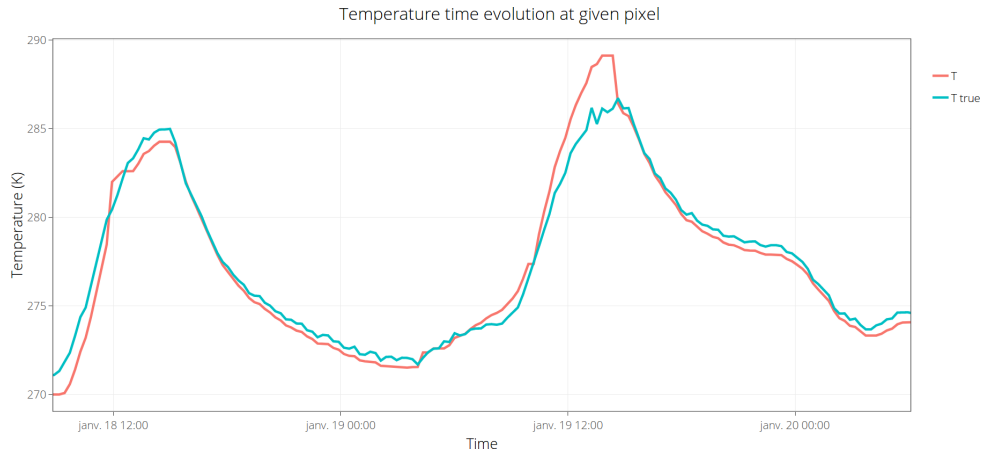
Figure 6: Température estimée sur un pixel avec la méthode KIKF: evolution de la température au cours du temps. Comparaison avec la valeur attendue.

Conclusions et perspectives

Les travaux effectués dans cette thèse mettent en perspective l'utilisation de la thermographie infrarouge pour la surveillance de structures comme outil quantitatif. Une étude bibliographique a permis de mettre en avant trois moyens d'améliorer la précision de la mesure de la température.

Premièrement, l'exploitation de données issues de multiples capteurs permet de corriger les effets environnementaux et d'améliorer ainsi l'estimation de la température. Lorsque les données locales ne sont pas disponibles, des solutions en ligne et libres d'accès sont proposées.

Ensuite, le développement d'un simulateur d'échanges radiatifs diffus sur des scènes 3D complexes permet d'optimiser de nouvelles solutions d'instrumentation et ce, en amont de toute exploitation. Ce simulateur peut être amélioré pour intégrer d'autres phénomènes physiques tels que la diffusion ou la convection, ainsi que des environnements participatifs. De plus amples modèles de caméras et détecteurs peuvent aussi être ajoutés.

Enfin, un ensemble de méthodes utilisant l'inférence statistique a été proposé. Chacune de ces méthodes s'appuie sur un ensemble d'hypothèses et des méthodes de résolutions différentes. Si les méthodes MCMC et CMA-ES offrent de bons résultats, leur temps d'exécution reste long et un suivi dynamique des paramètres n'est pour l'instant pas possible. *A contrario*, la méthode KIKF offre une résolution en un temps relativement faible. Des travaux complémentaires, notamment sur l'optimisation des paramètres intrinsèques au filtre pourraient être conduits. De plus, d'autres approches d'optimisation pourraient être étudiée en vue d'améliorer la convergence du filtre.

Dans tous les cas, les méthodes proposées offrent un résultat prometteur. En effet, une quantification de la mesure de la température par thermographie infrarouge est réalisée grâce à une estimation de l'incertitude, inhérente à ces méthodes statistiques.

Table of Contents

Nomenclature

Subscript

$\cdot_a$	Aerosol related
$\cdot_{atm}$	Atmospheric contribution
$\cdot_{env}$	Environmental contribution
$\cdot_g$	Mixed gas related (mainly O_2 and CO_2)
$\cdot_\lambda$	Spectral quantity (wavelength dependent) $\qquad\qquad$ μm
$\cdot_{\bar{\Delta}\lambda_i}$	Defined on the median of the spectral band
$\cdot_{\Delta\lambda_i}$	Defined on a given spectral band ($\int_{\lambda_i - \Delta\lambda/2}^{\lambda_i + \Delta\lambda/2} (\cdot)\, d\lambda$)
$\cdot_{land}$	Land surface contribution
$\cdot_n$	Nitrogen dioxide related
$\cdot_o$	Ozone related
$\cdot_{obj}$	Object contribution
$\cdot_{opt}$	Optical contribution
$\cdot_R$	Rayleigh scattering related
$\cdot_{sun}$	Solar contribution
$\cdot_{tot}$	Total contribution
$\cdot_w$	Water vapor related
$\cdot_{pc}$	Pixel p and channel c

Superscript

$\cdot^\circ$	Black-body related quantity
$\cdot^{(i,j)}$	Defined for a given pixel (i,j)
$\cdot^\uparrow$	Up-welling quantity
$\cdot^\downarrow$	Down-welling quantity

Constants

C_1	First radiation constant ($= 2hc^2$)	$3.741\,771\,790\,075\,259 \times 10^{-16}\,\mathrm{J \cdot m^2 \cdot s^{-1}}$

C_2	Second radiation constant ($= \frac{hc}{k_B}$)	$1.438\,777\,353\,827\,7 \times 10^{-2}\,\mathrm{m \cdot K}$
k_B	Boltzmann constant	$1.380\,649 \times 10^{-23}\,\mathrm{J \cdot K^{-1}}$

Greek letters

α	Absorption coefficient	
δ	Declination angle	
ϵ	Emissivity	
γ_s	Azimuth angle	
λ	Wavelength	μm
ν	Electromagnetic wave frequency of a given photon	Hz
Ω	Solid angle	sr
Φ	Radiant flux	W
ϕ	Flux emitted	W
φ	Latitude	
ρ	Reflection coefficient	
σ	Stefan-Boltzmann constant	$5.670\,374 \times 10^{-8}\,\mathrm{W \cdot m^{-2} \cdot K^{-4}}$
τ	Transmission coefficient	
θ_z	Zenith angle	
$\boldsymbol{\Sigma}_v$	Covariance matrix of process noise	
$\boldsymbol{\Sigma}_w$	Covariance matrix of measurement noise	

Other Variables

(c_x, c_y)	Principle points
$\boldsymbol{A}_k$	State matrix
$\boldsymbol{B}_k$	Control matrix
$\boldsymbol{C}_k$	Observation matrix
$\boldsymbol{D}_k$	Feedthrough matrix
$\boldsymbol{K}_k$	Kalman gain
$\boldsymbol{u}_k$	Control vector
$\boldsymbol{x}_k$	State vector

$\boldsymbol{y}_k$	Output vector	
a	Aspect ratio	
B	Thermal calibration constant 2	
c	Speed of light in vacuum	$299\,792\,458\,\mathrm{m \cdot s^{-1}}$
E	Irradiance	$\mathrm{W \cdot m^{-2}}$
e	Photon energy	J
h	Planck constant	$6.626\,070\,15 \times 10^{-34}\,\mathrm{J \cdot s}$
I	Radiant intensity	$\mathrm{W \cdot sr^{-1}}$
$\tilde{I}$	Directional emissive power	$\mathrm{W \cdot sr^{-1}}$
J	Radiosity	$\mathrm{W \cdot m^{-2}}$
L	Radiance	$\mathrm{W \cdot m^{-2} \cdot sr^{-1}}$
l	Longitude	
M	Radiant exitance	$\mathrm{W \cdot m^{-2}}$
M_T	Transformation matrix	
n	n^{th} day of the year	
Q	Radiant energy	J
R	Thermal calibration constant 1	
s	Skew	
T	Temperature	K
F	Thermal calibration constant 3	
f	Focal length	
$F_{i \to j}$	Form factor (or view factor) from element i to element j	

List of Figures

List of Algorithms

Introduction

INFRARED thermography is a non-contact and full-field measurement technique used in numerous infrastructure diagnosis applications thanks to its non-invasive nature and large scale implementation potential. In the context of energy savings, the thermal characterization of buildings envelopes plays an essential role. Determining the thermal properties of a structure helps at refining and integrating various phenomena in diagnostics tools in order to measure its thermal efficiency but also its aging. Obtaining refined data on large infrastructures makes it possible to optimize repair costs, and thus save both energy and economic expenses. The latest improvements on uncooled infrared (IR) cameras have brought new opportunities for inexpensive thermal diagnostics in the Civil Engineering's field. Although the technological maturity is ready, the methods to retrieve an accurate temperature measurement or at least with a known uncertainty are missing. In fact, the radiative flux that arrives at the camera's sensors is not directly related to the observed target temperature. Instead, this flux depends on a combination of many parameters that varies with the 3D geometry of the scene, the meteorological conditions, the self radiative properties of the objects, *etc.* In such context, the estimation of the temperature from those measurements is not trivial. If some parameters can be estimated from other local or online measurements, the most influential radiative quantity that will affect the temperature estimation is locally difficult to evaluate. This quantity is called the emissivity and depends on the object's own characteristics (material, roughness), but also the angle of view and the temperature. As a consequence, *in-situ* infrared thermography for the thermal inspection of infrastructures is mainly used as a qualitative tool.

The work done in this thesis aims at providing methods to estimate simultaneously the emissivity and the temperature during long-term and *in-situ* multi-spectral infrared thermography measurements. One of the advantage of *in-situ* measurements is the presence of the sun as a natural heating source to excite the observed surfaces. Those various conditions coupled with long-term monitoring and multi-spectral measurements provide a set of data from which the emissivity and temperature can be extracted under some assumptions and statistical priors. Even more, a confidence interval is deducted to make the infrared thermography usable as a quantitative tool. In order to develop such method, the equations that govern the radiative heat transfer are introduced and presented in Chapter 2. The radiative phenomenon is complex and the equations are written at different observations scales to understand fully the assumptions used in the final model equation. In particular, it is shown that even if multi-spectral data brings more information to the system it is still insufficient to estimate

simultaneously the temperature and the emissivity.

A bibliographical review on the IR measurement process and radiative balance, the numerical simulation of 3D thermal infrared scenes and the simultaneous estimation of emissivity and temperature through multi-spectral infrared thermal measurements has been conducted. This bibliographical study, presented in Chapter 3, highlights three ways to improve the temperature estimation accuracy.

The first way is to correct the environmental effects and exploit *in-situ* complementary environmental measurements. In Chapter 4, a description of how coupled environmental data measurements can improve the estimation result is presented. Moreover, when local measurements from multi-sensor acquisitions are not available, it is shown that open-data available online can also be exploited.

Then, in order to optimize new instrumentation solutions for *in-situ* studies and test developed emissivity and temperature estimation methods, a 3D simulation software tool has been developed and studied. This software, presented in Chapter 5 is implemented on graphic hardware to reduce the computational time. It enables the numerical simulation of the radiative diffuses exchanges of 3D complex scenes. Thanks to an atmospheric spectral solar irradiance model and sun modelization, *in-situ* dynamic scenes can be simulated. Furthermore, the acquisition chain of the irradiance received at the camera sensor is also modeled and complete the measurement process simulation.

Developed mathematical models for the simultaneous estimation of the temperature and emissivity are presented in Chapter 6. Two methods from the literature have been implemented and three new methods are then proposed. The first one is an interacting Kalman filter which is the combination of a particle filter on a Kalman filter. While the Kalman filter tracks the temperature evolution through time, the particle filter estimates the model's parameters. However, satisfying results are obtained only for small spectral bandwidth, which is suitable for hyper-spectral applications only. The second one is based on the covariance matrix adaptation evolution strategy (CMA-ES) meta-heuristic optimization algorithm. By assuming that on a given pixel's neighborhood the temperature is homogeneous, the equation system can be solved on a single-band image, at every moment. However, this robust method induces important computational costs. To overcome those previous issues, a combination of an interacting Kalman filter with a spatial approximation is proposed. By using a priori on the spatial distribution of the emissivity through Kriging, a single-band or multi-band emissivity and temperature estimation method that tracks the parameters through time is proposed. Those methods are applied to a study case, obtained by the previous introduced simulation tool and compared.

Finally, conclusions and perspectives on *in-situ* infrared thermography for the monitoring of infrastructures as well as obtained results are proposed and discussed.

Context and problem positioning

P ROPOSED methods in this thesis aim at making the temperature measurement of infrared thermal structure monitoring more accurate, or at least with a quantifiable error. This can only be achieved by identifying the surface properties of the observed object and in particular its emissivity, as exposed in this chapter. After introducing the concepts of the radiative transfer theory, the main equations to describe radiative exchanges in this thesis will be derived. Then, the main scientific challenges for *in-situ* infrared thermal monitoring are presented as well as the technological context on which this thesis is part of.

2.1   Structural Health Monitoring (SHM)

Structural health monitoring aims at detecting and characterizing damages on engineering structures. It has become a major public concern due to aging infrastructures and their intensive use [113]. A damage can be any change in the geometry or the material's properties of the infrastructure system that may affect its performance. The system is observed over time thanks to various sensors. This gathered information is then exploited to prevent catastrophic events, save maintenance costs but also for prognostic. Therefore, SHM relies on multiple pillars (see Fig. 2.1). The first one concerns the diagnosis of the infrastructure through the observed physical phenomenon, which can be seen as an extension of Non Destructive Evaluation (NDE) [13]. The second one is the monitoring system in itself and all the parameters that defines it: sensors technology, acquisition architecture, data processing, communication, *etc.* Therefore, SHM needs to consider the monitoring as a whole sub-system inside the structure: embedded and connected monitoring system integration, communication layers, power and even real-time data processing ability. All those elements need to reconsider the design and management of the structure to make it "smarter". Such complete systems have been studied for the monitoring of road transport infrastructures for example in [95], [176] or [49]. In those SHM applications, the monitoring is long-term, which needs to have a fully autonomous acquisition systems. Those systems are therefore critical for the monitoring and need to be scalable to any specific application. This implies to handle the various sensors [50], the amount of data to store, send [108] and process [63, 62]. One concern about most civil engineering structures is their subject to environmental parameters due to their outdoor characteristic.

The thermal monitoring of infrastructure is an interesting component of SHM. In fact, outdoor
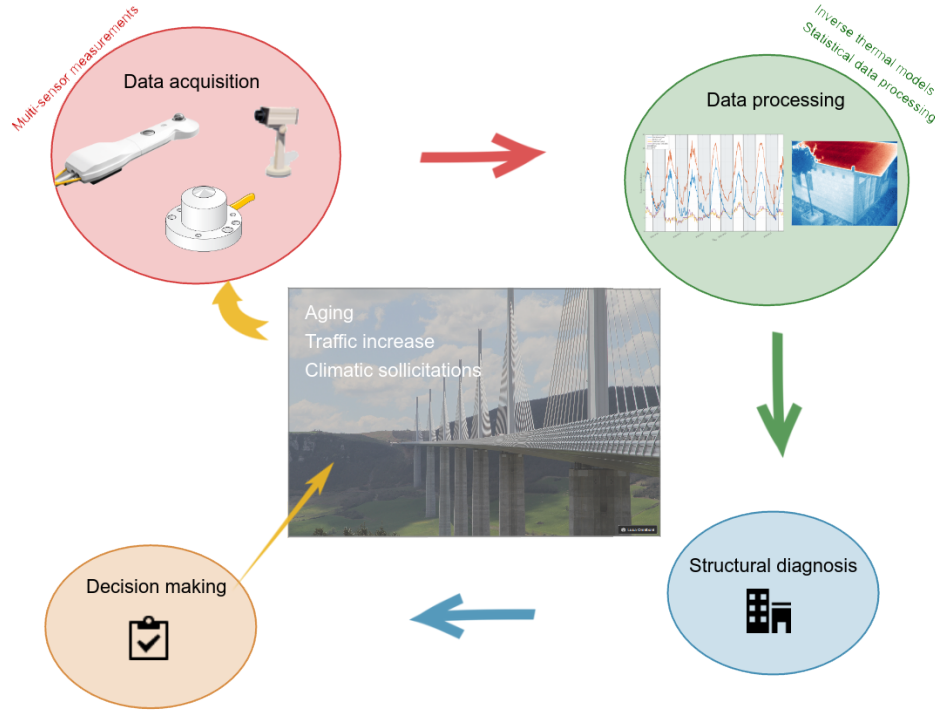
Figure 2.1: Main steps of structural health monitoring process

structures are subject to environmental parameters and more particularly to the solar flux, a natural heating excitation source for the system [105]. Thermo-physical properties of the structure can therefore been followed through time. In this thesis, emphasis is given to the monitoring of infrastructures through infrared cameras. In fact, infrared thermography cameras have experienced significant technological development over the last decade. Uncooled cameras with good resolution, relatively inexpensive and even multi-spectral are now possible [193]. Infrared thermography offers an interesting solution to monitor structure's thermal evolution through time. First, infrared cameras provide a non-contact and non-destructive technique which simplifies the setting up on field. Then, infrared thermography offers a multi-point measurement with a large field of view, suitable for large scale structures [36]. However, *in-situ* infrared measurements are often used as qualitative data only [80]. In fact, the measured flux that arrives at the sensor depends on the surrounding and meteorological environment (see Section 2.4.1 and Chapter 4). As a consequence, complementary data measurements are needed to estimate the temperature of the observed object. The objective of this thesis is to use this multiple sensors configuration and propose a method to estimate the temperature with a known uncertainty. The sensitivity of the measurement model to its different parameters will be studied on actual measurements.

Some elements of the thermal radiative transfers theory are given in the following section. This theoretical background will be used to derive the equations that govern the main phenomena that

occur during infrared thermography.

2.2 Thermal radiative transfers

Radiation transfer is one of the three means for thermal energy to be transferred with conduction and convection. The diffusion represents the transfer of heat via molecular interactions whereas radiation represents the transfer of heat via photons/electromagnetic waves. The radiative transfer theory (RT) aims at studying the absorption, emission and scattering of electromagnetic radiation as it passes through a medium. Such interactions can be described mathematically by the equation of radiative transfer. RT is involved in a wide variety of fields, including remote sensing, atmospheric science, astrophysics, optics *etc.* The purpose of this section is to provide the main principles of the RT theory and to introduce its concepts through the equations. Those definitions and a more detailed description of RT theory can be found in standard textbooks [93, 26, 53]

2.2.1 Radiative transfer theory

For the purposes of this study, radiation is viewed as the transport of energy in electromagnetic waves. All substances continuously emit electromagnetic radiation due to the different interactions and motions of the molecules and atoms they are made. The term *thermal radiation* is reserved for the visible and infrared portion of the spectrum since the related radiation should be detected by either heat or light [23]. The infrared part of the electromagnetic spectrum is given in Fig. 2.2.
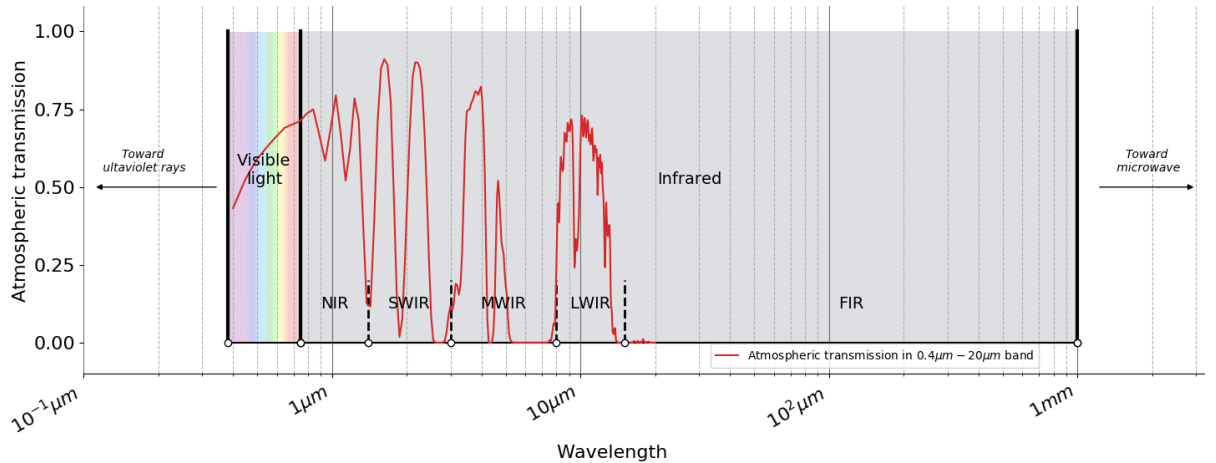
Figure 2.2: Focus on the IR part of the electromagnetic spectrum. Common spectral bands of IR are shown: *Near-infrared* (NIR: $0.75 - 1.4\mu m$); *Short-wavelength infrared* (SWIR $1.4 - 3\mu m$); *Mid-wavelength infrared* (MWIR $3 - 8\mu m$); *Long-wavelength infrared* (LWIR $8 - 15\mu m$); *Far infrared* (FIR $15 - 1000\mu m$). The atmospheric transmission is also represented.

In addition to emitting their own radiation, the radiation can be scattered, absorbed, reflected and

transmitted (also, those effects can affect the frequency of the radiation beam). All those different interactions make the thermal radiation a complex phenomenon. Absorption and scattering effects can be studied on their own: the size of the particle relative to the wavelength of the radiation, the index of refraction (complex) of the material and also the shape of the particle will have an impact on the scattered proportion of the incident radiation. All of this leads to the study of different parameters such as the particle's *phase function*, the *scatter cross section* and the *absorption cross section*. The purpose of this study is not to go too deeply in the description of all the physical effects that occur since it will make the approach too complicated. However, it is still important to know that other physical phenomena exist at lower scales that cannot be taken into account in the presented models. For more readings on this subject, see [182] and [21]. Two approaches can be used for describing the radiation theory: the classical electromagnetic theory and quantum mechanics. Whereas quantum mechanics will take into account the microscopic interactions of the radiation with matter, the classical theory such as Maxwell's equations will only derive the macroscopic behavior. As explained before, we cannot go too deeply onto lower scales since the mathematical complexity of the approach will increase and become unusable for practical study cases. Therefore, a combination of the two approaches is generally used in the litterature. For example, the spectral intensity is derived from the photon model whereas the absorption, transmission and emissivities are considered from a more classical approach as constant of proportionnality in the equations. This is why most of the time those parameters are derived from experimental studies even though they could be developed by the quantum mechanics through Einstein's transition probability coefficients. In such context, the main physical quantities involved into the RT theory and mainly described in [93] will be presented. Then, based on those definitions, the different radiative transfer equations will be derived.

2.2.2 Basic radiation properties

Let first start by giving some definitions on temperature. One can define temperature differently [16]:

- *Thermodynamic temperature:* T is defined according to the second principle of thermodynamics, for a medium in thermal equilibrium, which can be measured directly by a thermometer. For non-isothermal bodies, the temperature $T(x, y, z)$ is defined as the temperature of an elementary isothermal volume at the location (x, y, z). The surface temperature is then defined by the limit of $T(x, y, 0)$ with a width approaching zero.

- *Radiometric temperature:* This time, the temperature is defined from the radiance L_λ emitted by a surface and defined later on in this chapter. The radiance L_λ measured by a radiometer may be written with appropriate approximations and the temperature retrieved from those equations.

It is important to note that the surface temperature is well-defined for homogeneous materials at thermal equilibrium. However, it becomes much more complicated when considering non isothermal and heterogeneous materials. Most of the time in this thesis, the materials will be considered homogeneous

and that thermal equilibrium is reached.

Radiation can be viewed as the transport of energy by discrete photons for which the relationship between its energy and frequency ν (or wavelength $\lambda = c/\nu$) is given by Planck's relation:

$$\epsilon = h\nu = \frac{hc}{\lambda} \tag{2.1}$$

where h is Planck's constant and c the speed of light in vacuum. Eq. (2.1) shows an interesting fact: the energy of the radiation is inversely proportional to the radiation wavelength. This property can be easily observed by looking at the change of a burning flame as it goes hotter. Hotter temperatures mean higher energy which leads to a shift in frequency and therefore, in its color. In the XIX$^{\text{th}}$ century, the Stefan-Boltzmann law was established, deriving the maximum emission value for a given surface at a given temperature.

Definition 2.2.1 (Stefan-Boltzmann Law). We can describe the power radiated from the black-body in terms of its temperature:

$$M^\circ(T) = \sigma T^4 \tag{2.2}$$

This relation describes the power radiated from a black body surface (denoted with the superscript $^\circ$) which depends only on the temperature of the surface and is expressed in W/m^2. All real surfaces will emit radiation at a rate smaller than M° which represents an ideal surface called black body (more details on black body and real surfaces radiation will be given in the next section). Then, at the beginning of the XX$^{\text{th}}$ century, Planck showed that the radiative energy emitted by a black body could be expressed over the spectrum a relation now called the Planck's law.

Definition 2.2.2 (Planck's law). Planck's law describes the radiative energy emitted by a black body to its temperature (T) over the spectrum through the Planck's function denoted $L_\lambda^\circ(T)$. An increase of temperature will lead to an increase of the emitted frequency.

$$L_\lambda^\circ(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \tag{2.3}$$

Two constants are often used to simplify the writing of Planck's law $C_1 = 2hc^2$ and $C_2 = \frac{hc}{k_B}$:

$$L_\lambda^\circ(T) = \frac{C_1}{e^{\frac{C_2}{\lambda T}} - 1} \tag{2.4}$$

By integrating Planck's Law over the spectrum, the Stefan-Boltzmann law is found:

$$\int_0^\infty L_\lambda^\circ(T)\mathrm{d}\lambda = \sigma T^4 \tag{2.5}$$

Fig. 2.3 shows the Planck's law as a function of wavelength, for different temperatures values. The maximum of this distribution is shifted as the temperature changes. This is known as the displacement
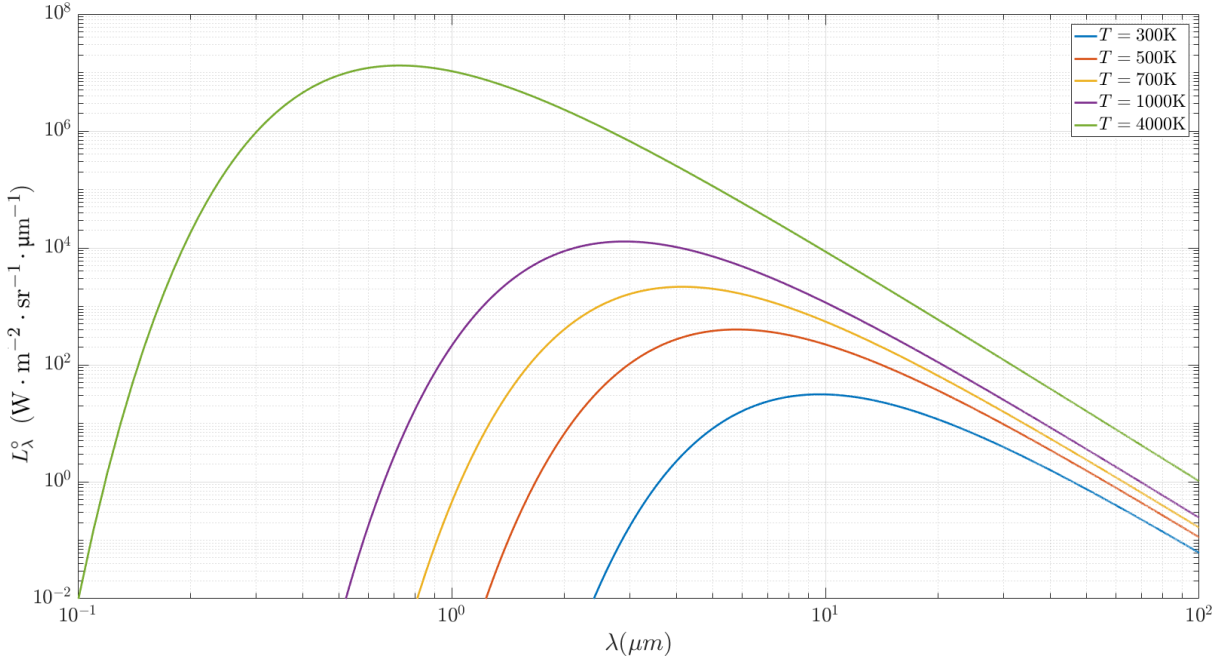
Figure 2.3: Planck's law as a function of wavelength at different temperature values

law of Wien: the wavelength that maximizes the emission relies on the temperature but the product $\lambda_{max}(T)T \approx 2.898 \times 10^{-3}mK$ remains constant. Such property explains why an incandescent bulb filament emits in the red at about $700°C$ and becomes white and then blue for higher temperatures.

At the same time, two other relations linking the radiative energy emitted by a black body were found the Rayleigh-Jeans formula and the Wien's approximation law. Both of those formulae can be retrieved by Planck's law and may be used for particular applications as an approximation of Planck's law.

Definition 2.2.3 (Rayleigh-Jeans formula)**.** By expanding the exponential of the denominator of Eq. (2.21) in a series:

$$e^{\frac{hc}{\lambda kT}} = \sum_{n=0}^{\infty} \frac{(\frac{hc}{\lambda kT})^n}{n!}, \quad \forall \frac{hc}{\lambda kT} > 0 \text{ (always the case)} \tag{2.6}$$

For $\lambda T >> \frac{hc}{k}$, we can approximate the denominator by using the development until $n = 1$ which gives:

$$L_\lambda^\circ(T) = \frac{2ckT}{\lambda^4} \tag{2.7}$$

Definition 2.2.4 (Wien's approximation law)**.** This was the model originaly proposed by Wien to describe the complete spectrum of thermal radiation until Planck's one.

Wien's approximation can now be derived from Planck's law when $h\nu \gg kT$:

$$L_\lambda^\circ(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \approx \frac{2hc^2}{\lambda^5} e^{\frac{-hc}{\lambda k_B T}} \qquad (2.8)$$

The temperature of a black body can therefore be deducted from Planck's law:

$$T = \frac{C_2}{\ln\left(\frac{(C_1 + \lambda^5 M_\lambda^\circ)^\lambda}{(\lambda^5 M_\lambda^\circ)^\lambda}\right)} \qquad (2.9)$$
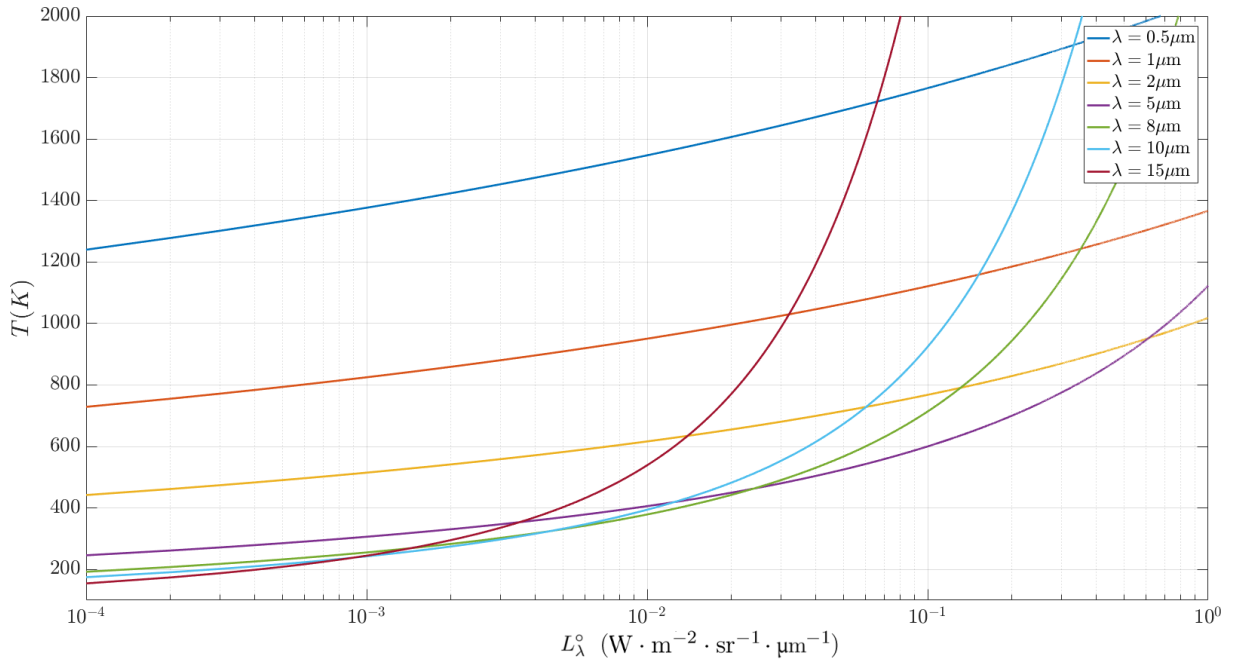
Figure 2.4: Temperature as a function of exitance at different wavelengths.

Fig. 2.4 shows the temperature value as a function of radiant exitance, for different wavelength values. This figure shows two main trends regarding the temperature and exitance dependence. In the MWIR-LWIR bands, highest energy levels correspond to a more important increase of the temperature than in the NIR-SWIR bands.

In the next section, the main quantities of the radiative transfer theory will be defined in order to derive - at the end - the radiative transfer equations and get a better understanding of this physical phenomenon.

> **Note**
>
> Wien's approximation is often used to approximate Planck's law and particularly to linearize the equation system as shown on the emissivity - temperature section. This approximation can be justified in common civil engineering applications: since $\frac{hc}{k} = 14388\mu m \cdot K$, in the LWIR band ($\lambda \in [8\mu m; 14\mu m]$), we must have $959.2 >> T$.

2.2.3 Radiative transfer quantities definitions

Radiative transfer represents exchanges between surfaces and inside medium. Due to the complexity of the phenomenon it is important to keep in mind that the radiative transfer deals with waves that propagate among surfaces and through a medium. Therefore, the surfaces particularities and the geometry between the surfaces will have an effect on this heat transfer. To describe those exchanges, it is necessary to remind the solid angle definition.

Definition 2.2.5 (Solid angle). The solid angle is an equivalent of the planar angle but in three-dimensional space. It represents a part of $3D$ space delimited by a cone (the vertex of the cone is the vertex of the solid angle). It is a measure of the field of view from a particle point that an object covers and therefore used in radiometry to determine the exchanges between two bodies. The solid angle is defined as the ratio of its base area to the square of chord length. Its dimension is the steradian sr:

$$\mathrm{d}\Omega = \frac{A}{r^2} \tag{2.10}$$

The solid angle can also be defined on an unit hemisphere, which will be used later on (see Fig. 2.5):

$$\mathrm{d}\Omega = \frac{r\mathrm{d}\theta r \sin(\theta)\mathrm{d}\Phi}{r^2} = \sin\theta\mathrm{d}\theta\mathrm{d}\Phi \tag{2.11}$$

In radiometry, the quantities are defined for a given solid angle, to represent the beam direction. However, for approximation and simplicity needs or sometimes due to uniformity, the quantities may be referred to as *hemispherical.* In that case, it means that the quantity is integrated on the unit hemisphere over the surface. Hemispherical properties get rid of the angle component and propose a more global representation of the quantity, as an average over the surface.

Definition 2.2.6 (Radiance). The analysis of radiation field deals with the analysis of the amount of energy $\mathrm{d}Q_\lambda(x, \Omega, t)$ that is transported across an element of area $\mathrm{d}A$ [26]. The radiation will then be highly dependent on the spatial geometry and the location of this area element. Let $\mathfrak{D} \subset \mathbb{R}^d$ ($d \in \{2, 3\}$) and A be the unit sphere of $\mathbb{R}^d$. This energy is defined by unit of projected surface, $\mathrm{d}A\cos(\theta)$
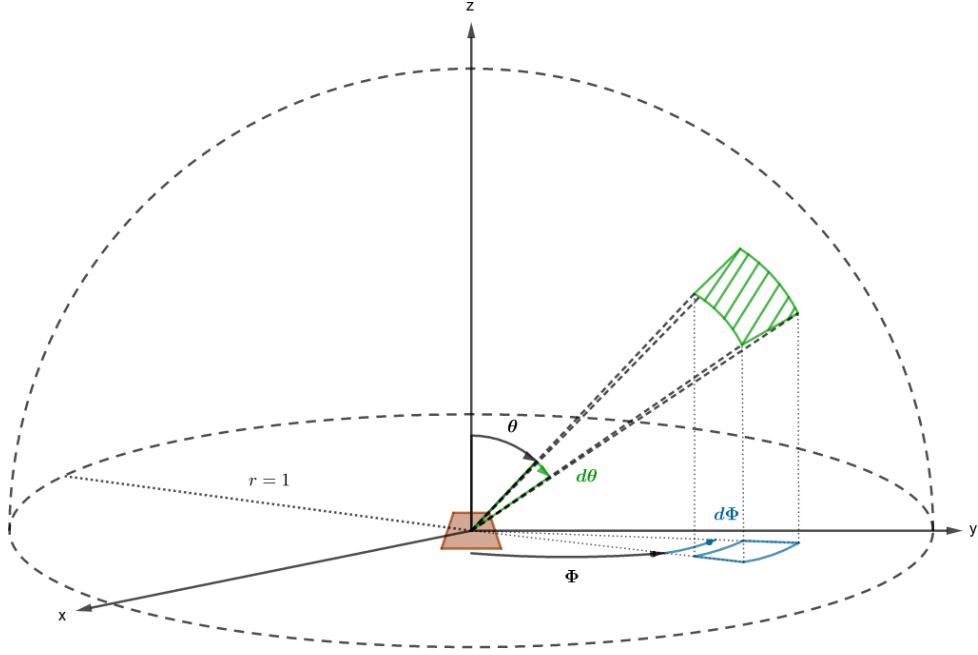
Figure 2.5: Solid angle on hemisphere. Note that the solid angle is not necessary a circular cone

to be located at x in time $\mathrm{d}t$, in the solid angle $\mathrm{d}\Omega$ about the direction $\vec{n}$ in the wavelength interval $[\lambda; \lambda + \mathrm{d}\lambda]$:

$$\mathrm{d}Q_\lambda(x, \Omega, t) = L_\lambda(x, \Omega, t) \cos\theta \mathrm{d}\lambda \mathrm{d}A \mathrm{d}\Omega \mathrm{d}t \qquad (2.12)$$

where θ is the angle which the direction considered makes the outward normal to $\mathrm{d}\vec{n}$. The quantity $L_\lambda(x, \Omega, t)$ is called spectral radiance ($\mathrm{W} \cdot \mathrm{sr}^{-1} \cdot \mathrm{m}^{-2} \cdot \mathrm{\mu m}^{-1}$) We have to note the fact that $L_\lambda(x, \Omega, t)$ depends on seven variables in total:

- 3 variables giving the position in space x,
- 2 variables giving the direction of the radiation Ω, in which we use the *inclination* angle θ and the *azimuth* angle φ,
- The time variable t,
- The wavelength λ or sometimes the energy $h\nu$ or the frequency $\nu = \frac{c}{\lambda}$.

This dimensionality makes the solving of the general radiative transfer equation difficult both analytically and numerically. Moreover, most of the time, the radiation has an impact on the crossed material, which also has an impact on the incomming radiation. Therefore, the resolution of the problem must be coupled, iteratively.

For simplification in the writting, and regarding the amount of dependent variables that $L_\lambda(x, \Omega, t)$

11

depends on, we may abusively write [1]:

$$L_\lambda \equiv L_\lambda(x, \Omega, t) \tag{2.13}$$

Definition 2.2.7 (Radiant flux). Radiant flux is the radiant energy emitted, reflected, transmitted or received, per unit of time (W).

$$\phi = \frac{\partial Q}{\partial t} \tag{2.14}$$

where Q is the radiant energy emitted, reflected, transmitted or received (J). From Eq. (2.1) we can define the spectral flux in frequency $\phi_\nu = \frac{\partial \phi}{\partial \nu}$ and the spectral flux in wavelength $\phi_\lambda = \frac{\partial \phi}{\partial \lambda}$ [1].

Finally, one can combine Eq. (2.4) and Eq. (2.1) to get the spectral radiance in terms of flux which represents how much of the power is emitted, reflected, transmitted or received by a surface when an optical system is looking at the surface from a specified angle of view as unit.

$$L_\lambda = \frac{\partial^2 \phi_\lambda}{\cos(\theta) \partial \Omega \partial A} \tag{2.15}$$

where:

- ϕ_λ is the radiant spectral flux (emitted, reflected, transmitted or received),
- Ω is the solid angle,
- $\partial A \cos(\theta)$ is the projected surface.

Definition 2.2.8 (Irradiance). The irradiance denoted E is defined as the local value of the ratio of the flux ϕ_r received by an object and the area of this object. It is the power received per unit area $(\mathrm{W} \cdot \mathrm{m}^{-2})$:

$$E = \frac{\partial \phi_r}{\partial A} \tag{2.16}$$

Definition 2.2.9 (Exitance). The radiant exitance denoted M is defined as the local value of the ratio of the flux ϕ_e emitted by an object and the area of this object. It is the power emitted per unit area $(\mathrm{W} \cdot \mathrm{m}^{-2})$:

$$M = \frac{\partial \phi_e}{\partial A} \tag{2.17}$$

Definition 2.2.10 (Radiosity). The radiosity of a surface denoted J is the radiant flux leaving (emitted, reflected and transmitted by) a surface per unit area. It is the power left per unit area $(\mathrm{W} \cdot \mathrm{m}^{-2})$. Therefore:

$$J = \frac{\partial \phi}{\partial A} = M + J_{\text{reflected}} + J_{\text{transmitted}} \tag{2.18}$$

When the surface is opaque, $J_{\text{transmitted}} = 0 \implies J = M + J_{\text{reflected}}$.

---

[1] The same notations are applied to the other quantities.

Definition 2.2.11 (Radiant intensity). The radiant intensity is the ratio of the flux ϕ_e emitted per unit solid angle $\mathrm{d}\Omega$:

$$I = \frac{\partial \phi_e}{\partial \Omega} \tag{2.19}$$

As explained in [93], the intensity of radiation from a black-body is defined on the basis of the normal area. It means that I is independent of the direction of emission, by definition which enables the description of a black-body intensity without defining the normal of the surface or the angles (θ, Φ).

Definition 2.2.12 (Directional emissive power). The radiant intensity is defined by its normal which means that the maximum energy emitted by a surface is reached if the surface is normal to the receiver. When there is an angle between the two surfaces, then the energy reaching the receiver is not the radiant intensity anymore. By assuming that the emission is uniform in all directions, the directional spectral emissive power for a black surface can be derived:

$$\tilde{I}(\theta, \Phi) = I \cos(\theta) = \tilde{I}(\theta) \tag{2.20}$$

where $\tilde{I}$ stands for the directional emissive power. This equations is valid for Lambertian surfaces which represent a perfect diffusion: the brightness of the surface appears the same no matter the observer's angle of view (Fig. 2.6). The emission is therefore isotropic and the intensity follows Eq. (2.20) called **Lambert's cosine law**.
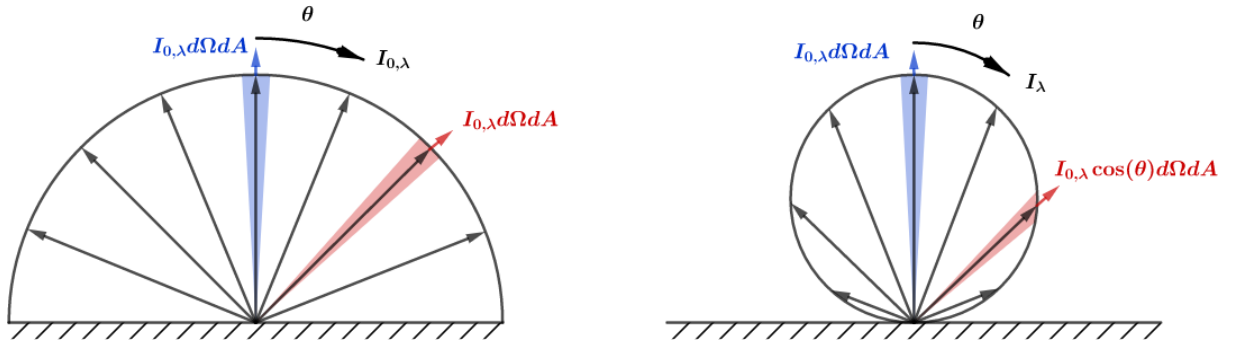
Figure 2.6: Emission rate in a normal direction and off direction with the Lambert's cosine law on the right.

Based on this law, one can integrate the emission of a black-body through a unit hemisphere over the surface. The spectral emission from an infinitesimal area $\mathrm{d}A$ per unit of time and unit surface area passing through the element on the hemisphere is given by:

$$\tilde{I}_\lambda \cos(\theta)\mathrm{d}\Omega\mathrm{d}\lambda = \tilde{I}_\lambda \cos\theta \sin\theta\mathrm{d}\theta\mathrm{d}\Phi\mathrm{d}\lambda \tag{2.21}$$

By integrating over the hemisphere and using Eq. (2.20)

$$\tilde{I}_\lambda \mathrm{d}\lambda = I_\lambda \mathrm{d}\lambda \int_{\Phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} \cos\theta \sin\theta \mathrm{d}\theta \mathrm{d}\Phi = \pi I_\lambda \mathrm{d}\lambda \tag{2.22}$$

The black-body hemispherical emissive power is therefore π times the black-body intensity.

We also have the equivalent in luminance. For a flux leaving the surface in an isotropic way:

$$M_\lambda = \pi L_\lambda \tag{2.23}$$

Definition 2.2.13 (Bouguer's law). Finally, Bouguer's law defines the relation between the irradiance E of a receiving surface, due to a source S and the intensity of that source in the direction of the receiver lying at a distance d. In the case of a nonparticipating media.

$$E = \frac{I \cos(\theta)}{d^2} \tag{2.24}$$

The Lambert's cosine law can be recognized in this expression as well as the inverse square law.

Now that the main quantities that describe the radiation theory have been introduced, the next section aims at developing the principles behind the actual phenomena that occur at a given surface.

2.2.4 Surface properties

Until now, the black body has been said to be a body that emits the maximum of energy for a given temperature. To be more accurate, a black body is an idealized physical body that absorbs all incident electromagnetic radiation, regardless of the frequency or the angle of incidence [144]. At the opposite of a black body, a white body will reflect all incident rays uniformly and entirely in all directions. Whereas black body represents an idealized object, real materials are considered to emit energy at a fraction of a black body energy levels. The ratio that links the two materials is called emissivity (ϵ, more rigorously defined later on). Finally, surfaces that have their properties independent of wavelength are called grey surfaces. When dealing with real radiating materials, multiple interactions at interfaces may occur and are schematized in Fig. 2.7. The emission has been defined previously for a black body, as well as the incident radiation. However, the reflectivity, absorption and transmission have not been mentioned yet and are addressed in this section.

To understand well the complexity of those phenomena, it is necessary to view those quantities as local properties of the material. As for the previous quantities, they will depend on the solid angle, the wavelength and the temperature but also on the intrinsic characteristics of the surface's material. Wood and metal will not absorb or reflect the same way; as well as polished iron will not have the same properties as unpolished one. From a practical point of view, giving values to those quantities is difficult. In fact, when the reflection or the absorption of a given material is given in the literature

(usually the hemispherical one), it cannot take into account the local roughness of the material or its detailed atomic composition. However, some analytic results exist for specific materials, in particular for dielectric ones. Determining the material's properties represents a field on its own.
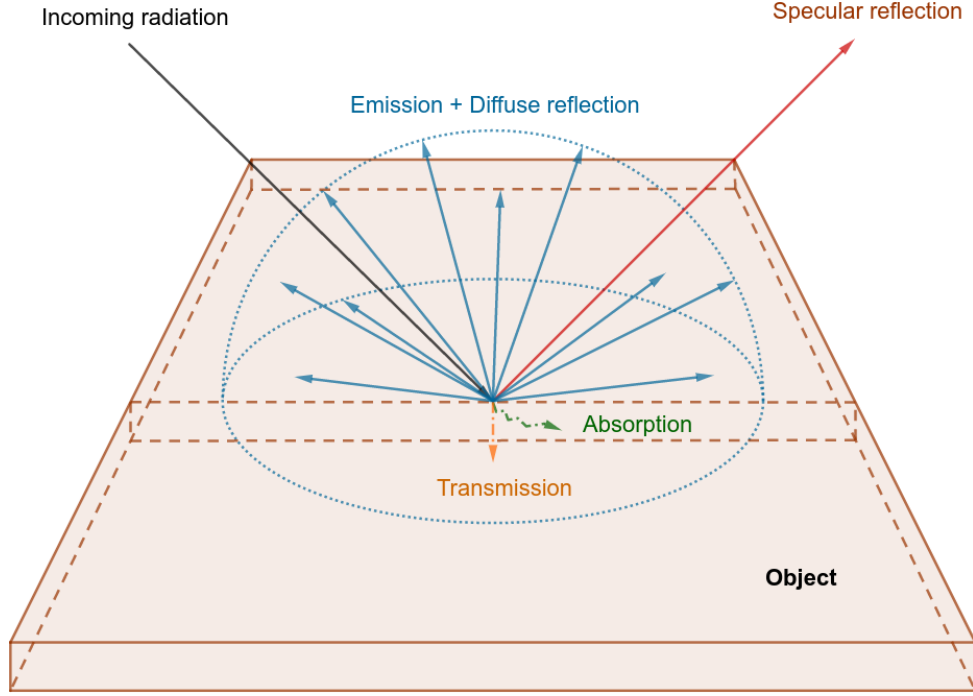
Figure 2.7: Interactions that occur at the interface of an object

Definition 2.2.14 (Reflectivity and transmissivity). The reflectivity represents the effectiveness of a surface of a material in reflecting incoming radiant energy. If the specular reflectance can be known for particular materials such as dielectric ones, derived from the electromagnetic theory, analytical formulations are most of the time unknowns for other materials. The reflectivity can be defined as a bidirectional function and denoted by $\rho_\lambda(\theta_1, \phi_1, \theta_2, \phi_2)$. In that case, it means that the reflected beam depends on the incoming beam solid angle.

The directional hemispherical spectral reflectivity can be derived from this definition:

$$\rho_\lambda(\theta, \phi) = \frac{\mathrm{d}^2 Q_\lambda, r(\theta, \phi)}{\mathrm{d}^2 Q_\lambda, i(\theta, \phi)} = \int \rho_\lambda(\theta_r, \phi_r, \theta, \phi) \cos(\theta_r) \mathrm{d}\Omega \qquad (2.25)$$

Similarly, the transmissivity can be defined as a bidirectional spectra quantity $\tau_\lambda(\theta_1, \phi_1, \theta_2, \phi_2)$. For an opaque material, $\tau_\lambda(\theta_1, \phi_1, \theta_2, \phi_2) = 0$.

Definition 2.2.15 (Emissivity). The emissivity represents the effectiveness of a surface of a material in emitting energy as thermal radiation. Based on the Planck's law and as explained previously, the directional spectral emissivity $\epsilon_\lambda(\theta, \phi, T)$ can be defined mathematically as the ratio of the radiance

$L_\lambda(T)$ emitted by a body at temperature T to the radiance $L_\lambda^\circ(T)$ emitted by a black body at the same temperature T:

$$\epsilon_\lambda(\theta, \phi, T) = \frac{L_\lambda(T)}{L_\lambda^\circ(T)} \tag{2.26}$$

Definition 2.2.16 (Absorptivity). The absorptivity represents the part of the incoming energy that is absorbed by the material. It is denoted as $\alpha_\lambda(\theta, \phi, T)$ and is strongly related to absorbing surface's temperature property. If the incident surrounding of the surface is a black body at temperature T° and the total absorbed energy is $Q_{\lambda,a}(\theta, \phi, T)$, the absorptivity can be written as:

$$\alpha_\lambda(\theta, \phi, T) = \frac{\mathrm{d}^2 Q_{\lambda,a}(\theta, \phi, T)\mathrm{d}\lambda}{I_\lambda^\circ(T^\circ)\mathrm{d}A\cos(\theta)\mathrm{d}\Omega\mathrm{d}\lambda} \tag{2.27}$$

Definition 2.2.17 (Kirchhoff's law). Kirchhoff's law links the absorptivity and the emissivity capacities of a given body. It actually depends on which of the spectral, directional or hemispherical quantities are considered. A general statement is that at thermal equilibrium, for a given body in a radiative enclosure, the body needs to return the same amount of energy its has absorbed to the enclosure in order to keep the thermal equilibrium.

By using the definition of the emissivity, the energy emitted per unit time by an element $\mathrm{d}A$ in a wavelength interval $\mathrm{d}\lambda$ and solid angle $\mathrm{d}\Omega$ is

$$\mathrm{d}^2 Q_{\lambda,e}(\theta, \phi, T)\mathrm{d}\lambda = \epsilon_\lambda(\theta, \phi, T)I_\lambda^\circ(T)\mathrm{d}A\cos(\theta)\mathrm{d}\Omega\mathrm{d}\lambda \tag{2.28}$$

To ensure thermal equilibrium and isotropy within the black enclosure:

$$\mathrm{d}^2 Q_{\lambda,e}(\theta, \phi, T)\mathrm{d}\lambda = \mathrm{d}^2 Q_{\lambda,a}(\theta, \phi, T)\mathrm{d}\lambda \tag{2.29}$$

which leads by using the definition of emissivity and absorptivity to:

$$\alpha_\lambda(\theta, \phi, T) = \epsilon_\lambda(\theta, \phi, T) \tag{2.30}$$

Please note that when considering the polarization of the radiation, this equality holds only for the components that have the same polarization.

> **Note**
>
> For an opaque material, we can write the directional spectral emissivity and reflectivitiy $\epsilon_\lambda(\theta, \phi, T) = 1 - \rho_\lambda(\theta, \phi, T)$.

The emissivity is one of the most important parameters in the physical process of radiation heat transfer. The emissivity depends on the view angle, wavelength, temperature and intrinsic characteristics of the material. Fig. 2.8 shows some examples of different materials' emissivity profiles.
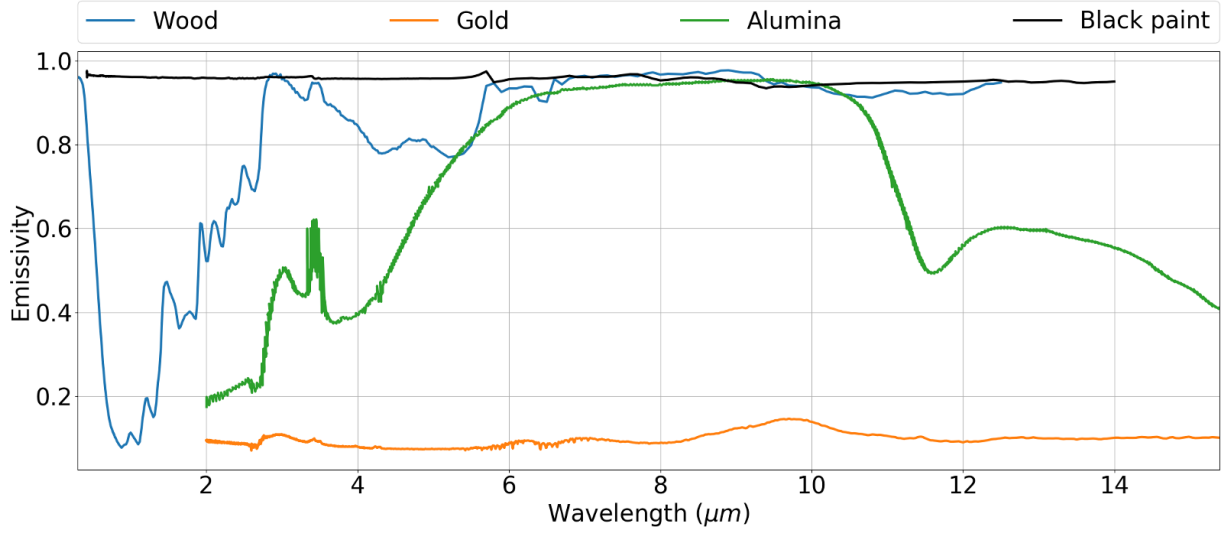
Figure 2.8: Spectral emissivity of a series of various materials. (Data from reflectance spectra in ASTER spectral library `http://speclib.jpl.nasa.gov`, Copyright: Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA; see also Balridge, A.M. et al., Remote Sens. Environ., 113, 711, 2009.). The alumina is from previous studies [130].

Most of the time, the emissivity of a material is given on a particular spectral band and integrated over the directions (called hemispherical emissivity). The material is then considered optically thick. As a consequence, emissivity values found in the literature must be taken with care. In fact, the variation of the emissivity of a surface with the observation's angle means that surfaces that are not flat will have a locally varying apparent emissivity, even for a given material. The roughness of a material will therefore impact the emissivity value, leading to important uncertainties (see Fig. 2.9).

2.3 Radiative Transfer Equation (RTE)

2.3.1 Differential form

The RTE represents what affects a beam radiation as it travels. Three main effects can occur:

- Absorption ⇒ Energy is lost
- Emission ⇒ Energy is gained
- Scattering ⇒ Energy is redistributed

We will see in that section how to consider those different effects and how it affects the final equation.

We can construct the radiative transfer equation by summing the different effects that will affect
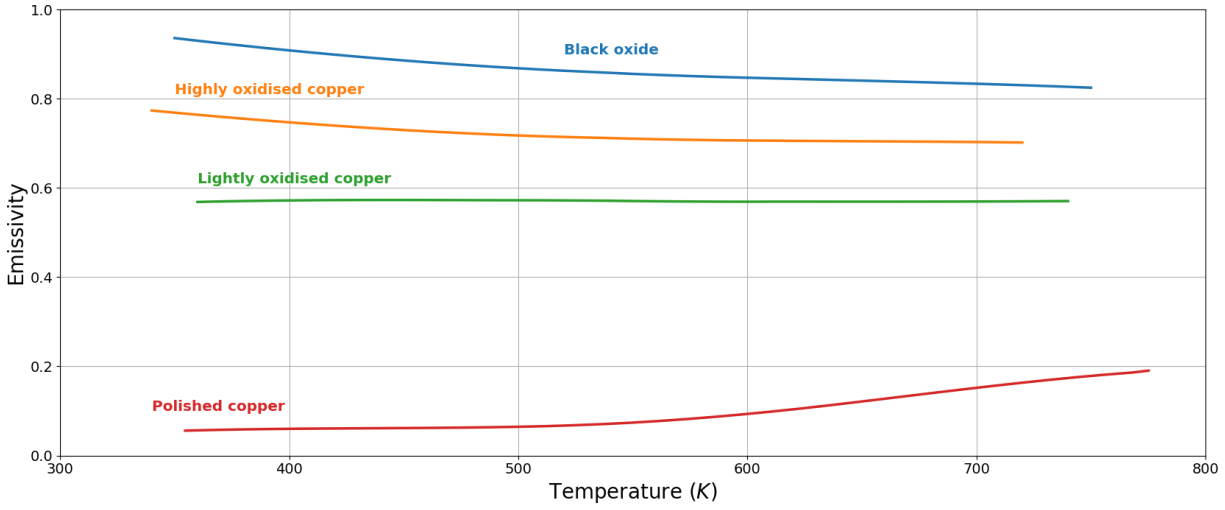
Figure 2.9: Effect of the state of the surface on emissivity (from [71])

the radiation [93]:

$$
\begin{aligned}
\text{Change in radiative energy} =& \text{Gain due to emission} - \text{Loss due to absorption} \\
& - \text{Loss due to out-scattering} + \text{Gain due to in-scattering}
\end{aligned}
\tag{2.31}
$$

This physical phenomenon can be formulated mathematically through the differential form of the radiative transfer equation:

$$
\frac{1}{c}\frac{\partial L_\nu}{\partial t} + \hat{\Omega}.\nabla L_\nu = j_\nu - (\kappa_{\nu,s} + \kappa_{\nu,a})L_\nu + \frac{1}{4\pi}\int_\Omega \kappa_{\nu,s}L_\nu \mathrm{d}\Omega
\tag{2.32}
$$

where:
- c is the speed of light,
- L_ν is the spectral radiance,
- $\hat{\Omega}$ is the propagation direction,
- $k_{\nu,a}$ is the absorption coefficient,
- $k_{\nu,s}$ is the scattering coefficient,
- j_ν is the emission coefficient,
- $\frac{\kappa_\nu}{4\pi}\int_\Omega L_\nu \mathrm{d}\Omega$ is the radiation scattered from other directions onto the surface.

Most of the time the term $\frac{1}{c}\frac{\partial L_\nu}{\partial t}$ is neglected since it would involve fast responsive systems compared to the involved distances. [2] More details on this equation can be found in [93]. Analytical solutions of the RTE exist only under some assumptions and numerical ones based on finite elements

---

[2]Fast here means systems for which the variation of L_ν in time is comparable to the light of speed. For most of the systems, we have $\frac{1}{c} \ll \frac{\partial L_\nu}{\partial t}$ which justifies the use of this approximation. However, when the distances become bigger like in astronomy for example, this approximation should be re-considered.

models are more often used. In the thermography field, this equation is replaced by a simplified radiometric equation, often partially integrated in space that represents the radiative exchanges at a more macroscopic level, presented in section Section 2.4.1.

2.3.2 Radiative transfer in nonparticipating medium

The previous equation is suited for participating mediums. However, obtaining a solution for such equation is non trivial especially in 3-dimensions. This equation is often simplified and used for solving complex systems through finite elements methods by using physical approximations. Eq. (2.32) describes a change in radiative energy during the propagation of a beam of radiative energy in a participating medium. However, for a numerous number of systems, the scattering from a participating medium can be neglected without introducing any significant error to calculations. When it is the case and that there is no scattering particles involved, the RTE is reduced to a partial differential equation.

Exchange between two surfaces

Let now consider a simpler case of nonparticipating (*i.e.* no scattering, no absorption and no emission) medium. Instead of looking at a single beam propagation phenomenon, let make the balance on a given surface since the medium is nonparticipating. Let consider two infinitesimal surfaces of area dA_1 and dA_2 and separated by a distance r. In this simplified case, the infinitesimal surfaces emit uniformly in all directions and reflect the energy received from other surfaces. Properties of emission and reflections will be more described later on. The energy $dQ_{\lambda,d1}$ leaving dA_1 in direction x_1 is given by:

$$dQ_{\lambda,d1} = \tilde{I}_{\lambda,d1} \cos\theta_1 d\lambda dA_1 d\Omega_1 \tag{2.33}$$

The solid angle can be written by:

$$d\Omega_1 = \frac{\cos(\theta_2)}{r^2} dA_2 \tag{2.34}$$

where $dA_2 \cos(\theta_2)$ is the projection of dA_2 on the direction x_1. When no absorption is considered in the medium, the radiative energy reaching dA_2 is therefore given by:

$$dQ_{\lambda,d1 \rightarrow d2} = \tilde{I}_{\lambda,d1} \left(\frac{\cos(\theta_1)\cos(\theta_2)}{r^2} \right) d\lambda dA_1 dA_2 \tag{2.35}$$

When Eq. (2.32) was integrated on the direction, Eq. (2.34) is now integrated on the area only. For finite element surface, the integration yields to:

$$Q_{\lambda,1 \rightarrow 2} = \int\int_{A_1,A_2} dQ_{\lambda,d1 \rightarrow d2} = \int\int_{A_1,A_2} \tilde{I}_{\lambda,d1} \frac{\cos(\theta_1)\cos(\theta_2)}{r^2} d\lambda dA_1 dA_2 \tag{2.36}$$

19

By considering the previous stated assumptions that the radiation leaving A_1 is directionally uniform, $\tilde{I}_{\lambda,d1} = \tilde{I}_{\lambda,1} = I_{\lambda,1}\pi$ and can be taken outside the integral:

$$
\begin{aligned}
Q_{\lambda,1\to 2} &= \tilde{I}_{\lambda,1} \int \int_{A_1,A_2} \frac{\cos(\theta_1)\cos(\theta_2)}{r^2} \mathrm{d}\lambda \mathrm{d}A_1 \mathrm{d}A_2 \\
&= I_{\lambda,1} \left(\int \int_{A_1,A_2} \frac{\cos(\theta_1)\cos(\theta_2)}{\pi r^2} \mathrm{d}A_1 \mathrm{d}A_2 \right) \mathrm{d}\lambda
\end{aligned}
\tag{2.37}
$$

The term inside the parenthesis is called a form factor (sometimes configuration or view factor) and represents the geometrical exchanges interaction coefficient between the two surfaces A_1 and A_2. It depends only on the geometry of the scene and is denoted $F_{1\to 2}$:

$$
Q_{\lambda,1\to 2} = I_{\lambda,1} A_1 F_{1\to 2} \mathrm{d}\lambda
\tag{2.38}
$$

where

$$
A_1 F_{1\to 2} = \int \int_{A_1,A_2} \frac{\cos(\theta_1)\cos(\theta_2)}{\pi r^2} \mathrm{d}A_1 \mathrm{d}A_2
\tag{2.39}
$$

Radiosity equation

Let now consider an enclosure $S \in \mathbb{R}^3$ with a nonparticipating medium inside and opaque surface. Let x be a parametric point of such enclosure and an infinitesimal surface $\mathrm{d}A$ (see Fig. 2.10). Based on its definition, the radiance at a given point x of the enclosure and given time can be derived as:

$$
L_{\lambda,\text{leaving}}(x,\Omega) = L_{\lambda,\text{emitted}}(x,\Omega) + L_{\lambda,\text{reflected}}(x,\Omega)
\tag{2.40}
$$

The reflected term $L_{\lambda,\text{reflected}}(x)$ is the radiation part that is received at x (by all the other radiated parts) and reflected by x. Based on the definition of the bidirectional reflectivity and Eq. (2.37), this term can be written as a sum of the energy received from all other directions.

$$
L_{\lambda,\text{reflected}}(x,\Omega) = \int_{(x',\Omega')\in S,\cos(\theta)>0} \rho_\lambda(x,\Omega,\Omega')V(x,x')L_{\lambda,\text{leaving}}(x',\Omega')\cos(\theta)\mathrm{d}\Omega'
\tag{2.41}
$$

where $\cos(\theta) = n_x.(x-x')/||x-x'||_2$ (Lambert's cosine law) and $V(x,x')$ represents a visibility factor which is equal to 1 if x sees x' and 0 otherwise, as illustrated in Fig. 2.10. By using the expression of $\mathrm{d}\Omega'$ as previously, the integral can be integrated with the areas and expressed through the form factors:

$$
L_{\lambda,\text{reflected}}(x,\Omega) = \int_{(x',\Omega')\in S,\cos(\theta)>0} \rho_\lambda(x,\Omega,\Omega')V(x,x')L_{\lambda,\text{leaving}}(x',\Omega')\frac{\cos(\theta)\cos(\theta')}{||x-x'||_2^2}\mathrm{d}A'
\tag{2.42}
$$
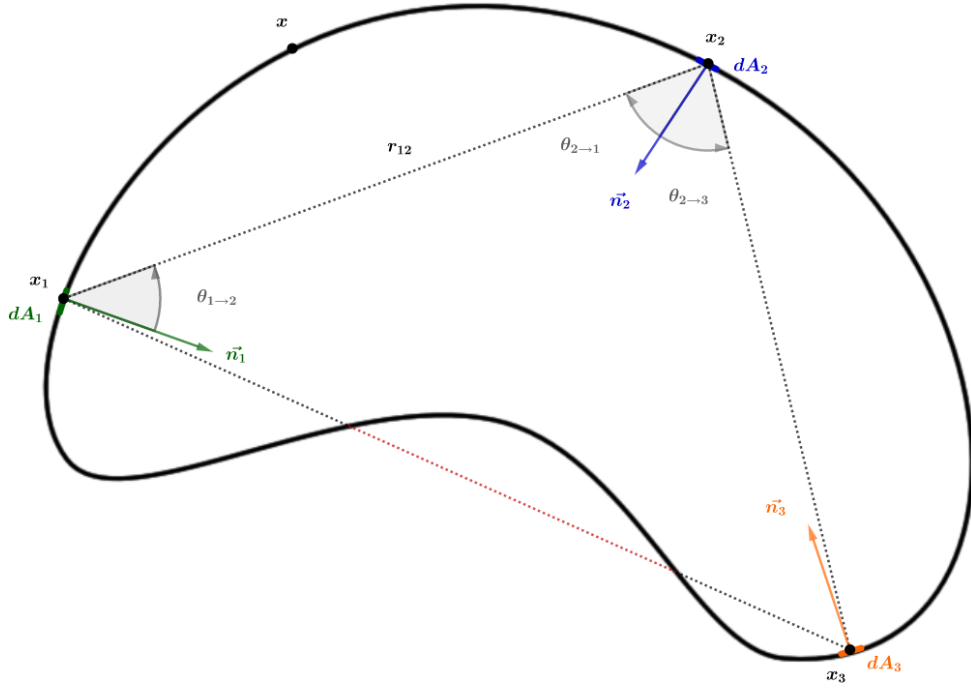
Figure 2.10: Enclosure for which the radiosity is computed at the surface. Point x_3 and x_2 cannot see them each other.

Finally, the radiance at x and solid angle Ω is:

$$L_{\lambda,\text{leaving}}(x,\Omega) = L_{\lambda,\text{emitted}}(x,\Omega) + \int_{(x',\Omega')\in S,\cos(\theta)>0} \rho_\lambda(x,\Omega,\Omega')F(x,x')L_{\lambda,leaving}(x',\Omega')\mathrm{d}A' \tag{2.43}$$

where

$$F(x,x') = V(x,x')\frac{\cos(\theta)\cos(\theta')}{||x-x'||_2^2} \tag{2.44}$$

Let suppose now that the surfaces are lambertians, and therefore does not depend on the direction. The Eq. (2.24) can be used for diffuse surfaces for which the reflectivity is independent of (θ_r,ϕ_r):

$$\rho_{\lambda,h} = \rho_\lambda(\theta,\phi)\int \cos(\theta_r)\mathrm{d}\Omega = \pi\rho_\lambda(\theta,\phi) \tag{2.45}$$

which is analogous to the Eq. (2.22). As pointed by [93], since $\rho_{\lambda,h} = 1 - \alpha_\lambda$, the reflectivity cannot depend on the angle of incidence:

$$\rho_{\lambda,h} = \pi\rho_\lambda \tag{2.46}$$

Therefore, for diffuse surfaces:

$$L_{\lambda,\text{leaving}}(x) = L_{\lambda,\text{emitted}}(x) + \frac{\rho_\lambda(x)}{\pi}\int_S F(x,x')L_{\lambda,leaving}(x')\mathrm{d}A' \tag{2.47}$$

Instead of dealing with punctual quantities, it is possible to integrate those quantities over the hemisphere:

$$\underbrace{\int_{\cap} L_{\lambda,\text{leaving}}(x)\cos(\theta)\mathrm{d}\Omega}_{J_\lambda(x)} = \underbrace{\int_{\cap} L_{\lambda,\text{emitted}}(x)\cos(\theta)\mathrm{d}\Omega}_{M_\lambda(x)}$$
$$+ \frac{\rho_\lambda(x)}{\pi}\int_S F(x,x')\underbrace{\int_{\cap}' L_{\lambda,leaving}(x')\cos(\theta')\mathrm{d}\Omega'\,\mathrm{d}A'}_{\int_S F(x,x')J_\lambda(x')\mathrm{d}A'} \tag{2.48}$$

$$J_\lambda(x) = M_\lambda(x) + \frac{\rho_\lambda(x)}{\pi}\int_S F(x,x')J_\lambda(x')\mathrm{d}A' \tag{2.49}$$

Eq. (2.48) is called the radiosity equation. Originally developed for solving heat transfer problems, it has been largely used in the image synthesis domain, for global illumination algorithms. The solving and discretization of this equation will be detailed in Chapter 3. This equation will be used as a forward model in order to simulate data in Chapter 5 which is a convenient way to test the efficiency and robustness of the developed algorithms.

2.4 Infrared thermography

2.4.1 Simplified radiometric equation

The main quantities involved in the radiation theory have been presented and two radiative transfer equations have been derived. In this part, a simplified model that governs the radiative exchanges for *in-situ* infrared thermography measurements will be exposed. This equation will rely on looking at the exchanges as fluxes on the scene instead of beams and focuses on the irradiance received at the camera's detectors. Thermography aims at determining the spatial distribution of heat in the observed scene as well as its time dependence. However, the actual radiation received at the sensors will not only depend on the observed object thermal properties but it will also depend on environmental conditions (*e.g.* weather), the thermo-optics and the geometrical properties of the measurement scene [27, 35, 183]. Therefore, the irradiance received at one sensor is the result of many radiometric interactions, different from scene to scene. Since most of the quantities derived in the previous sections are unknown when using an infrared camera, a simpler model is needed to be able to describe the quantity received at the sensor. The main interactions involved in such radiometric balance are represented in Fig. 2.11. To take into account the flux arriving at an infrared detector, a mathematical model is necessary and can be derived from a flux balance that reaches the detector [127].

Fig. 2.11 shows the different flux (ϕ emitted, Φ received) that are considered:

- ϕ_{obj}, the flux emitted by the object
- ϕ_{env}, the flux emitted by the ambient environment and reflected by the object

- ϕ_{atm}, the flux emitted by the atmosphere
- ϕ_{sun}, the sun flux contribution, reflected by the object
- ϕ_{opt}, the flux emitted by the optical system of the camera
- Φ_{tot}, the total flux received by the detectors
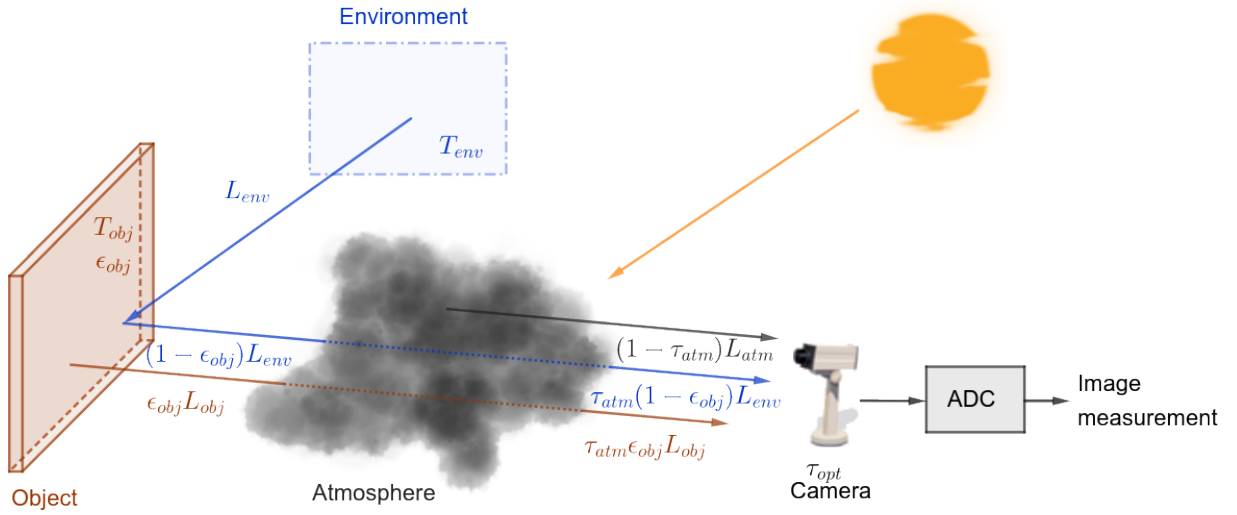
Figure 2.11: Main radiatives contributions received by the infrared camera

The total flux received at the sensor can then be considered as a sum of contribution from the others fluxes:

$$\Phi_{tot} = \phi_{obj} + \phi_{env} + \phi_{atm} + \phi_{sun} + \phi_{opt} \tag{2.50}$$

This radiative flux balance corresponds to the macroscopic exchanges between the different surfaces of the scene, without considering internal scattering. At the opposite, the Eq. (2.32) represents a local change in radiative energy for a given radiation beam. Doing a complete thermal and radiative balance would combine those two approaches, for which Eq. (2.49) would represent boundary conditions of Eq. (2.32). However, the simplified radiometric equation is used in thermography due to the approximation that the signal received at the sensors represents a combination of the emitted radiation of the observed surfaces.

Therefore, we can derive the simplified radiometric equation at pixel (i, j) for a given time [71]. In this equation, multiple approximations are implicitly done. First, the surfaces are considered as greybody in the considered band. Usually, surfaces are assumed to be diffuse (Lambertian hypothesis). The atmospheric, environment and sun contributions are assumed to be black-body equivalent radiative surfaces. The environmental contribution is considered to be an equivalent contribution of the sum of all the radiation that come from the surroundings. The sun contribution represents the radiance that arrives at the sensor, from the sun emittance that crosses the atmosphere both direct and indirect. The optical, atmospheric, sun and environmental contributions are supposed to be constant over the image

as a first approximation. The contribution from the scene can be expressed as:

$$
\begin{aligned}
L^{(i,j)}_{\Delta\lambda,\text{total}}(T) =& \tau_{opt} \left[\tau_{atm} \left(\epsilon^{(i,j)}_{\Delta\lambda,\text{obj}} L^{\circ(i,j)}_{\Delta\lambda,\text{obj}}(T^{(i,j)}_{obj}) + (1 - \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}})(L^{\circ}_{\Delta\lambda,env} + L^{\circ}_{\Delta\lambda,sun}) \right) \right. \\
& \left. + (1 - \tau_{atm})L^{\circ}_{\Delta\lambda,atm} \right] + (1 - \tau_{opt})L^{\circ}_{\Delta\lambda,opt}
\end{aligned}
\tag{2.51}
$$

Where τ_{atm} and τ_{opt} are respectively the atmospheric and optic transmissions.

From Eq. (2.50) we can find the radiative contribution of the observed object:

$$
\begin{aligned}
L^{\circ(i,j)}_{\Delta\lambda,\text{obj}}(T^{(i,j)}_{obj}) =& \frac{L^{(i,j)}_{\Delta\lambda,\text{total}}(T) - (1 - \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}})\tau_{atm}\tau_{opt}(L^{\circ}_{\Delta\lambda,\text{env}} + L^{\circ}_{\Delta\lambda,\text{sun}})}{\epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}\tau_{atm}\tau_{opt}} \\
& \frac{-(1 - \tau_{atm})\tau_{opt}L^{\circ}_{\Delta\lambda,\text{atm}} - (1 - \tau_{opt})L^{\circ}_{\Delta\lambda,opt}}{\epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}\tau_{atm}\tau_{opt}}
\end{aligned}
\tag{2.52}
$$

By using the contribution of the observed object and Planck's law, it is possible to find back the object's temperature. However, this can be achieved if and only if the other parameters of the model are well known. This is the topic of the following section.

2.4.2 Needs for emissivity-temperature simultaneous estimation methods

Once the contribution of our object has been derived, we still need to convert this quantity to temperature. We usually use a conversion model based on a thermal camera calibration to achieve this. Such step will be detailed in Chapter 4. Nevertheless, this equation shows how difficult it is to estimate the temperature from IR measurements without prior knowledge of the observed scene. In fact, even if the involved quantities can be determined through local weather reports and phenomenological laws with some errors (see Chapter 4), the system still needs to get the knowledge of the observed emissivity to retrieve the temperature from the measurements. Assuming that all the parameters are known except for the temperature and emissivity, the Fig. 2.12 shows the solution space for a given measurement value. Needs for methods that would estimate simultaneously the emissivity and the temperature can be emphasized from this figure. Without any knowledge about the emissivity, the temperature cannot be derived. Even more, an error made on the emissivity will lead to an error made on the temperature estimation.

Analytic values of spectral emissivities are often unknowns in practical applications. For an opaque material dielectric or metallic, typical directional and spectral emissivity can be derived from the Hagen-Rubens relation and based on the electro-magnetic theory. Another possibility is to obtain spectral emissivity values from databases. However, such approach is generally not enough for *in-situ* thermal monitoring of infrastructures to get an accurate estimate because of the lack of information about the observed material, the unknown roughness of the observed surface or even the angle of view with
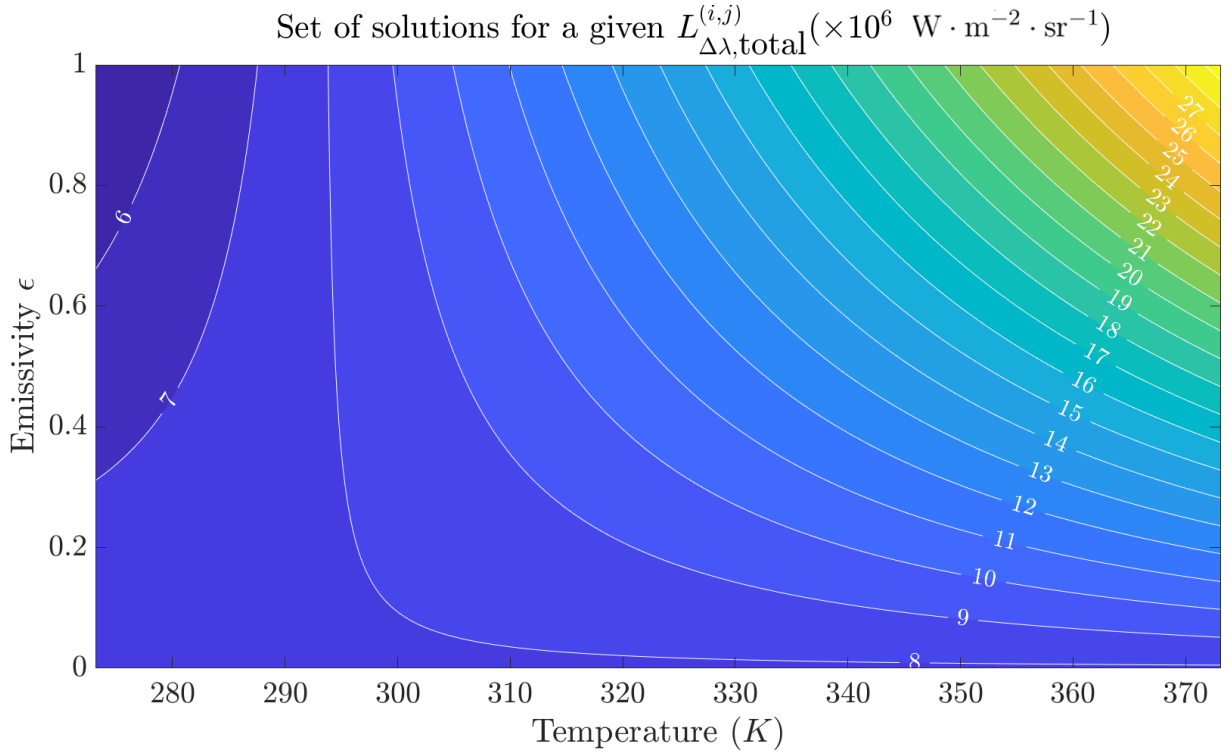
Figure 2.12: Possible solutions for a given measurement when everything but temperature and emissivity are known

the camera. Other possibilities involves local emissivity measurements. Such solution can be time-consuming for large infrastructures or even impossible in some cases for security purposes (*e.g.* remote recognition of weakening features of a rock cliff).

To overcome those issues, temperature-emissivity separation methods appeared in order to get rid of the emissivity in the thermal measurement procedure. Finally, with the development of Bayesian inference, new separation methods appeared in order to estimate conjointly the emissivity and the temperature from statistical *a priori*, having the potential to give an estimate with a known standard deviation error. In this thesis, different methods that separates emissivity and temperature that are used in the pyrometry or remote sensing fields are presented and compared to new methods involving spatial and temporal information usage.

The evaluation of emissivity and temperature can be performed through active methods, on which a surface is heated by a laser for example in the case of noncontact measurements. Those methods are often used for materials characterization, defects detection and are well-developed in those fields. However, active measurement for long-term infrared monitoring of infrastructure may not be possible for a couple of reasons. First of all, when dealing with large infrastructures, it may not be possible to perform a characterization on the entire structure. Such investigation will have an impact on the cost of the operation and may not be feasible for security purposes again. Furthermore, retrieving local

information may not be useful if the structure is observed in its entirety through a particular point of view. In this thesis, the focus is done on noncontact and passive temperature / emissivity retrieval methods for *in-situ* infrared thermal monitoring. This topic will have a dedicated bibliographical study in Chapter 3 and proposed method will be developed in Chapter 6

2.4.3   Measurements bias due to environmental and spatial conditions

The wide range of variables involved in the balance have to be identified and considered in any *in-situ* temperature estimation through IR thermography. We will review here the different impacts of the environment, the geometry of the scene and the digitization on our temperature estimate.

Spatial resolution and angle of view

The measurement set up will influence the obtained image. As we have seen previously, the emissivity depends on the material and its roughness but it also depends on the angle of view. As a consequence, the positioning and the angle of view of the camera on the scene will have an impact on the measurement (see Fig. 2.13b). Moreover, the perspective projection induced by the camera field of view will result on a non-constant spatial discretization, that will have to be considered in order to estimate the final temperature estimation error.

(a) Example of an IR image of a building   (b) Angle between camera and object
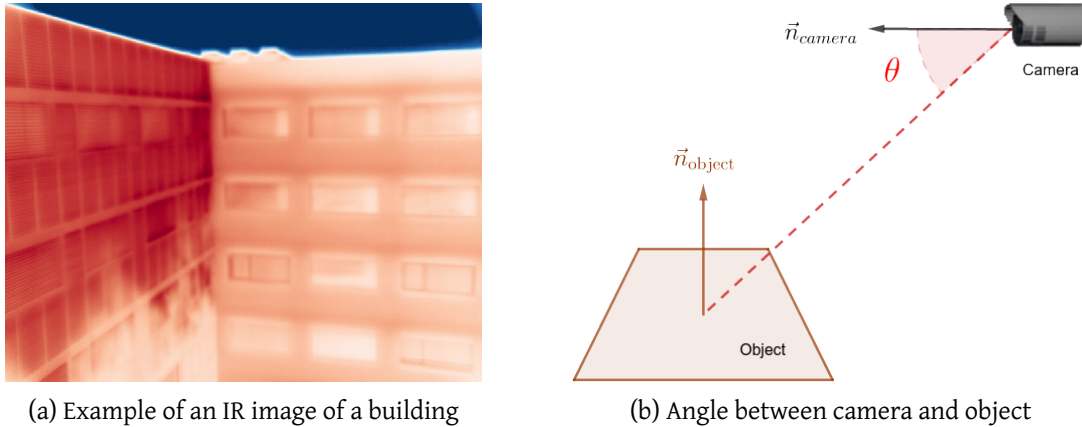
Figure 2.13: The geometry of the scene leads to different angle of view between the object and the camera in the image

Perspective projection   The image acquired by the IR camera will result in a projection of the 3D world coordinates to the 2D frame of the image. Therefore, the resulting image and projection will depend on the position of the camera onto the scene, the focal length, the aperture, the field of view *etc.* As said previously, the radiation measurement depends on the distance and angle of the sensor to the object. Most of the time, for *in-situ* civil engineering applications, large field of view are used to get a

full view of the structure and the complex geometry of those 3D structures leads to a wide variety of angles between the object and the camera view (see Fig. 2.13a).

Induced discretization   The position of the camera and its optical properties has to be considered in order to achieve a proper estimation of the temperature of the observed object. However, the angle of view is not the only geometrical quantity that will impact the measurement. The discretization and in particular the non-constant spatial sampling size will give a maximum accuracy reachable that will variate within the image, as represented in Fig. 2.14.
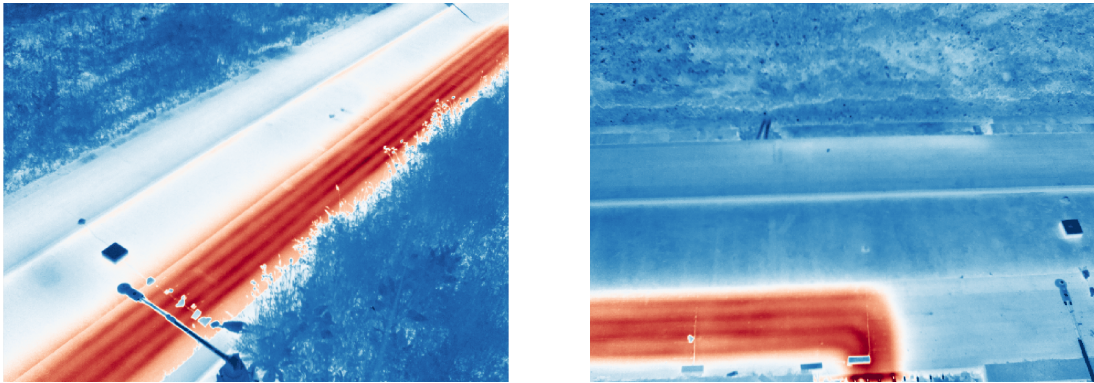
Figure 2.14: Different views of the same scene, the spatial sampling is not constant within the images

In Chapter 4, a spatial calibration procedure will be presented and the interpolation due to such resectioning will be discussed.

Mixed pixel effect   Last but not least, the discretization will lead to the so called "mixed pixel effect". In fact, one sensor may see multiple materials for which the measurement will lead to a mixture of those materials. Fig. 2.15 illustrates this effect. In a mixed pixel, the material property of the observed area can be considered as a combination of the properties of the mixed materials whereas in a pure pixel, we can consider only the material properties of the observed area. Therefore, most of the time, it is difficult to identify the emissivity of a particular pixel / material (especially when dealing with angles). As a consequence, the radiative properties are often estimated in terms of pixels *i.e.* an equivalent temperature and emissivity of the real observed surface.

2.4.4   Infrared cameras for thermal radiative measurements

As seen in the previous sections, the estimation of temperature from infrared measurements is a challenging task in which many factors are involved, from theoretical approximation in the radiative model to the actual measurements system and configuration. In order to understand the concept behind infrared thermography, IR cameras technologies principles are briefly introduced. Then a simple use-case measurement equation in single-band and multi-band thermography are derived.
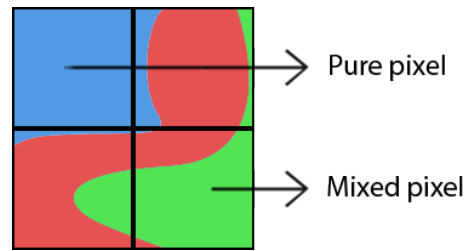
Figure 2.15: Mixed pixel effect illustration with three different materials.

History and basic principles

Since the discovery of infrared radiation by Herschel first experiment in 1800 with a prism and measuring the temperature of each color, needs for quantitative detection of radiation for many different application field, and in particular for Defense purposes appeared [147]. Over the past 80 years, many developments have been done in the infrared detectors which are used now in many fields, ranging from satellite defense applications to building insulation inspection. In particular, the latest improvements have lead to the development of small infrared uncooled cameras with the size of a smartphone camera: inexpensive and easy to deploy on site.

Because of the two bands of good transmission in the infrared range of the atmosphere, most detectors are divided into short-wave (SW - $1.4-3\mu m$) and long-wave (LW - $8-15\mu m$) camera devices. However, we can still find detectors that will work in the near IR (NIR - $0.75-1.4\mu m$) or even the far IR (FIR - $15-1000\mu m$)

Cameras can then be classified into two different groups: cooled or uncooled systems. Cooled systems have to maintain a low temperature at ranges that may vary between 4K to 203.15K depending on type and performance levels. At the opposite, uncooled systems, appeared more recently, operate at ambient temperature. Such systems are cheaper but measurements may not provide the same quality. We aimed at using our developed model on such devices, for their simplicity of exploitation and price.

Uncooled cameras   Uncooled cameras come with thermal detectors (microbolometer) that when heated through IR radiation, measure a temperature change. When an infrared radiation with wavelength between $8-15\mu m$ strikes the IR absorbing metal, it heats it, thus changing its electrical resistance. The change of the resistance is measured and processed through a CMOS (Complementary metaloxidesemiconductor) ROIC (readout integrated circuit) [189]. Additionally, a reflector can be put beneath the absorbing metal to send back to the absorption layer the energy that it did not absorbed previously, thus enhancing the maximum energy absorption. Since the system is uncooled, the absorbing material must be thermally isolated from the bottom. This is achieved by performing a sacrificial layer. This micro-bridge structure makes an optical resonant cavity possible at $\lambda/4$ (*i.e.* $2\,\mu m$ for the $8\,\mu m$ to $14\,\mu m$ spectral band) between the probe and the substrate. An example of a microbolometer structure is shown on Fig. 2.17.

Figure 2.16: Small uncooled IR camera with 3D-printed case. A coin of 1€ is placed at the top as a scale factor.
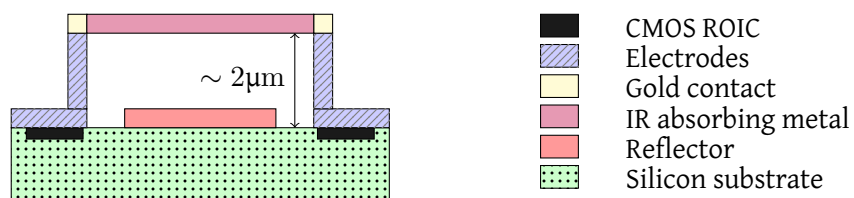
Figure 2.17: Cross-sectional view of a bolometer pixel structure principle

$\sim 2\mu m$

CMOS ROIC
Electrodes
Gold contact
IR absorbing metal
Reflector
Silicon substrate

The drift of such system may be important and the calibration is an essential part of the trust you can accord to the temperature measurement [120].

Main advantages	Main drawbacks
• Low power consumption • Lower cost • Moderate sensitivity • No mechanical cooling	• Fixed and long-time constants • Relatively slow frame rate • No control over timing or triggering • Sensitivity to ambient temperature $\implies$ needs for thermal calibration

Cooled cameras In order to count the number of photons, such cameras use quantum detectors that need to be cooled for cutting thermal noise. The materials are mainly semiconductors with narrow band gaps. Incoming photons will bring energy into those semiconductors causing excitation that will be measured. The necessity of cooling those detectors makes the manufacturing more complicated but the sensitivity of those cameras is much better than the uncooled systems.

Main advantages	Main drawbacks
• Very short integration times • Fast frame rates • Superior sensitivity • Ability to see and measure small targets	• High cost • Power consumption • Requires cryogenic cooling

Thermograms creation In order to create an image from an array of detectors, two main technologies exist. The first one is to use a single detector or a ruler of detectors and perform a scan of the observed area. In a single detector camera, the image is built point by point, in a ruler of detectors, it is constructed line by line (see Fig. 2.18). On a single point detector, one can note that all points of a thermogram have identical parameters because the temperature measured at every point is measured by the same detector. Such property can be useful in detecting the temperature difference at two points of a homogeneous object. Historically, the point scanner cameras was the first system used. Then, rulers of detectors and finally arrays of detectors were created. In the last systems, there is no mechanical scanning parts: the array observes the scene through the camera optics. Such system enables the creation of cameras that can record fast thermal processes such as 100 thermograms per second. Each pixel is read 25 times per second (PAL system - Europe) or 30 times (NTSC system - USA) [128].

(a) Point scanning   (b) Line scanning   (c) Array of detectors
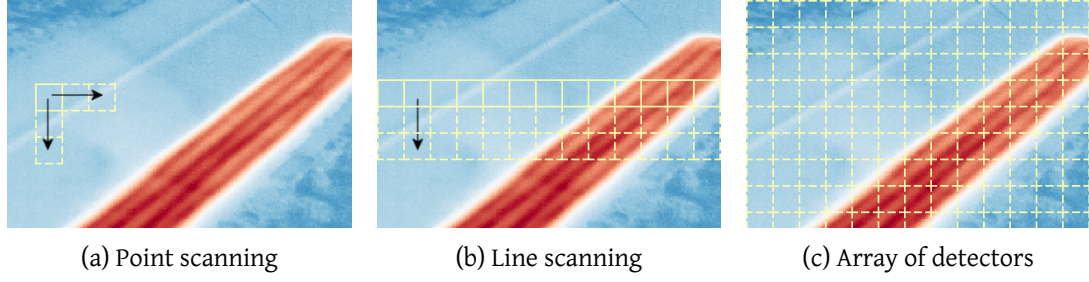
Figure 2.18: Different types of detectors reading

Single-band pyrometry

In order to emphasis again the needs for emissivity and temperature separation, let consider a simple case in which the irradiance at a sensor located at a distance r of an object is derived. Such case of study needs some assumptions:

> **Assumptions**
>
> - Flat surfaces are small enough to be considered punctuals.
> - Only the diffuse interaction between the sensor and the object is considered.
> - The scene geometry is known
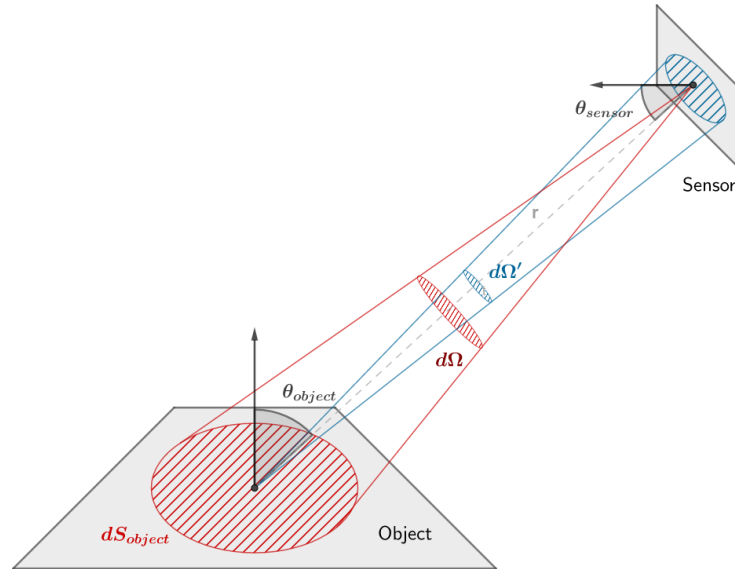
Figure 2.19: Sensor / Object configuration

Under those assumptions, the irradiance received at the sensor can be derived from Bouguer's law:

$$E_{\text{sensor}} = \frac{I_{\text{object}} \cos(\theta_{\text{sensor}})}{r^2} \qquad (2.53)$$

Where E_{sensor} is the irradiance at the sensor, I_{object}, the radiant intensity leaving the object, r and θ_{sensor} are defined in Fig. 2.19. With the infinitesimal surfaces hypothesis, one can get the luminance from the intensity:

$$dI_{\text{object}} = L_{\text{object}} \cos\left(\theta_{\text{object}}\right) dS_{\text{object}} \tag{2.54}$$

By using the fact that the surfaces are punctuals, the irradiance received at the sensor is finally:

$$E_{\text{sensor}} = \frac{\cos\left(\theta_{\text{object}}\right) \cos(\theta_{\text{sensor}}) S_{\text{object}}}{r^2} L_{\text{object}} \tag{2.55}$$

By using Planck's law and defining $g : (\theta_{sensor}, \theta_{object}, r) \rightarrow \frac{\cos\left(\theta_{object}\right) \cos(\theta_{sensor}) S_{object}}{r^2}$, assumed to be known in this case:

$$E_{sensor} = g(\theta_{sensor}, \theta_{object}, r) \epsilon_{object} \frac{C_1}{\lambda^5 \left(e^{\frac{C_2}{\lambda T_{object}}} - 1\right)} \tag{2.56}$$

with $C_1 = \frac{2hc^2}{\lambda^5}$ and $C_2 = \frac{hc}{k}$.

By using Wien's approximation and taking the logarithm of the previous expresion, the emissivity and temperature can be separated to derive the following expression:

$$\ln(E_{\text{sensor}}) + \ln\left(\frac{\lambda^5}{C_1 g(\theta_{sensor}, \theta_{object}, r)}\right) = \ln(\epsilon_{object}) - \frac{C_2}{\lambda T_{object}} \tag{2.57}$$

The first term in Eq. (2.56) is the physical quantity that will be measured at the sensor. The second term depends on the geometry of the scene and is assumed to be known here. Finally, the last two terms represent the unknowns of the system: the temperature and the emissivity of the object. Therefore, one can see that a passive infrared thermography measurement system is facing an undetermined problem with one equation and two unknowns.

Multi-band thermography

Multiband sensors provide a measurement set of spectrally integrated radiance in various finite-bandwidth spectral channels. An idea that could come into mind is to perform a measurement at another wavelength in order to bring new information to the system. However, new unknowns are also added when working in multi-band thermography, the spectral emissivity at those supplementary wavelengths [15]. Let N be the number of spectral bands, $i \in \mathbb{N}, i \in [1, N]$ and $\bar{\Delta}\lambda_i$ be the median

value of the i$^{\text{th}}$ spectral band:

$$
\begin{cases}
\ln\left(E_{\bar{\Delta\lambda}_1,sensor}\right) + \ln\left(\frac{\bar{\Delta\lambda}_1^5}{C_1 g(\theta_{sensor},\theta_{object},r)}\right) = \ln\left(\epsilon_{\bar{\Delta\lambda}_1,object}\right) - \frac{C_2}{\Delta\lambda_1 T_{object}} \\
\ln\left(E_{\bar{\Delta\lambda}_2,sensor}\right) + \ln\left(\frac{\bar{\Delta\lambda}_2^5}{C_1 g(\theta_{sensor},\theta_{object},r)}\right) = \ln\left(\epsilon_{\bar{\Delta\lambda}_2,object}\right) - \frac{C_2}{\Delta\lambda_2 T_{object}} \\
... \\
\ln\left(E_{\bar{\Delta\lambda}_N,sensor}\right) + \ln\left(\frac{\bar{\Delta\lambda}_N^5}{C_1 g(\theta_{sensor},\theta_{object},r)}\right) = \ln\left(\epsilon_{\bar{\Delta\lambda}_N,object}\right) - \frac{C_2}{\Delta\lambda_N T_{object}}
\end{cases}
\tag{2.58}
$$

As a consequence the system is still undetermined and is composed of N equations for $N + 1$ unknowns. Even if the equation system is unknown, multispectral measurements did brought information. Indeed, the infrared energy radiated by an object is proporional to its absolute temperature T_{object} and emissivity ϵ_{object}. However, sources of stray radiation (background, sky, sun and so on) having a temperature T_{env} and emissivity ϵ_{env} can add, after reflection a disturbing radiative contribution to the previous intrinsic emission and reflectance of the object. Of course, an infrared imager will be sensitive to the sum of all those energies and will become a function of all the parameters that will interfere. In the case of the grey-body assumption, a measurement on a single spectral band will give one relation between T_{obj} and ϵ_{obj}. Multispectral measurements can therefore provide more information and particularly when ϵ_{obj} is a function of wavelength. For example, if the same radiance is produced in one band ($\Delta\lambda_2$ in Fig. 2.20) because of a lower emissivity compensated by a higher temperature; it will probably not be the case for another band ($\Delta\lambda_1$):

2.5   Synthesis

Infrared thermography for structural health monitoring has been introduced in this chapter. Equations that govern the physical phenomena of radiative transfers have been derived and studied for the particular case of IR thermography measurements. It has been shown that the IR thermography measurement process involves many variables that need to be taken into account in the hope of getting accurate temperature measurements. In particular, the emissivity plays an important role in the equations. Without emissivity, the temperature estimation uncertainty may be high and therefore, emissivity-temperature simultaneous estimation methods are needed. The objective of this thesis is to develop such method, applied to *in-situ* long-term thermal monitoring of infrastructures. To achieve this, the understanding of the full measurement process is needed. In particular and as shown in the former sections, the knowledge of the environmental parameters helps at improving the accuracy of the temperature estimation. Moreover, the infrared camera needs to be calibrated both spatially and thermally. These topics will be discussed in Chapter 4. Furthermore, it has been shown that the set-up of the instrumentation will have an impact on the obtained measurement and therefore the potential temperature estimate. In order to optimize such instrumentation but also to test the emissivity and
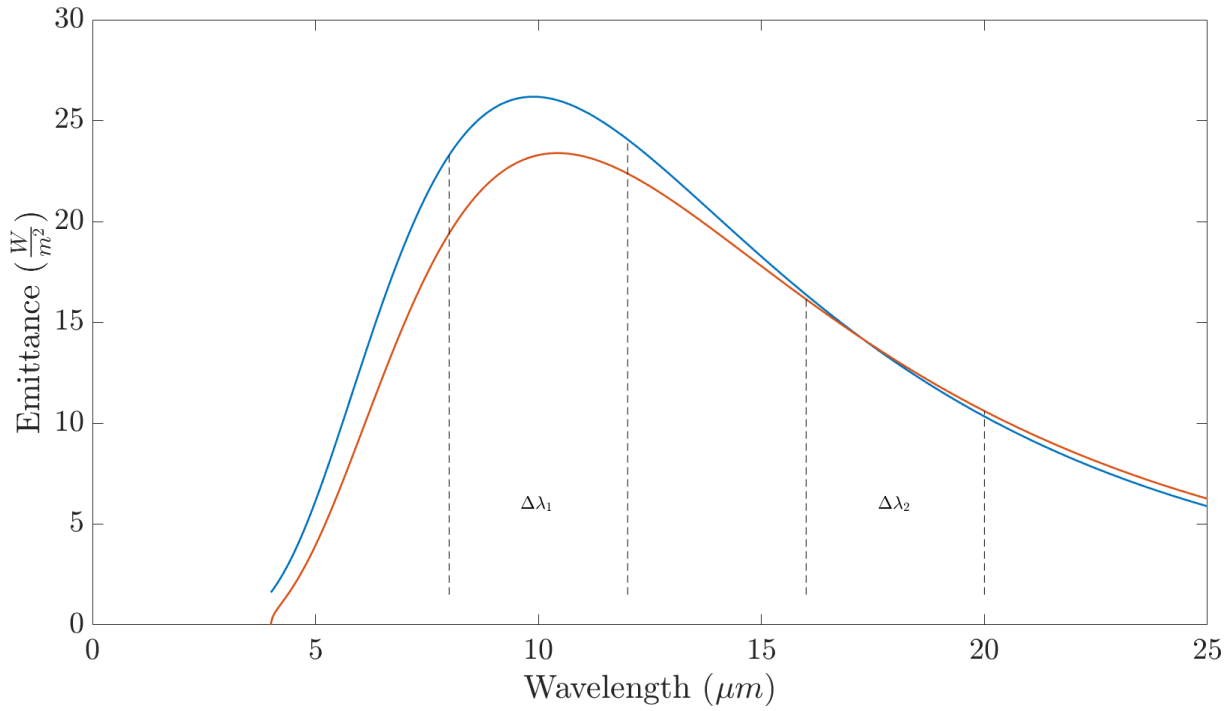
Figure 2.20: Multi-band measurements bring more information

temperature estimation methods, a 3D simulation software tool has been developed and studied.

In the next chapter, the methods developed in the literature for separating the emissivity and the temperature will be presented. Then, a bibliographical study about solving the radiosity equation will also be done in view of implementing a custom simulation software.

Bibliographical study

Ɪɴ this chapter we will review the problem of extracting the temperature and the emissivity information from thermal infrared multispectral radiance data. We will present the different methods that exist in the litterature and discuss them. Being able to identify the emissivity and the temperature at the same time while knowing the approximation made on the measurement is a challenge that could enhance infrared data exploitation and infrared cameras applications possibilities. We will focus here on the simultaneous evaluation of emissivity and surface temperature of objects observed by infrared thermography, for *in-situ* applications.

Moreover, in order to test the efficiency, and robustness of those methods but also to optimize the *in-situ* instrumentation, a simulation software tool using the forward radiosity equation solution has been developed. Thanks to literature's models, infrared images can be simulated from 3D models. A brief state of the art of illumination algorithms to render such images is therefore presented in this chapter. As explained previously, the temperature cannot be deduced from IR measurements without a good emissivity estimate.

3.1   Emissivity and temperature separation methods

As seen before, the lack of knowledge on radiation properties of the real scene makes the thermal measurement through infrared thermography complicated. In first approximation, the irradiance received by a sensor for a given wavelength depends on the objects' temperature and its emissivity at this wavelength, as explained in the previous chapter. However, the radiative system of such configuration leads to an undetermined system. Therefore, retrieving the temperature from such a measurement needs the knowledge of the object, or by default, the use of inferences and approximations on the observed system. The methods are classified into two categories. A first category regroup the optimization based methods that are derived from a mathematical approximation at some point to reduce the number of variables or to increase the redundancy of data to get more equations. The second one gathers the Bayesian methods for which a prior statistical information is added to the system in order to be able to solve it. The literature has different models for estimating the emissivity from remotely sensed images and most of those models are used for computing satellite data. The main difficulty here is to separate the two unknowns of the equation: the emissivity and the temperature. Even if the equations behind airborne applications and *in-situ* applications are close, the conditions and assump-

tions to be applied may differ, in particular when considering the atmospheric contribution. However, the methods used in the remote field sensing can be inspiring for *in-situ* long-term thermal monitoring. [102] is a good introduction for having a global point of view on general noncontact temperature measurement by radiative methods.

3.1.1 In the field of Remote Sensing

Land surface temperature (LST) is an important parameter for various applications. Such indicator provides information on the temporal and spatial variations of the surface equilibrium state and is used for vegetation monitoring, hydrological cycle, urban and environmental studies or climate change for example. Many methods have been developed to retrieve the LST from satellite infrared data. The radiative transfer equation for measurements performed at the top of the atmosphere is different than the one used for *in-situ* measurements and presented in the previous chapter. Those measurements have to take into account the atmospheric path that the beam has taken until it reaches the satellite sensor. Even though the model is different, it is still interesting to take a look at the developed emissivity / temperature separations methods and see if it can be applied to *in-situ* long-term IRT on ground. The measured radiance at sensor level can be written [117]:

$$L_\lambda(\theta, \phi, T) = \tau_\lambda(\theta, \phi) \left(\epsilon_\lambda(\theta, \phi) L^\circ_{\lambda, land}(T) + (1 - \epsilon_\lambda(\theta, \phi))(L^\downarrow_{\lambda, a} + L^\downarrow_{\lambda, sun}) + \tilde{L}_{\lambda, sun} \right) + L^\uparrow_{\lambda, atm} + \tilde{L}^\uparrow_{\lambda, atm}$$

$$(3.1)$$

where (see Fig. 3.1):

- $\tau_\lambda(\theta, \phi)$ is the atmospheric transmission,
- $\epsilon_\lambda(\theta, \phi)$ is the land surface emissivity,
- T is the land surface temperature,
- $L^\circ_{\lambda, land}(T)$ is the land surface emission,
- $L^\downarrow_{\lambda, atm}$ is the down-welling atmospheric radiance reflected by the surface,
- $L^\downarrow_{\lambda, sun}$ is the down-welling diffuse solar radiance reflected by the surface,

– $L_{\lambda, sun}$ is the direct solar radiance,

- $\tilde{L}_{\lambda, sun}$ is the direct solar radiance reflected by the surface,
- $L^\uparrow_{\lambda, atm}$ is the atmospheric emission,
- $\tilde{L}^\uparrow_{\lambda, atm}$ is the atmospheric scattering.

and $\tilde{L}_{\lambda, sun}$ is defined by:

$$\tilde{L}_{\lambda, sun} = \rho_\lambda(\theta, \phi, \theta_s, \phi_s) L_{\lambda, sun} \cos(\theta_s) \tau_\lambda(\theta_s, \phi_s) \tag{3.2}$$

> **Note**
>
> For simplification in the writting, the formalism of the Remote Sensing field will be used in this section. In fact, due to the narrow bands involved, λ notation is used in place of $\Delta\lambda$.

$$\tilde{L}^{\uparrow}_{\lambda,atm} \qquad L^{\uparrow}_{\lambda,atm} \qquad \tau_\lambda(\theta,\phi)$$

$$\epsilon_\lambda(\theta,\phi)L^{\circ}_\lambda(T) \;+(1-\epsilon(\theta,\phi))(L^{\downarrow}_{\lambda,atm}+L^{\downarrow}_{\lambda,sun})+\;\tilde{L}_{\lambda,sun}$$
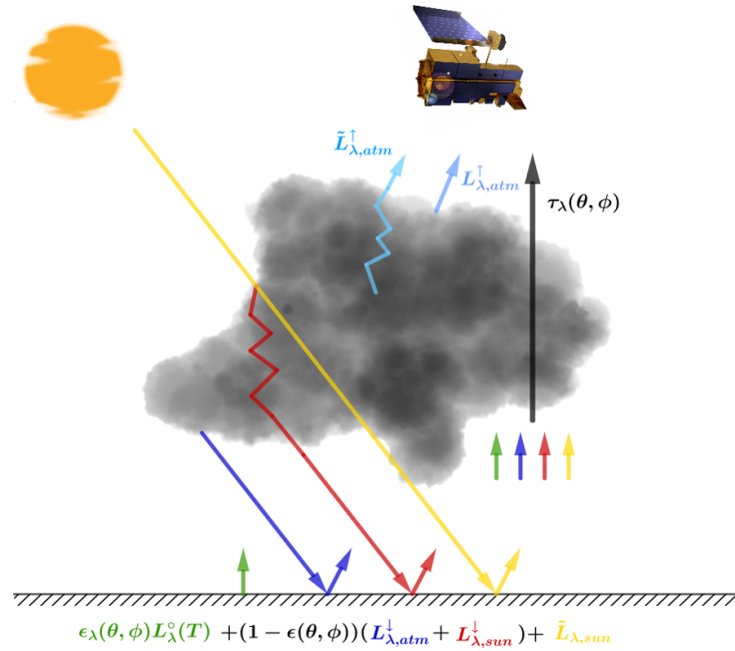
Figure 3.1: Radiative transfer equation for satellite measurements, done at the top of the atmosphere.

Moreover, when dealing with satellite data, the atmospheric effects along the path from the ground to the sensor are assumed to be properly estimated. This case makes the emissivity retrieval close to the problem that occurs on land measurements (N measurements for $N + 1$ unknowns). In order to get rid of those parameters, atmospheric radiative transfer models such as SMARTS2 ([82]), MODTRAN ([19]) or MATISSE ([67]) can be used. In this field, the unknowns are referred to LSE and LST for land surface emissivity and land surface temperature reciprocally. Many of the temperature and emissivity separation methods presented here are introduced and discussed in [117] state of the art.

Known emissivity

The following methods are temperature - emissivity separation methods but those methods do not actually estimate conjointly the temperature and emissivity. They need prior knowledge on LSE [117]. Those methods are still presented here for their data information extraction from IR measurements.

Thermal log residuals By using Wien's approximation of Planck's law Eq. (2.8) the log can be expressed as:

$$\ln\left(\epsilon \times \frac{C1}{\pi\lambda^5 \exp\left(\frac{C_2}{\lambda T}\right)}\right) = \ln(\epsilon) + \ln(C_1) - 5\ln(\lambda) - \ln(\pi) - \frac{C_2}{\lambda T} \tag{3.3}$$

By multiplying this equation by λ, we can dissociate T from λ. Let X_{pc} be the resulting quantity for the pixel p and the channel c of a multispectral infrared sensor.

$$X_{pc} = \lambda_c \ln(\epsilon_{pc}) + \lambda_c \ln(C_1) - 5\lambda_c \ln(\lambda) - \lambda_c \ln(\pi) - \frac{C_2}{T_p} \tag{3.4}$$

Then, the "residuals" (Y_{pc}) are defined as the remaining of this quantity after having substracted the average value over the image ($\tilde{X}_c$), the average value over the channels ($\tilde{X}_p$) and the average value over both images and channels ($\tilde{X}$):

$$Y_{pc} = X_{pc} - \tilde{X}_p - \tilde{X}_c + \tilde{X} \tag{3.5}$$

With N the number of spectral channels and M the number of pixels in one image:

$$\begin{aligned}
\tilde{X}_p &= \frac{1}{N}\sum_{j=1}^{N} X_{pj} \\
\tilde{X}_c &= \frac{1}{M}\sum_{i=1}^{M} X_{ic} \\
\tilde{X} &= \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{N} X_{ij}
\end{aligned} \tag{3.6}$$

From equations Eq. (3.4) and Eq. (3.6):

$$Y_{pc} = \lambda_c \ln(\epsilon_{pc}) - \frac{1}{N}\sum_{j=1}^{N}\lambda_j \ln(\epsilon_{pj}) - \frac{\lambda_c}{M}\sum_{i=1}^{M}\ln(\epsilon_{ic}) + \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M}\lambda_j \ln(\epsilon_{ij}) \tag{3.7}$$

Finally, let $l_c = \frac{\lambda_c}{\tilde{\lambda}}$ and $\epsilon_{pc} = e_{pc}^{-l_c}$ where $\tilde{\lambda}$ is the mean of the central wavelengths of all channels, then:

$$\begin{aligned}
\frac{Y_{pc}}{\tilde{\lambda}} &= \ln(e_{pc}) - \frac{1}{N}\sum_{j=1}^{N}\ln(e_{pj}) - \frac{1}{M}\sum_{i=1}^{M}\ln(e_{ic}) + \frac{1}{NM}\sum_{j=1}^{N}\sum_{i=1}^{M}\ln(e_{ij}) \\
&= \ln(e_{pc}) - \frac{1}{N}\ln\left(\prod_{j=1}^{N} e_{pj}\right) - \frac{1}{M}\ln\left(\prod_{i=1}^{M} e_{ic}\right) + \frac{1}{NM}\ln\left(\prod_{j=1}^{N}\prod_{i=1}^{M} e_{ij}\right)
\end{aligned} \tag{3.8}$$

Therefore, the thermal log residual enables to separate the variation of temperature and emissivity to a variation of combined emissivity.

Alpha residuals   Similarly to thermal log residuals, the alpha log residuals performs a mean substraction but only with the wavelength-weighted log pixel this time [90].

$$\alpha_{pc} = X_{pc} - \tilde{X}_p \tag{3.9}$$

Therefore, let $d : x_c \mapsto x_c - \frac{1}{N} \sum_{c=1}^{N} x_c$, the alpha log residuals equals to:

$$\alpha_{pc} = d(\lambda_c \ln(\epsilon_{pc})) + \underbrace{(\ln(C_1) - \ln(\pi))d(\lambda_c) - 5d(\lambda_c \ln(\lambda_c))}_{K_c} \tag{3.10}$$

Where K_c is a constant known for a given channel and depends only on the wavelength.
Let r be one reference channel. Therefore, it is possible to write:

$$\frac{\epsilon_{pc}}{\epsilon_{pr}^{n_c}} = e^{\frac{1}{\lambda_c}[(\alpha_{pc}+K_c)-(\alpha_{pr}+K_r)]}, \quad n_c = \frac{\lambda_c}{\lambda_r} \tag{3.11}$$

The same authors showed that

$$\alpha_{pc} = d(\lambda_c \ln(B_{pc})) + K_c \tag{3.12}$$

Meaning that the alpha residuals can be directly computed from the measured radiance.

Those two methods (alpha log and thermal log residuals) have the advantages of being simple and applicable to one channel sensors only. However, some priors on the emissivity are required and the important source of errors in those methods will be the accuracy of the emissivity pixel's estimate. Moreover, the uncertainty on atmospheric estimated parameters will have a strong effect on LST retrieval [117].

Other methods with known emissivity   Other methods with known emissivity rather than the log residuals have been developed. Those methods use different assumptions to improve the accuracy of the estimated LST or, more generally to reduce the strong knowledge requirement of known emissivity at each pixel. Multi-channel algorithms for example use different atmospheric absorption in adjacent channels and reduce the need for accurate atmospheric profiles [18, 2, 167, 123].

Similarly, another way is to use the angle dependence of the atmospheric parameters. A change of the angle will change the atmospheric absorption due to the different path-lengths from different views. Again, such method reduces the need for accurate atmospheric profiles. However, the angular emissivity variation with angle needs to be known and also geometric registration becomes fundamental [169, 168, 137].

Unknown emissivity - two steps-methods

The previous presented methods were assuming that the LSE was known at some point. However, in practice, knowing LSE is challenging due to the heterogeneity of the surfaces and its angular and spectral variations. Moreover, an uncertainty on the emissivity leads to an important uncertainty on the temperature estimation (see next chapter for a sensitivity study regarding *in-situ* long-term monitoring). Therefore, methods for estimating both LSE and LST have been developed in the literature. In [117] those methods are classified into two groups: the two-steps methods and the simultaneous estimation of LSE and LST. The first group methods estimate first the LSE and then use one of the former method to estimate the LST. At the opposite, the simultaneous estimation methods aims at providing *a-priori* on the system to estimate at the same time LSE and LST.

Temperature Independent Spectral Indices (TISI) $TISI$ coefficients have been proposed for spectral analysis in the TIR (thermal infrared, centered at $8.9\mu m$) band. Those coefficients aim at getting rid of surface temperature in their expression to only express a variation of emissivity through a reference channel. Those coefficients are called Temperature-Independent Spectral Indices ($TISI$). Please note that $TISI$ theory and algebra have been developed in the original article and different application examples are provided in [17]. Based on those coefficients, a new two-steps method has been proposed by assuming that $TISI$ coefficients on particular channels will not vary substantially between day and night [114, 115]. The $TISI_{ij}$ are referred by two channels i and j, usually i is the MIR (mid infrared, centered at $3.8\mu m$) channel and j the TIR (thermal infrared, centered at $8.9\mu m$) reference channel. $TISI_{ij}$ are defined by using the power law approximation of Planck's function:

$$L_i^\circ(T) = a_i T^{n_i} \tag{3.13}$$

where a_i and n_i are channel-specific constants. $TISI_{ij}$ indices are therefore defined by:

$$TISI_{ij} = \frac{L_i^\circ(T_{gi}) - L_{i,a}^\downarrow}{L_i^\circ(T_{gj}) - L_{i,a}^\downarrow} \tag{3.14}$$

which they shown mathematically that [17]:

$$TISI_{ij} \approx \frac{\epsilon_i}{\epsilon_j^{n_i/n_j}} \tag{3.15}$$

Those spectral indices are therefore independent of ground surface temperature and are an indicator of the LSE evolution in the two channels. By assuming that the $TISI$ variation can be neglected between daytime without the contribution of solar illumination and nighttime then the bidirectional reflectivity in MIR channel is found back by removing the emitted radiance during the day and comparing it with night $TISI_{ij}$ coefficients. LSEs in the TIR channels are then deducted based on the $TISI$ and

emissivity in the MIR channel. Former methods are then used to estimate the LST.

$TISI$ indices have been used in different methods and have shown satisfying overall results [116]. However, due to the constant emissivity ratio during day and night assumption, a strong requirement is to have the same observation view during night and day with accurate geometry registration. Furthermore, this method works only for one MIR channel and one TIR channel at least.

Unknown emissivity - simultaneous estimation of LSE and LST

The following methods aim at estimating conjointly LSE and LST. This time, the methods provide directly both estimates, without estimating first the LSE and then the LST as in two-steps methods.

Normalized-Emissivity separation method (NES)   Let assume that the maximum emissivity ϵ_{max} is known, without knowing especially its associated wavelength or band. Firstly, a relatively high temperature $\tilde{T}$ is chosen. The following radiometric model is then solved for each spectral band at each pixel to get a first estimate of the emissivity :

$$\tilde{\epsilon}_\lambda = \frac{L_{\lambda,land}(T) - L_\lambda^\downarrow}{L_{\lambda,land}^\circ(\tilde{T}) - L_\lambda^\downarrow} \tag{3.16}$$

Where $L_\lambda^\downarrow = L_{\lambda,a}^\downarrow + L_{\lambda,s}^\downarrow$ and $L_{\lambda,land}$ is the ground-leaving radiance. The temperature $\tilde{T}$ is then decreased respectively increased until $\tilde{\epsilon}_\lambda = \epsilon_{max}$ if $L_{\lambda,land}(T) > L_\lambda^\downarrow$, respectively $L_{\lambda,land}(T) < L_\lambda^\downarrow$. Another way to achieve such separation is to make the gray body assumption where $\epsilon_\lambda = \epsilon_{\max}$. In this case, the temperature can be found when $L_{\lambda,land}(T) > L_\lambda^\downarrow$:

$$\tilde{T} = \max_\lambda \left[L^{\circ-1} \left(\frac{L_{\lambda,land}(T) - L_\lambda^\downarrow}{\epsilon_{\max}} + L_\lambda^\downarrow \right) \right] \tag{3.17}$$

When $L_{\lambda,land}(T) < L_\lambda^\downarrow$, a minimum must be found instead. Once $\tilde{T}$ has been estimated, the emissivity spectrum is deduced from Eq. (3.16).

According to [102], in the case of bare soils with rocks, this method is quite precise (0.02 emissivity uncertainty, $\epsilon_{\max} = 0.96$). However, when the surface is unknown, the underlying hypothesis is not satisfied and the inacurracy increase and becomes finally high (± 3K).

Temperature-Emissivity separation method (TES)   In order to overcome the high inaccuracies induced by the NES method, the TES method has been developed. Proposed by [3], the TES method is based on the fact that authors have observed that the relative emissivity of $\beta_\lambda = \tilde{\epsilon}_\lambda / \bar{\tilde{\epsilon}}$ is rather insensitive to the error made on the temperature estimation [102]. $\tilde{\epsilon}_\lambda$ is derived from Eq. (3.16) and $\bar{\tilde{\epsilon}}$ its average on wavelength. In order to obtain the absolute spectrum from this relative one, [3] proposed a correlation:

$$\epsilon_\lambda = \beta_\lambda \frac{\epsilon_{\min}}{\beta_{\min}}, \quad \epsilon_{\min} \approx A - B(\beta_{\max} - \beta_{\min})^C \tag{3.18}$$

The coefficients A, B and C are obtained through a regression depending on a material database. In their article, $A = 0.994$, $B = 0.687$ and $C = 0.737$ but other values can be found in the literature. Finally, the temperature is derived from:

$$T = L^{\circ-1}\left[\frac{L_{\lambda_M}(T) - L_{\lambda_M}^\downarrow}{\epsilon_{max}} + L_{\lambda_M}^\downarrow\right], \quad \lambda_M = \arg\max_\lambda(\epsilon_\lambda) \tag{3.19}$$

TES method is used to calclulate surface temperature for ASTER satelite. The TES has been validated on different sites, showing that TES performs well for ASTER data ($+1.5$K and $+0.015$ of correct values) [150]. However, the correlation is highly dependent on the chosen database materials and thus the generalization to any site and any scale of this method seems tricky [74].

Classification based methods   Classification methods are widespread for identifying parameters in images. In order to derive LSE from the measurements, a classification algorithm is performed to classify a given pixel to a given class. By using the right database these methods offer good accuracy results (about 0.01) [166]. Once the LSE has been estimated, LST is derived from previous methods. However, the classification algorithm needs a strong knowledge of the emissivity of each class as well as robust and reliable classification map which becomes complicated when considering soil moisture, seasoning, snow *etc.* Moreover, classification is also more difficult to perform for coarse pixels, due to the mixed materials observation and then strong correlation within classes [73].

Other methods   Please note that this method list is not comprehensive, and other methods could be quoted. For example, the spectra smoothness method which states that if LST is not well estimated, then saw-teeth will appear on the estimated LSE signal due to atmospheric absorption lines [117]. When the estimated LST comes closer to the real temperature, the estimated LSE becomes progressively smoother. The estimate is then found when LSE is the smoothest. This method offers interesting results for satelite and airborne applications. However, the underlying hypothesis is to use the atmospherical spectrum properties to have a better estimate of the temperature and emissivity. Furthermore, this method requires hyperspectral instruments and a good high spectral resolution in order to be able to distinguish the atmospheric spectral features from the emissivity features. Such method is therefore out of scope for *in-situ* multispectral measurements.

3.1.2   General separation methods

Whereas the previous section introduced Remote Sensing field specific methods, this section presents a more general overview of the temperature and emissivity separation problem, from a mathe-

matical basis. Whatever the field of application, once the environmental or atmospheric parameters have been identified, it is possible to bring the model back to the case: $\gamma_{\Delta\lambda_i} = f(\Delta\lambda_i, \epsilon_{\Delta\lambda_i}, T)$ where f is known without accounting for specific channels or atmospheric properties to estimate the temperature and emissivity. Therefore, some presented methods such as the grey body and multi-temperature methods have been actually developed in the Remote Sensing field. In the following, the conjoint estimation of emissivity and temperature will be studied, based on the sensor - object Eq. (2.56).

Linearization of the problem

Let $\gamma_{\Delta\lambda_i}$ be $g^{-1}(\theta_{sensor}, \theta_{object}, r)C_1^{-1}\Delta\lambda_i^{5}E_{\Delta\lambda_i, sensor}$. This quantity is assumed to be known in the following, representing the measurements for which we have prior knowledge about the geometry. As a reminder, we want to solve the system:

$$\ln(\gamma_{\Delta\lambda_i}) = ln(\epsilon_{\Delta\lambda_i}) - \frac{C_2}{\Delta\lambda_i T}, \quad \forall i \in [1; N] \tag{3.20}$$

Monomial model

In order to solve this underdetermined problem, a first approach is to estimate $\ln(\epsilon_{\Delta\lambda_i})$ by a polynom. Let $(\alpha_0, \alpha_1, ..., \alpha_m) \in \mathbb{R}^{m+1}$ where $m \leq N - 2$ such as:

$$\ln(\epsilon_{\Delta\lambda_i}) = \sum_{k=0}^{m} \alpha_k \Delta\lambda_i^{k}, \forall i \in [|1; N|] \tag{3.21}$$

Such approximation leads to the following equation system:

$$\Gamma = MX = \begin{pmatrix} 1 & \Delta\lambda_1 & \Delta\lambda_1^2 & \dots & \Delta\lambda_1^m & -\frac{C_2}{\Delta\lambda_1} \\ 1 & \Delta\lambda_2 & \Delta\lambda_2^2 & \dots & \Delta\lambda_2^m & -\frac{C_2}{\Delta\lambda_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta\lambda_N & \Delta\lambda_N^2 & \dots & \Delta\lambda_N^m & -\frac{C_2}{\Delta\lambda_N} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \\ \frac{1}{T'} \end{pmatrix} \tag{3.22}$$

Where $\Gamma = (ln(\gamma_{\Delta\lambda_i}))_{1 \leq i \leq N}$ and M is a $[N \times (m + 2)]$ matrix and T' is the estimated temperature obtained from the polynomial approximation and measurement. In the following, we will remove the Δ notation inferring that the sensor integrates on a small band. Then, the λ_i notation will refer to a band and not a particular wavelength.

Monomial model (*$m = N - 2$*) Let study the case where we have exactly N equations for N unknowns. This is possible by using a $N - 2$ order polynomial. By rewriting Eq. (3.22), we get a square matrix on

which we recognize the shape of a Vandermonde matrix and we can compute the determinant:

$$\det(M) = C_2(-1)^{N+1}\frac{\prod_{1\leq i<j\leq N}(\lambda_i - \lambda_j)}{\prod_{i=1}^{N}\lambda_i} \leq C_2(-1)^{N+1}\frac{\Delta\lambda_{max}^N}{\prod_{i=1}^{N}\lambda_i} \tag{3.23}$$

where $\Delta\lambda_{max}$ is the maximum difference of two bands. We can see that the absolute value of this determinant decreases exponentially as N increases, meaning that M is approaching singularity when the number of bands increases. The literature shows that the polynomial approximation leads to un-realistics temperatures for $N > 3$ [37]. Such behavior can be explained by the fact that even if the Taylors serie development of the ln function enables us to estimate the logarithm in Eq. (3.21) because of the boundaries of ϵ, we need to get a N large enough to get a good approximation as shown on Fig. 3.2. Therefore, for a N too small, the variables will be considered sufficiently independents but the approximation is rough. At the opposite, when N increases, the approximation of the logarithm becomes satisfying but we induce correlation between the different variables which makes the system difficult to inverse.
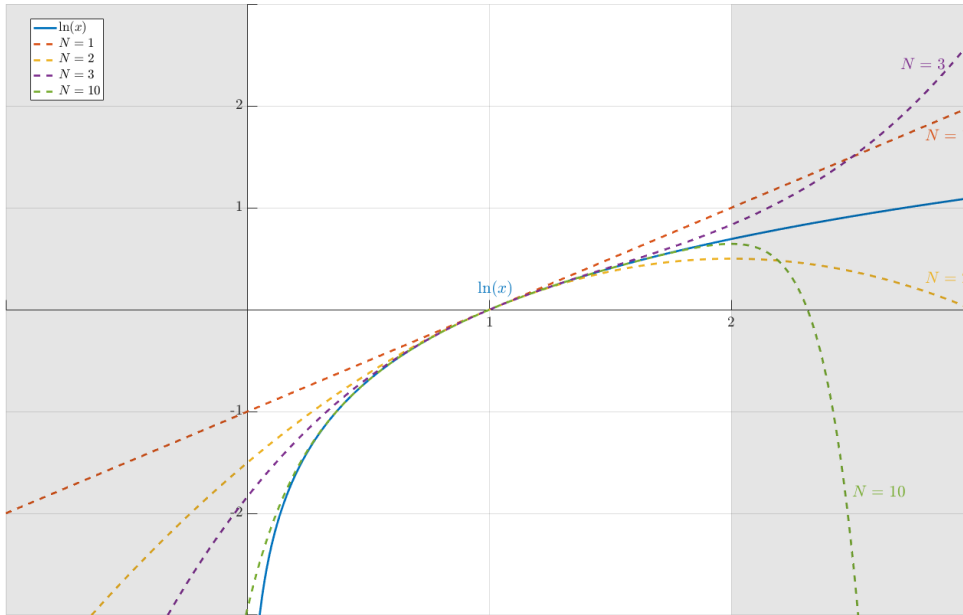
Figure 3.2: ln approximation with its Taylor serie

Since the polynom has exactly N equations for N unknowns, it means that the fitting will pass exactly by the N values of γ_{λ_i}. As shown in [37], by multiplying Eq. (3.21) by λ_i and rearranging Eq. (3.20), one can find the set of N equations

$$\lambda_i ln(\epsilon_i) = \sum_{j=0}^{N-2} a_{j-1}\lambda_i^j \tag{3.24}$$

where a_{-1} is the constant term evaluated on $\lambda_i = 0$, giving an idea of the error made on the temperature, from Eq. (3.20):

$$a_{-1} = C_2 \left(\frac{1}{T} - \frac{1}{T'} \right) \tag{3.25}$$

Since the polynom will pass exactly through the N values of $\lambda_i ln(\epsilon_i)$, we can use Lagrange polynomials to find the values

$$p(\lambda) = \sum_{j=0}^{N-2} a_{j-1} \lambda^j = \sum_{i=1}^{N} \left[\lambda_i ln(\epsilon_i) \prod_{j=1, j \neq i}^{N} \frac{\lambda_j - \lambda}{\lambda_j - \lambda_i} \right] \tag{3.26}$$

With Eq. (3.18), we get the value of a_{-1} [37]:

$$a_{-1} = C_2 \left(\frac{1}{T} - \frac{1}{T'} \right) = \prod_{j=1}^{N} \lambda_j \sum_{i=1}^{N} \left[\frac{ln(\epsilon_i)}{\prod_{j=1, j \neq i}^{N} (\lambda_j - \lambda_i)} \right] \tag{3.27}$$

Therefore, only a polynom that will exactly match the N values of $ln(\epsilon_i)$ will have a null extrapolation error and be the real emissivity; which is unlikely to happen.

Based on those equations, a particular study of the different pyrometry techniques (two, three and four colors) can be done ([100, 37, 135]). We can observe the conditions that are necessaries to get a null error (*id est* solving $\alpha_{-1} = 0$). Let $\lambda_i \neq \lambda_j, \forall i, j \in [|1, N|], i \neq j$

$$
\begin{aligned}
N = 2 \quad & C_2 \left(\frac{1}{T} - \frac{1}{T'} \right) = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2} ln(\frac{\epsilon_2}{\epsilon_1}) = 0 \iff \epsilon_1 = \epsilon_2 \\
N = 3 \quad & C_2 \left(\frac{1}{T} - \frac{1}{T'} \right) = 0 \iff \epsilon_1 \epsilon_3 = \epsilon_2^2 \quad \text{(equidistant bands)}
\end{aligned} \tag{3.28}
$$

Those results show that strong hypothesis are needed on the system we use that may lead to uncertain measurements without prior knowledge on emissivities. Finally, it is well known that overfitting of the orignal data on interpolation based methods leads to errors (Runge's phenomenon). Such methods can be retained only for simple pyrometers (two or three wavelengths at most) [37].

To overcome the previous overfitting issue, one may try to reduce the order of the polynom used to fit the data. In that case, the estimated temperature value is obtained by a least-square method, to solve the overdetermined problem.

Monomial model ($m < N - 2$)   To overcome some numerical issues, it is necessary to normalize the system matrix in order to reduce its condition number. This can be achieved by replacing the wavelength band in the previous polynomial expression by its reduced centered value such as: $\lambda_i^* \in [-1; 1]$

[135]:

$$\lambda_i^* = 2 \frac{\lambda_i - \lambda_{min}}{\lambda_{max} - \lambda_{min}} \tag{3.29}$$

For the same reasons, we normalize the temperature with T_{ref} such as $\frac{C_2}{\lambda_i T_{ref}}$ is close to one. Therefore, we can write the new normalized system:

$$\Gamma = MX = \begin{pmatrix} 1 & \lambda_1^* & \lambda_1^{*2} & \dots & \lambda_1^{*m} & -\frac{C_2}{\lambda_1^* T_{ref}} \\ 1 & \lambda_2^* & \lambda_2^{*2} & \dots & \lambda_2^{*m} & -\frac{C_2}{\lambda_2^* T_{ref}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_N^* & \lambda_N^{*\,2} & \dots & \lambda_N^{*\,m} & -\frac{C_2}{\lambda_N^* T_{ref}} \end{pmatrix} \begin{pmatrix} \alpha_0^* \\ \alpha_1^* \\ \alpha_2^* \\ \vdots \\ \alpha_m^* \\ \frac{T_{ref}}{T'} \end{pmatrix} \tag{3.30}$$

In that case, the least-square method resolution leads to solve the normal equations, to minimize the sum of the square differences .

$$(M^T M)X = M^T \Gamma \tag{3.31}$$

Many methods exist in the literature to solve such equation system such as the Singular Value Decomposition (SVD), the QR decomposition, *etc.* [52]

Orthogonal basis ($m < N - 2$)   The approximation of taking the logarithm as a polynomial could be improved by choosing the right polynomial basis instead of the previous presented basis function $\{P_0, P_1, ..., P_N\}$. In fact, it is clear that the approximation of a function to a polynomial is like a projection on the space spanned by the basis function. Using orthogonal polynomials helps to have uncorrelated explanatory variables. Let $\{\Phi_0, \Phi_1, ..., \Phi_m\}$ an orthogonal set of functions of $\mathbb{R}^{+*}$ and scalars $(\alpha_0, \alpha_1..., \alpha_m)$ such as:

$$\Gamma = MX = \begin{pmatrix} \Phi_0(\Delta\lambda_1^*) & \Phi_1(\Delta\lambda_1^*) & \dots & \Phi_m(\Delta\lambda_1^*) & -\frac{C_2}{\lambda_1 T_{ref}} \\ \Phi_0(\Delta\lambda_2^*) & \Phi_1(\Delta\lambda_2^*) & \dots & \Phi_m(\Delta\lambda_2^*) & -\frac{C_2}{\lambda_2 T_{ref}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi_0(\Delta\lambda_N^*) & \Phi_1(\Delta\lambda_N^*) & \dots & \Phi_m(\Delta\lambda_N^*) & -\frac{C_2}{\lambda_N T_{ref}} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_m \\ \frac{T_{ref}}{T'} \end{pmatrix} \tag{3.32}$$

By combining equations Eq. (3.31) and Eq. (3.32), one can see that the orthogonal basis will increase

diagonal coefficients and reduce other coefficients :

$$
M^T M = \begin{pmatrix}
\sum_{i=1}^{N} \Phi_0^2(\Delta\lambda_i^*) & \sum_{i=1}^{N} \Phi_0(\Delta\lambda_i^*)\Phi_1(\Delta\lambda_i^*) & \dots & \sum_{i=1}^{N} \Phi_0(\Delta\lambda_i^*)\Phi_m(\Delta\lambda_i^*) & -\sum_{i=1}^{N} \Phi_0(\Delta\lambda_i^*)\frac{C_2}{\lambda_i T_{ref}} \\
\sum_{i=1}^{N} \Phi_0(\Delta\lambda_i^*)\Phi_1(\Delta\lambda_i^*) & \sum_{i=1}^{N} \Phi_1^2(\Delta\lambda_i^*) & \dots & \sum_{i=1}^{N} \Phi_1(\Delta\lambda_i^*)\Phi_m(\Delta\lambda_i^*) & -\sum_{i=1}^{N} \Phi_1(\Delta\lambda_i^*)\frac{C_2}{\lambda_i T_{ref}} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
-\sum_{i=1}^{N} \Phi_0(\Delta\lambda_i^*)\frac{C_2}{\lambda_i T_{ref}} & -\sum_{i=1}^{N} \Phi_1(\Delta\lambda_i^*)\frac{C_2}{\lambda_i T_{ref}} & \dots & -\sum_{i=1}^{N} \Phi_m(\Delta\lambda_i^*)\frac{C_2}{\lambda_i T_{ref}} & \sum_{i=1}^{N} \left(\frac{C_2}{\lambda_i T_{ref}}\right)^2
\end{pmatrix}
\tag{3.33}
$$

In order to illustrate the effect of the orthogonalization onto the final estimation, a simple use-case has been experimented. The Eq. (3.20) is computed for $\Delta\lambda_i = \{2.5\mu m, 3\mu m, 4\mu m, 5\mu m, 8\mu m, 10\mu m, 12\mu m\}$, a given temperature profile (represented in Fig. 3.4), random emissivity values, $\theta_{sensor} = \theta_{object} = 0$ and $r = 1$. Once the data have been simulated, the emissivities and temperatures are found back with the polynomial model presented in Eq. (3.30) and the Tchebychev polynomials of first kind applied to the system Eq. (3.32). The systems are solved with the constraint : $ln(\epsilon_{\Delta\lambda_i}) \leq 0, \quad \forall i \in [1, N]$. Moreover, the emissivity profile is generated randomly. The purpose of this is to test the robustness of the fitting on a random profile. The inversion becomes more difficult when the standard deviation of the emissivity profile is larger, due to the important discontinuity that appears and the small amount of band measurements.
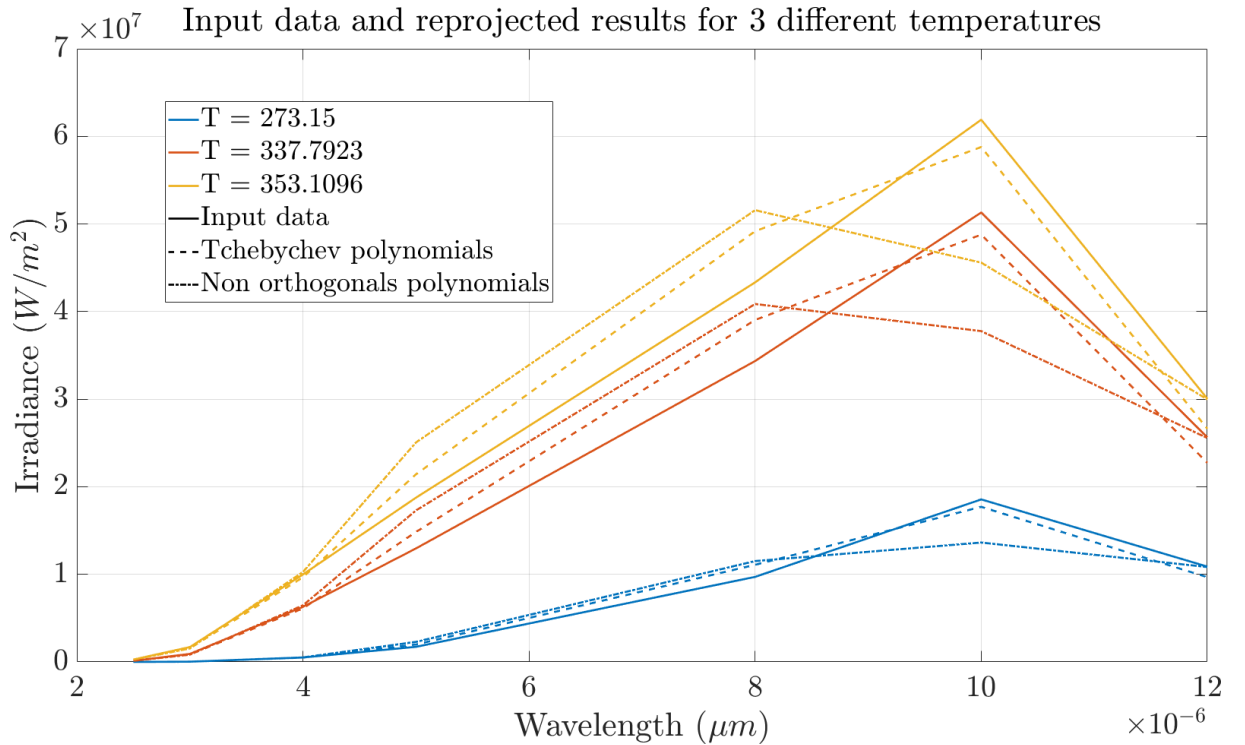
Figure 3.3: Original irradiance at sensor compared to the irradiance with the estimated emissivity and temperature

The Fig. 3.3 shows the simulated irradiance in comparison to the irradiance computed with the estimated parameters. One can see that this reprojection error can be important since the algorithm stops when a local minimum has been found. However, the Fig. 3.4 shows that the estimation is slightly different between the Tchebychev basis and the monomial basis. The orthogonal basis tends to be more robust than the monomial's one, as expected.
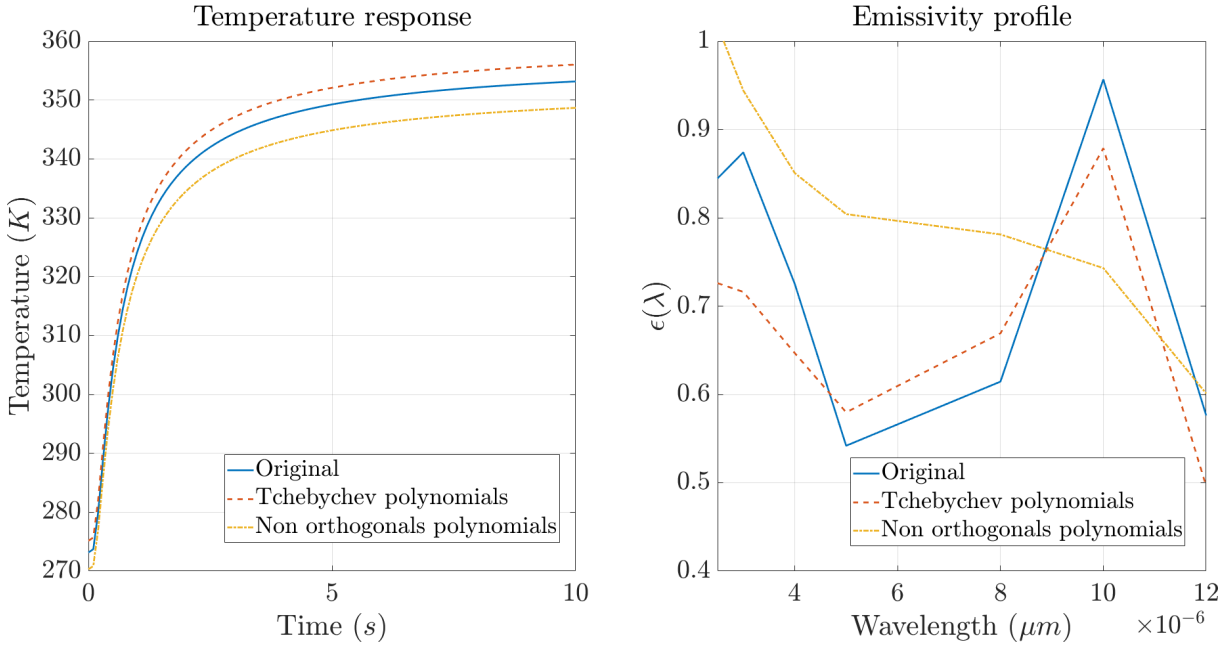
Figure 3.4: Found temperature and emissivity profiles

However, the results are not quite satysfying in the simple test case, even when using an orthogonal basis. In fact, the polynomial approximation is also depending on the temperature estimation sensitivity [102].

Splines   In the previous approximation procedures, the emissivity was evaluated to a single identical polynomial. It is possible instead to split the domain into different region and define a polynomial's function piecewise. Such function is called a spline and is often prefered to the previous methods for interpolation problems. Let now $S_j(\lambda), j \in [1, N-1]$ be the polynoms that interpolate the $\ln(\epsilon_{\Delta\lambda_i}), i \in [1, N]$ knots values and m the degree of the polynoms.

$$S_j = \sum_{k=0}^{m} \alpha_{j,k}\Phi_k(\lambda^*), \forall j \in [1, N-1] \tag{3.34}$$

48
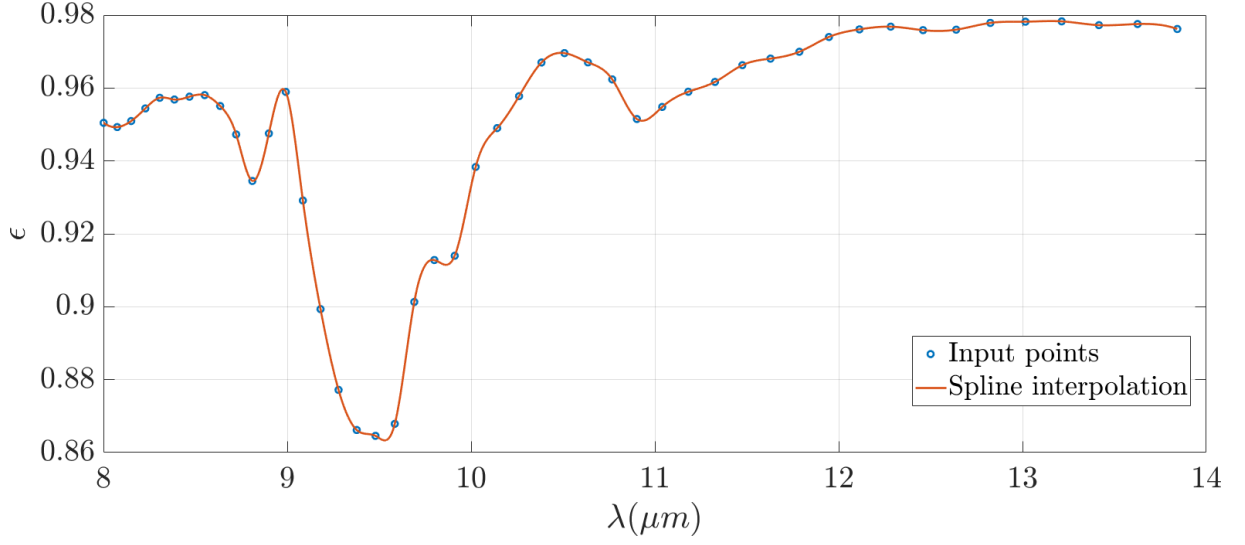
Figure 3.5: Cubic spline interpolation with quadratic boundaries conditions.

To satisfy the knot's equality :

$$(\mathcal{C}_0) = \begin{cases} S_1(\Delta\lambda_1^*) = \ln\left(\epsilon_{\Delta\lambda_1^*}\right) \\ S_1(\Delta\lambda_2^*) = \ln\left(\epsilon_{\Delta\lambda_2^*}\right) \\ S_2(\Delta\lambda_2^*) = \ln\left(\epsilon_{\Delta\lambda_2^*}\right) \\ S_2(\Delta\lambda_3^*) = \ln\left(\epsilon_{\Delta\lambda_3^*}\right) \\ \dots \\ S_{(N-1)}(\Delta\lambda_N^*) = \ln\left(\epsilon_{\Delta\lambda_N^*}\right) \end{cases} \tag{3.35}$$

The Eq. (3.35) implies $2(N-1)$ equations, to satisfy the knots equalities. In order to be able to solve the system and reduce the number of unknowns, one needs to impose conditions to the system. In the case of cubic interpolation ($m = 3$), the number of unknowns is : $4(N-1)+1$ (4 coefficients for $N-1$ polynoms and the temperature). The constraint of equality of the first derivatives at knot's points can be added, to smooth the curves and avoid discontinuities. Therefore, we have the second equation system:

$$(\mathcal{C}_1) = \begin{cases} S_1'(\Delta\lambda_2^*) = S_2'(\Delta\lambda_2^*) \\ S_2'(\Delta\lambda_3^*) = S_3'(\Delta\lambda_3^*) \\ \dots \\ S_{N-2}'(\Delta\lambda_{N-1}^*) = S_{N-1}'(\Delta\lambda_{N-1}^*) \end{cases} \tag{3.36}$$

Moreover, one can also add the same constraint for the second derivatives:

$$(\mathcal{C}_2) = \begin{cases} S_1''(\Delta\lambda_2^*) = S_2''(\Delta\lambda_2^*) \\ S_2''(\Delta\lambda_3^*) = S_3''(\Delta\lambda_3^*) \\ \dots \\ S_{N-2}''(\Delta\lambda_{N-1}^*) = S_{N-1}''(\Delta\lambda_{N-1}^*) \end{cases} \tag{3.37}$$

Finally, $(\mathcal{C}_0) + (\mathcal{C}_1) + (\mathcal{C}_2)$ constraints lead to $4N - 6$ equations, for $4(N-1) + 1$ unknowns, an undetermined system. To overcome this, boundaries constraints can be added. For example, a quadratic boundary constraint and a second derivative equal to zero for the first and last polynomials will result in a overdetermined system. Let modify the first and last polynomial to be quadratic. Eq. (3.34) becomes :

$$S_j(\lambda^*) = \begin{cases} \sum_{k=0}^2 \alpha_{j,k}\Phi_k(\lambda^*) & \text{if } j \in \{1, N-1\} \\ \sum_{k=0}^3 \alpha_{j,k}\Phi_k(\lambda^*) & \text{if } j \in [2, N-2] \end{cases} \tag{3.38}$$

We also add their first and second derivatives to be equal at the boundaries (called "Periodic Spline"):

$$\begin{cases} S_1'(\Delta\lambda_1^*) = S_{N-1}'(\Delta\lambda_N^*) \\ S_1''(\Delta\lambda_1^*) = S_{N-1}''(\Delta\lambda_N^*) \end{cases} \implies \begin{cases} \alpha_{1,1} = \alpha_{N-1,1} \\ \alpha_{1,2} = \alpha_{N-1,2} \end{cases} \tag{3.39}$$

In that case, the number of unknowns becomes $4N - 5$ for $4(N - 1)$ equations. The system Eq. (3.32) to be solved can be updated with the constraints :

$$
\begin{cases}
\Gamma^* = \begin{pmatrix} \ln\left(\gamma_{\Delta\lambda_1}\right) \\ \ln\left(\gamma_{\Delta\lambda_2}\right) \\ \ln\left(\gamma_{\Delta\lambda_2}\right) \\ \vdots \\ \ln\left(\gamma_{\Delta\lambda_N}\right) \end{pmatrix} = \underbrace{\begin{pmatrix} \Phi_0(\Delta\lambda_1^*) & \Phi_1(\Delta\lambda_1^*) & \Phi_2(\Delta\lambda_1^*) & 0 & 0 & \dots & 0 & \frac{-C_2}{\Delta\lambda_1 T_{\text{ref}}} \\ \Phi_0(\Delta\lambda_2^*) & \Phi_1(\Delta\lambda_2^*) & \Phi_2(\Delta\lambda_2^*) & 0 & 0 & \dots & 0 & \frac{-C_2}{\Delta\lambda_2 T_{\text{ref}}} \\ 0 & 0 & 0 & \Phi_0(\Delta\lambda_2^*) & \Phi_1(\Delta\lambda_2^*) & \dots & 0 & \frac{-C_2}{\Delta\lambda_2 T_{\text{ref}}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \Phi_2(\Delta\lambda_N^*) & \frac{-C_2}{\Delta\lambda_N T_{\text{ref}}} \end{pmatrix}}_{M_0} \begin{pmatrix} \alpha_{1,0} \\ \alpha_{1,1} \\ \vdots \\ \alpha_{N-1,2} \\ \frac{T_{ref}}{T'} \end{pmatrix} \\[20pt]
0 = \underbrace{\begin{pmatrix} \Phi_0'(\Delta\lambda_2^*) & \Phi_1'(\Delta\lambda_2^*) & \Phi_2'(\Delta\lambda_2^*) & -\Phi_0'(\Delta\lambda_2^*) & \dots & 0 & 0 \\ 0 & 0 & 0 & \Phi_0'(\Delta\lambda_3^*) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & 0 & \dots & -\Phi_3'(\Delta\lambda_{N-1}^*) & 0 \end{pmatrix}}_{M_1} \begin{pmatrix} \alpha_{1,0} \\ \alpha_{1,1} \\ \vdots \\ \alpha_{N-1,2} \\ \frac{T_{ref}}{T'} \end{pmatrix} \\[20pt]
0 = \underbrace{\begin{pmatrix} \Phi_0''(\Delta\lambda_2^*) & \Phi_1''(\Delta\lambda_2^*) & \Phi_2''(\Delta\lambda_2^*) & -\Phi_0''(\Delta\lambda_2^*) & -\Phi_1''(\Delta\lambda_2^*) & \dots & 0 & 0 \\ 0 & 0 & 0 & \Phi_0''(\Delta\lambda_3^*) & \Phi_1''(\Delta\lambda_3^*) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -\Phi_2''(\Delta\lambda_{N-1}^*) & 0 \end{pmatrix}}_{M_2} \begin{pmatrix} \alpha_{1,0} \\ \alpha_{1,1} \\ \vdots \\ \alpha_{N-1,2} \\ \frac{T_{ref}}{T'} \end{pmatrix} \\[20pt]
0 = \underbrace{\begin{pmatrix} 0 & 1 & 0 & \dots & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & -1 & 0 \end{pmatrix}}_{M_3} \begin{pmatrix} \alpha_{1,0} \\ \alpha_{1,1} \\ \vdots \\ \alpha_{N-1,2} \\ \frac{T_{ref}}{T'} \end{pmatrix}
\end{cases}
\tag{3.40}
$$

The matrices M_0, M_1 and M_2 and M_3 can be merged together to form the following linear system:

$$
\begin{pmatrix} \Gamma^* \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ M_2 \\ M_3 \end{pmatrix} \begin{pmatrix} \alpha_{1,0} \\ \alpha_{1,1} \\ \vdots \\ \alpha_{N-1,2} \\ \frac{T_{ref}}{T'} \end{pmatrix}
\tag{3.41}
$$

Using splines interpolation has the advantage of adding continuity constraints to the interpolating function. Due to the fact that the emissivity is a continuous function in wavelength, splines interpolation should be preferred to the simple polynomial fitting.

Multi-temperature optimization

One way to overcome the undetermined system is to perform multiple measurements, at multiple temperatures, assuming that the emissivity will not change. In principle, such assumption makes the equations' system solvable by removing one temperature variable. This multi-temporal method seems attractive in theory. However, this method is highly sensitive to measurement noise due to the constant emissivity hypothesis between the two measurements [117, 102].

> **Note**
>
> Similarly, multi-temperature has been used in the Remote Sensing field by assuming that the emissivity is invariant at two times. However, this method requires accurate geometry registration to get observation under similar angles (which favors geostrationary satellite data). Moreover, it requires multi-temporal data and important temperature gradient between the measurements.

Grey body hypothesis

The grey body emissivity method tackles also the equation system unknown issue by assuming that on two given area of the spectrum, the emissivity is constant. Such flat region in the emissivity profile is actually a strong requirement and is not applicable to most of the observed materials.

> **Note**
>
> Similarly, grey body methods have been used in the Remote Sensing field [15] but for the same reason, the assumption is not satisfyed in most cases.

3.1.3   Bayesian methods

In this section are presented the Bayesian inference based methods. Those statistical methods are based on the Bayes' theorem, used to update the probability for a hypothesis as more information becomes available. With the increase of computation means, those methods have become common. More details on Bayesian inference are given on Chapter 6.

Dynamic filtering with Kalman and particle filters

The Kalman Filter applied to the emissivity / temperature retrieval has been used in the literature, in [121, 185] with SEVIRI data for example. In particular [121] formulates a four-dimensional

approach (*i.e.* space-time) but applies only a temporal constraint in their article. The Kalman filter is used conjointly with the $\sigma - $ SEVIRI model based on look-up tables. Such method is specific to the SEVIRI measurements and cannot be applied directly to the *in-situ* IRT measurements case. However, the Kalman Filter appears to be an interesting approach to perform parameters estimation on the emissivity/temperature retrieval problem. In particular, the temporal approach for long-term monitoring through a Kalman filter is used in Chapter 6 on which more details on the Kalman Filter are given. Likewise, particle filters which are also Bayesian filters have been used in the Remote sensing field. However, particle filter has not been particularly used for LST / LSE simultaneous retrieval to our knowledge. This can be particularly explained by the fact that satellite measurements represent a huge amount of data for which particle filtering may not be suitable due to its complexity. Yet, particle filtering and ensemble kalman filters have been used for data assimilation [118] or data downscaling purposes [124]. The particle filter is also more described in Chapter 6.

Monte-Carlo Markov Chains (MCMC)

The Monte-Carlo Markov Chains (MCMC) methods are meta-heuristic methods that sample probability distributions. The basic idea is to be able to browse a Markov Chain that will have the targeted distribution as equilibrium distribution. Such method can be employed for the temperature and emissivity retrieval [10]. This method has been implemented for comparison purposes in this thesis and is therefore more detailed here. In this case, knowing the measurement vector $\boldsymbol{\gamma}$, the objective is to determine the probability distributions of $p(T|\boldsymbol{\gamma})$ and $p(\boldsymbol{\epsilon}|\boldsymbol{\gamma})$. For this, let suppose that the emissivity $\boldsymbol{\epsilon} = (\epsilon_{\lambda_i})_{1 \leq i \leq N}$ and temperature T are independent, and follow a given prior:

$$p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\mu_\epsilon}, \boldsymbol{\Sigma_\epsilon})$$
$$p(T) = \mathcal{U}(T_{\min}, T_{\max})$$

(3.42)

The radiative equation used is derived from Eq. (2.50) where the optical transmission is assumed equal to 1 [20] and the sun contribution is included in the environmental contribution. A difference is made by considering the variables as stochastic and therefore adding a measurement noise $\boldsymbol{\eta}$:

$$L_{\lambda,\vec{d},tot}^{(i,j)}(T) = \tau_{atm}\left(\epsilon_\lambda L_\lambda^\circ(T_{obj}^{(i,j)}) + (1-\epsilon_\lambda)(L_{\lambda,env})\right) + (1-\tau_{atm})L_{\lambda,atm}^\circ + \boldsymbol{\eta}$$

(3.43)

This equation can be written:

$$\boldsymbol{\gamma} = \boldsymbol{D_T}\boldsymbol{\epsilon} + c + \boldsymbol{\eta}$$

(3.44)

where $\boldsymbol{D_T} = \text{diag}(\tau_{atm}\left(L_\lambda^\circ(T_{obj}^{(i,j)}) - L_{\lambda,env}\right)$ and $c = \tau_{atm}L_{\lambda,env} + (1-\tau_{atm})L_{\lambda,atm}^\circ$. Finally, it is possible to derive the conditional distributions with this equation. Since $\boldsymbol{\gamma}$ is a linear function of a Gaussian variable $\boldsymbol{\epsilon}$ plus noise $\boldsymbol{\eta}$, then the distribution of $\boldsymbol{\epsilon}$ knowing $\boldsymbol{\gamma}$ has also to be Gaussian [10]. By

using Gaussian conditioning:

$$p(\boldsymbol{\epsilon}|T, \boldsymbol{\gamma}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\epsilon}, \tilde{\boldsymbol{\Sigma}}_{\epsilon})$$
$$\tilde{\boldsymbol{\mu}}_{\epsilon} = \boldsymbol{\mu}_{\epsilon} + \boldsymbol{\Sigma}_{\epsilon} \boldsymbol{D}_{\boldsymbol{T}} \left(\boldsymbol{D}_{\boldsymbol{T}} \boldsymbol{\Sigma}_{\epsilon} \boldsymbol{D}_{\boldsymbol{T}} + \boldsymbol{\Sigma}_{\boldsymbol{\eta}}\right)^{-1} \left(\boldsymbol{\gamma} - \boldsymbol{D}_{\boldsymbol{T}} \boldsymbol{\mu}_{\epsilon} - c\right) \quad (3.45)$$
$$\tilde{\boldsymbol{\Sigma}}_{\epsilon} = \boldsymbol{\Sigma}_{\epsilon} - \boldsymbol{\Sigma}_{\epsilon} \boldsymbol{D}_{\boldsymbol{T}} \left(\boldsymbol{D}_{\boldsymbol{T}} \boldsymbol{\Sigma}_{\epsilon} \boldsymbol{D}_{\boldsymbol{T}} + \boldsymbol{\Sigma}_{\boldsymbol{\eta}}\right)^{-1} \boldsymbol{D}_{\boldsymbol{T}} \boldsymbol{\Sigma}_{\epsilon}$$

The temperature distribution is determined up-to proportionality:

$$p(T|\boldsymbol{\epsilon}, \boldsymbol{\gamma}) \propto p(\boldsymbol{\gamma}|T, \boldsymbol{\epsilon})p(T|\boldsymbol{\epsilon})$$
$$= p(\boldsymbol{\gamma}|T, \boldsymbol{\epsilon})p(T) \quad (3.46)$$

Since the joint density *a posteriori* of $\boldsymbol{\epsilon}$ and T are difficult to derive and interpret, a Gibbs sampler will allow to sample the probability of $p(\boldsymbol{\epsilon}, T|\boldsymbol{\gamma})$ based on the conditional probabilities $p(\boldsymbol{\epsilon}|T, \boldsymbol{\gamma})$ and $p(T|\boldsymbol{\epsilon}, \boldsymbol{\gamma})$.

Let π a distribution over a universe Ω. Drawing randomly an element of Ω following π can be done through a Gibbs sampler [66].

Algorithm 1: Pseudo code for a *Gibbs sampler*

Function Gibbs $(p(T), p(\boldsymbol{\epsilon}|T_{i-1}, y), p(T|\boldsymbol{\epsilon}_i, \boldsymbol{\gamma}), N_{mcmc}, N_{burn})$

$\boldsymbol{\epsilon}(0..N_{mcmc}) \leftarrow 0$;
$T(0..N_{mcmc}) \leftarrow 0$;
Draw $T(0) \sim p(T)$;
For $i \leftarrow 1..N_{mcmc}$
  Draw $\boldsymbol{\epsilon}(i) \sim p(\boldsymbol{\epsilon}|T(i-1), \boldsymbol{\gamma})$;
  Draw $T(i) \sim p(T|\boldsymbol{\epsilon}(i), \boldsymbol{\gamma})$;
Remove N_{burn} samples from T and $\boldsymbol{\epsilon}$;

However, since the distribution $p(T|\boldsymbol{\epsilon}, \boldsymbol{\gamma})$ is difficult to compute, a slice-sampling algorithm [72] is used to sample from it.

Algorithm 2: Pseudo code for a *slice sampling*

Function Slice-sampling(f)

Choose x such as $f(x) > 0$;
Repeat *once*
  Draw $y \sim \mathcal{U}(0, f(x))$;
  Plot an horizontal line at y;
  Sample the point (x, y) with the edge created at the intersection of $f(x)$ and the
    horizontal line;

This type of sampling is called a slice-within-Gibbs sampling. Even though this method provides quite accurate results and its convergence is ensured in this case, the required computation is heavy

and processing time important. Such method can therefore be used for post-process or handling a small amount of data but the complexity of it makes it difficult to use on full resolution IR images and real-time processing. A comparison of MCMC method with other ones is done in Chapter 6.

Artificial Neural Network (ANN)

Artificial neural network can robustly perform non-linear and parallel computations and have therefore been largely used for remote sensing [6, 117, 5, 146]. A neuron represents a particular transfer function that will transform the inputs to outputs according to a particular rule (which can be summation, thresholding *etc.*). Neurons are then connected between them with different possible patterns. Through statistical (usually Bayesian) learning methods based on a particular data-set, interconnected neurons store their acquired knowledge by updating their weight value (this behavior replicates the synaptic plasticity of biological networks). However, such methods are facing multiple difficulties such as overfitting which occurs when a given model is too dependent on training data and may not correspond to the actual targeted model. As a consequence, the selection of training data and their characteristics are essential for the method to give accurate results. Furthermore, the architecture used for the network is also one of the key-point [177]. Unfortunately, there is no standard way to help in the choice of those parameters. Most of the time, a subjective trial and error approach is performed to set the network. In addition, the retrieval LST/LSE process cannot be well controlled and improving the output is complex due to the difficulty to interpret the network weights.

3.1.4   Synthesis

A review of the different emissivity / temperature separation methods have been made. Generally speaking, the algorithms can be distinguished into three main families:

- Purely mathematics algorithms where a mathematical model aims at simplifying the equation
- Derived from physical characteristics of the measurements or the observed object
- Derived from a statistical prior on the measurement or the observed object

It has been shown that adding more wavelengths to the system does not help to solve the equations' system. The sole introduction of an emissivity polynomial model introduces high correlations and therefore leads to poor results [102]. However, using some prior knowledge about the observed materials is facing the mixed pixel issue: the sensor does not see a unique material but a combination of materials with their own properties. The resulting measurement is a new equivalent material with its own equivalent thermal properties. It is important to note also that some methods that works well in the Remote Sensing field cannot be applied for *in-situ* infrared thermography measurements. As a conclusion, the Bayesian approaches seem to offer interesting prospects, particularly due to the low priors that can be put on the system. One difficulty of those methods is to adjust well the priors. One drawback also can be the computation time of such methods, particularly when meta-heuristics are

involved.

> **Note**
>
> Other methods than the presented one exist. For example, active methods but out-of-scope in this application (see previous chapter). It is also possible to put a reference target with a known emissivity on the field. However, for long-term practical applications of infrastructures, the reference emissivity may drift through time leading to inaccuracies. It may also been difficult to place the target so that the reference measurement is not too much influenced by its surroundings or modifies the dynamic of the measure.

3.2 Solving the radiative transfer equation

The equations presented in Chapter 2 can be used as a forward model to simulate the radiative exchanges that occur in a 3D scene. Such simulation tool can be useful for sizing infrared thermography on site before any exploitation and also generate data to test or train algorithms. As a matter of fact, recent buildings constructions come with a building information modeling (BIM) that contains a model of the constructed building, usable for simulation purpose. The presented RTE (Eq. (2.32)) can be solved numerically as in [180] where diffusion, convection and radiative transfers are coupled on a discretized domain in space and angles. However, even if such methodology provides accurate results relatively to its discrete domain, it is also computationally intensive and unfortunately not suitable for fast computation on large refined 3D models. Instead of using the full equation governing the radiation beam in a participating medium Eq. (2.32), the radiosity equation reminded in Eq. (2.48) for the particular case of nonparticipating mediums and diffuse surfaces is used.

$$J_\lambda(x) = M_\lambda(x) + \frac{\rho_\lambda(x)}{\pi} \int_S F(x, x^{'}) J_\lambda(x^{'}) \mathrm{d}S(x^{'}) \tag{3.47}$$

> **Note**
>
> Eq. (3.47) is called a Fredholm integral equation of the second kind.

3.2.1 Finite element approach

Analytical solutions to Eq. (2.48) exist in particular cases only. Instead, a numerical solution can be computed by decomposing the radiosity function on a given basis. Such approximation is called a Galerkine method, on which the domain of study is meshed and the estimated function is interpolated

on this mesh. The Galerkine methods are a set of methods that transform a continuous problem to a discrete one and is often used in finite elements methods. Finite element method needs to make the study domain discrete. In the radiosity equation (nonparticipating medium), it leads to cut the domain into a finite number of elementary surfaces (mesh). Let $S \subset \mathbb{R}^3$ be the study domain and $\mathcal{L}^2(S)$ the space of square-integrable functions (Lebesgue space):

$$\mathcal{L}^2(S) = \left\{ f \in S \quad \middle| \quad \int_S |f(x)|^2 \mathrm{d}S(x) < \infty \right\} \tag{3.48}$$

That is to say, the function is (almost) finite everywhere in the domain. Let define the inner product on $\mathcal{L}^2(S)$ $\langle \cdot, \cdot \rangle$:

$$\forall u, v \in \mathcal{L}^2(S), \langle u, v \rangle = \int_S uv \mathrm{d}S \tag{3.49}$$

Let now $V = H^m(S)$ be an Hilbert space of functions of $\mathcal{L}^2(S)$ with the inner product Eq. (3.49). Let also $\|\cdot\|_V = \|\cdot\|_{H^m}$. A weak form can be derived from Eq. (3.48). Let $v \in V$:

$$\int_S J_\lambda(x)v(x)\mathrm{d}S(x) = \int_S M_\lambda(x)v(x)\mathrm{d}S(x) + \int_S \frac{\rho_\lambda(x)}{\pi} \int_S F(x, x^{'})J_\lambda(x^{'})v(x)\mathrm{d}S(x^{'})\mathrm{d}S(x) \tag{3.50}$$

A bilinear form $a(\cdot, \cdot)$ and a linear form $L(\cdot)$ defined respectively over $\mathcal{L}^2(S) \times \mathcal{L}^2(S)$ and $\mathcal{L}^2(S)$ such as:

$$a(J_\lambda, v) = \langle J_\lambda, v \rangle_V - \int_S \frac{\rho_\lambda(x)}{\pi} \int_S F(x, x^{'})J_\lambda(x^{'})v(x)\mathrm{d}S(x^{'})\mathrm{d}S(x)$$
$$L(v) = \langle M_\lambda, v \rangle_V \tag{3.51}$$

Such writing gives the following variationnal form: finding $v \in \mathcal{L}^2(S)$ such as:

$$a(J_\lambda, v) = L(v), \quad \forall v \in \mathcal{L}^2(S) \tag{3.52}$$

In order to use the Lax-Milgram theorem, let show that $a(\cdot, \cdot)$ is continuous on $V \times V$ and V-elliptical and that $L(\cdot)$ is a linear continuous form on V. $F(x, x^{'})$ is square integrable. By using Cauchy-Schwartz and triangular inequality, it follows that:

$$\forall u, v \in V, \quad \|a(u, v)\| \leq 2\|u\|_V \|v\|_V$$
$$\forall u \in V, \quad \|L(u)\| \leq \|M_\lambda\|_V \|u\|_V \tag{3.53}$$

and therefore $a(\cdot, \cdot)$ is continuous on $V \times V$ and $L(\cdot)$ is continuous on V. From a physical point of view, since the emissivity ranges in the interval $]0; 1[$, thus there exists a positive number ϵ_0 such as [122]:

$$\forall x \in S, \quad \epsilon(x) \geq \epsilon_0 > 0 \tag{3.54}$$

Which shows that $a(\cdot, \cdot)$ is V-elliptical:

$$\forall v \in V, a(v,v) \geq \epsilon_0 \|v\|_V^2 \tag{3.55}$$

As a conclusion, Lax-Milgram theorem ensures that a solution $v \in \mathcal{L}^2(S)$ of Eq. (3.52) exists and is unique.

Based on this weak formulation, let $V_h \subset \mathcal{L}^2(S)$ a finite-dimensional subspace. And let suppose $(\phi_i)_{1 \leq i \leq N}$ a basis function of V_h. The study domain is now a subset of $H^1(S)$ made of polynomial functions defined on every element of the mesh. Based on the Galerkine formulation, it is possible to approximate the continuous function J_λ to a discrete one $\tilde{J}_\lambda \in V_h$ expressed as:

$$a(\tilde{J}_\lambda, v) = b(v), \quad \forall v \in V_h \tag{3.56}$$

Since $\tilde{J}_\lambda \in V_h$, $\tilde{J}_\lambda$ can be expressed with the basis functions of V_h:

$$J_\lambda \approx \tilde{J}_\lambda = \sum_{j=1}^{n} \tilde{J}_{\lambda,j} a(\phi_j, \phi_i) = b(\phi_i) \tag{3.57}$$

With Eq. (3.51) and Eq. (3.57):

$$\sum_{j=1}^{n} \tilde{J}_{\lambda,j} \left(\langle \phi_j, \phi_i \rangle_{V_h} - \int_S \frac{\rho_\lambda(x)}{\pi} \int_S F(x,x')\phi_j(x')\phi_i(x)\mathrm{d}S(x')\mathrm{d}S(x) \right) = \int_S \phi_i(x)M_\lambda(x)\mathrm{d}S(x) \tag{3.58}$$

For orthonormal basis functions:

$$\langle \phi_j, \phi_i \rangle_{V_h} = \delta_{ij}$$

$$\int_S \frac{\rho_\lambda(x)}{\pi} \int_S F(x,x')\phi_j(x')\phi_i(x)\mathrm{d}S(x')\mathrm{d}S(x) = \rho_i F_{i \to j} \tag{3.59}$$

$$\int_S \phi_i(x)M_\lambda(x)\mathrm{d}S(x) = M_{\lambda,i}$$

where $F_{i \to j}$ are the form factor defined in the previous chapter, ρ_i is the average reflectance of element i and $M_{\lambda,i}$ the average emission of element i. For an opaque material it follows that:

$$J_{\lambda,i} = M_{\lambda,i} + (1 - \epsilon_{\lambda,i}) \sum_{j=1}^{n} F_{i \to j} J_{\lambda,j} \tag{3.60}$$

Eq. (3.60) is the discrete version of the radiosity equation. In the following, methods for solving this equation will be reviewed. In particular, even if the origin of this radiosity equation is derived from the thermal transfer theory, it has been widely used in a very different field: graphics rendering [75]. In fact, if radiative heat transfer is interesting in the propagation of thermal radiation through sur-

faces, the equations are analogous for light propagation to render diffuses surfaces. Therefore, the following methods are presented from this former field, even though those methods could be applied from a more finite element point of view. In this thesis, the mathematical quantities refer to radiometry and not photometry. The photometry aims at measuring the light in terms of human's perception whereas radiometry measure radiant energy (including light) in terms of absolute power. Therefore, terms radiance, radiant flux and radiant intensity are used in place of luminance, luminous flux and luminous intensity for consistency. Please note that the two fields aim at different goals. When thermal specialists look for an accurate thermal solution of radiative exchanges equations, computer graphics specialists look for a fast solution as real as possible (through the following presented methods). In this thesis, the aim is to develop a simulation tool that gathers those two approaches.

3.2.2   3D image synthesis

Image synthesis history starts during the end of the 1950s for automotive and aeronautic researches purposes. First electronic pens appeared in the first 1960s into an IBM computer so drafting the first mechanical parts through a computer for General Motors was possible [94]. Further development have been performed during the 1960s in order to develop computer aided design (CAD) programs for 2D and 3D production drawings. The Computer Graphics term appeared in 1961 and is attributed to Boeing art's CAD director William Fetter. In the late 1960s, Utah university specialized in the 3D modeling which aimed at forming a computer model of an object's shape. In beginning of 1970s, Martin Newell at Utah university proposed a 3D model of what will become the famous Utah teapot, which was an object challenging to model but still possible with curves and reflective surfaces. In the late 1970s, 3D computer graphics software appeared for home computers such as *3D Art Graphics* in the Apple II. The development of 3D computer graphics were slowdown during the 1970s due to the high hardware cost but with the apparition of home computers, the 1980s are an upsurge time for computer graphics. But it is only in the 1990s that image synthesis and 3D become trendy and now used in many application thanks to the hardware development of graphical processing units (GPU): scientific simulations, CAD, art but also video games. Nowadays, the many algorithms and various applications exist for image synthesis and it is a constant evolving field mixing video games, scientific simulation and animation movies. In this thesis, the objective is to develop a scientific simulation software to render infrared images from *in-situ* 3D scenes. The light transport equations used in the image synthesis field are similar to the radiation transfer equation. However, one of the main differences between the radiation theory and the 3D image synthesis is the initialization condition. In fact, in most image synthesis algorithms, the emissivity of the surfaces are initialized as follows:

$$\epsilon = \begin{cases} 0, & \text{if it is not a light object} \\ 1, & \text{if it is a light object} \end{cases} \tag{3.61}$$

Then, the reflectivity of the surface is used to reflect light. Even if emissive objects are sometimes considered, it represents only a - usually - small part of the scene. At the opposite, in the radiative transfer field all surfaces are supposed to emit which is a non negligible fact. A bibliographic overview of image synthesis is therefore presented in this section.

3D image synthesis algorithms can be classified into two categories: local illumination and global illumination. Local illumination methods as their name stipulate consider only the local parameters of the scene (usually the probe lights sources, meshes material properties and normal) to compute the apparent luminosity of the meshes. Those algorithms when executed on material hardware are rendered in real-time really efficiently. However those algorithms (Lamberts, Gouraud, Phong, Cook-Torrance, Kajiya, *etc.*) are only based on the aesthetic of the rendering and computed empirically. Those methods take into account direct lightning only and therefore exchanges between the elements other than the light $\rightarrow$ object interactions are not considered. This can even lead to inconsistencies with the first thermodynamic law.

At the opposite, global illumination (GI) based algorithms compute both direct and indirect lightning that take into account the incoming radiation (or light) and also the different recursives reflections with the surrounding environment. Those algorithms are mostly based on the radiosity equation presented below.

> **Note**
>
> Most recent developed algorithms are now called PBR for physically based rendering (the 2004 reference book has now its 3rd free online edition [142]) and tend to respect as most as possible physics' laws to make the rendering as real as possible.

GI algorithms are used in 3D image synthesis and take into account not only the direct illumination of light sources but also the scattered and reflected light form other patches present in the scene. Therefore, each elementary surface's illumination cannot be computed individually and the system that modelize all the radiative exchanges can only be rendered globally. Many applications are involved such as animation movies (*e.g.* Disney, Pixar), special effects or video games for texture rendering. Two main types of algorithms exists [163]:

- Ray tracing based methods (ray tracing, photon mapping, metropolis light transport, *etc.*)
- Direct computation of the radiosity's coefficients equation (radiosity method, finite elements, optimization, *etc.*)

A review of those methods are presented in this section.

60

Ray tracing

Ray tracing methods principle is to browse the light path backward from reality: a ray is thrown from the camera toward the 3D scene and bounces until it reaches the light. This behavior is the opposite than reality where the ray goes from the light to the camera. Each time a ray encounter an obstacle, its energy can be distributed according to the different physical phenomena that occur on the object (absorption, reflection, transmission, diffraction, *etc.*). The first algorithm was developed in 1968 by Arthur Appel from IBM [8]. The idea was to throw a ray from the eye, one per pixel and stop once the closest object blocked the ray. This led to determine the shading of an object. Then, in 1979, the algorithm was improved by performing multiple bounces and casting new rays from an incoming ray that represents the reflection, refraction and shadow [190]. Then, improvements on the method and the consideration of many other phenomena were made on the literature [64]. A few methods for ray tracing based rendering are presented. Basic ray tracing algorithm are straightforward to implement nowadays (for simple considerations of physical phenomena) and can provide realistic results (see Fig. 3.6 for an advanced result).

Figure 3.6: Ray tracing images can be realistics, example with the POV-Ray software rendering

Moreover, only a geometrical description of the scene is required, and may not need to be discretized. Due to the independence of the rays, the algorithm can be parallelized. However the main drawbacks are that the rendering equation is not used in basic ray tracing methods and is therefore not considered as photo-realistic since the physic may not be respected. Also, the ray tracing algorithm produces noise due to the random sampling which reduces with the number of particles thrown per pixel (see Fig. 3.7). Finally, the rendered image is view-dependent for standard ray tracing which means that for any movement of the camera in static scenes, the algorithm needs to be computed again. This behavior enables the methods to compute the solution quickly. In fact, the rays are thrown from the pixels of the camera only. View independent ray tracing methods does exist, especially by rendering the scene from multiple views. However, such fix induces a high increase of computation time [64].
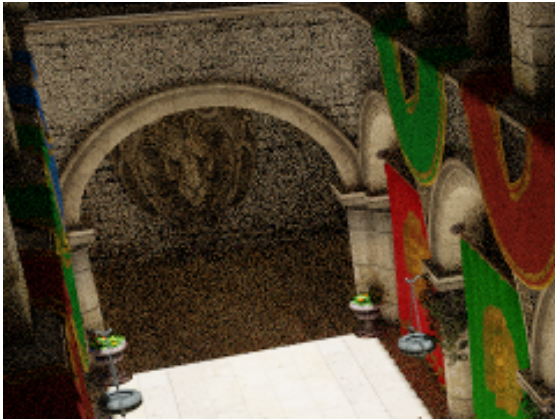
(a) Ray-traced Sponza sample scene with Cycles renderer (zoom).

(b) Ray-traced Sponza sample scene with Cycles renderer after denoising (zoom).

Figure 3.7: Example of ray-traced rendering with the Sponza scene and Cycles open source renderer.

Monte-Carlo ray tracing Monte-Carlo methods are a set of methods that compute a given numerical value based on stochastic processes. The numerical value is approximated by running multiple simulation.

> **Note**
>
> A simple example of this method is to approximate π value. Let $\mathcal{C}$ be a circle of $\mathbb{R}^2$ of radius r. Let now S be the square in which $\mathcal{C}$ is inscribed, x and y drawn from a uniform distribution in the square. The probability that the point $p(x, y)$ lies into $\mathcal{C}$ can be given by the ratio between the area of the square to the area of the circle:
>
> $$P(p \in \mathcal{C}) = \frac{A_S}{A_{\mathcal{C}}} = \frac{4r^2}{\pi r^2} \tag{3.62}$$
>
> Which means that $\pi = 4P(p \in \mathcal{C})$. By drawing multiple times p into the square and counting the number of occurrences that p lied into $\mathcal{C}$, then π can be approximated (see Fig. 3.8). Of course, this method is actually not efficient but it serves Monte-Carlo principle demonstration purposes.
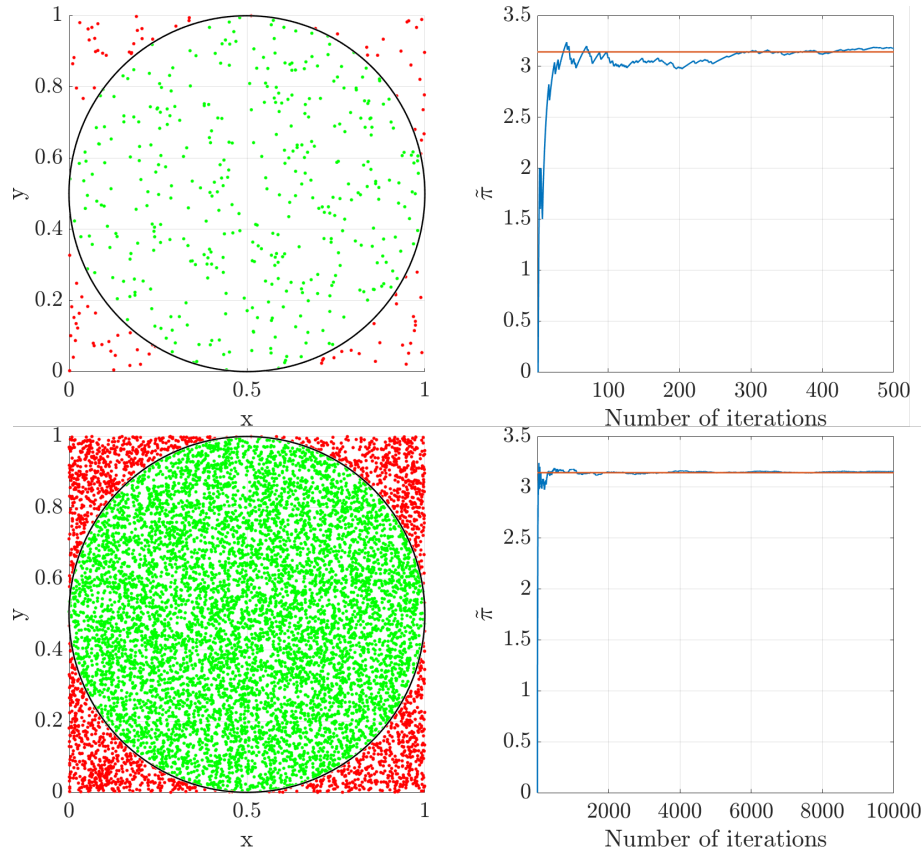
Figure 3.8: Monte-Carlo π approximation after 500 iterations and 10000 iterations.

The Monte-Carlo ray tracing method is the basic ray tracing that was used to throw a lot of random particles in random directions over the 3D scene and observe their evolution. This method has the advantage of being light and simple to develop and to consider all radiative transfer effects without approximation. However, as shown with the simple π approximation example, the convergence may take a long time and an important particles number to be thrown [158]. As a consequence, this method may lead to an important computational cost.

Path tracing   Instead of throwing randomly the rays, other methods were developed to optimize the path taken by the rays. In path tracing, the thrown rays may not have randomness in its process. This is the most common behavior used in most ray tracing algorithms. A ray can be drawn from a deterministic way even if some randomness can be used, particularly for taking into account advanced physical phenomena [64].

Radiosity methods

In order to get a more realistic global illumination algorithm, many developments have been made to solve the radiosity equation. This equation faces multiple challenges: computing the visibility between the surfaces, computing their form factor and finally solving the equation system. Even if the discrete version of the equation is a linear system, this system can become very large for refined meshes and therefore computationally intensive to solve. On the contrary of ray tracing algorithm, once the radiosity have been computed, the solution is global and does not depend on camera's view.

> **Note**
>
> In real-time 3D computer graphics, radiosity is used to compute the so-called lightmaps. Lightmaps are precomputed textures that aim at caching light data of a 3D scene. Once the radiosity of the scene have been computed, the brightness of the surfaces is stored on such texture. To render the scene, the texture is then mapped against the surfaces.

Progressive radiosity   To overcome this issue, Cohen *et al.* [41] proposed an iterative approach to the solution system. The advantage of this method is its ability to progressively solve the equation system and display the result at each iteration. The original algorithm takes into account diffuses surfaces on nonparticipating media with occlusion. Please note that improvements have been made on this algorithm over time to take into account transmission, absorption, specular reflection, participating media or bidirectional reflection [162, 164, 25, 40]. Thanks to the technological development of computer's graphical processing units and related software, this algorithm can be optimized to render more efficiently the radiative transfer exchanges inside 3D scenes. The parallelization task is however challeng-

ing due to the high dependence that exists between the mesh elements [42, 187, 7]. This algorithm has been retained in this thesis for its physical and view-independent approach. Therefore, more details on this algorithm and implementation will be given on the dedicated Chapter 5.

Improvements on the radiosity method   As stated previously, improvements for computing the radiosity have been done in the literature, in particular to increase the speed of the rendering or to inject more phenomena in the radiosity equation. Some possible improvements are described here.

As for example, general reflectance functions can be approximated by spherical harmonics [164]. This method consists in decomposing the radiance into a sum of spherical harmonics, linked to Legendre polynomials, and has been developed as well in the radiative transfer field [129].

Meshing is an important part of the radiosity process which will have an impact on both accuracy and speed computation. In fact, shadows can fall between mesh vertices and can be entirely missed for example. A common extension to the progressive radiosity method is to refine the mesh as the solution is computed [86, 184, 136] or in post process, which is common in finite element methods [156].

Hierarchical methods are a meshing strategy used to reduce the computational cost of the radiosity solution. A hierarchical subdivision method is based on the previous adaptive subdivision but this time, the radiosity solution is approximated at different scales [38, 40]. The same number of elements is kept but if a local patch needs to be refined due to an important gradient for example, a subdivision can be made on this part only of the solution and a hierarchy is performed which implies to keep only a hierarchical interaction (see Fig. 3.9). This method provides important improvements if the original
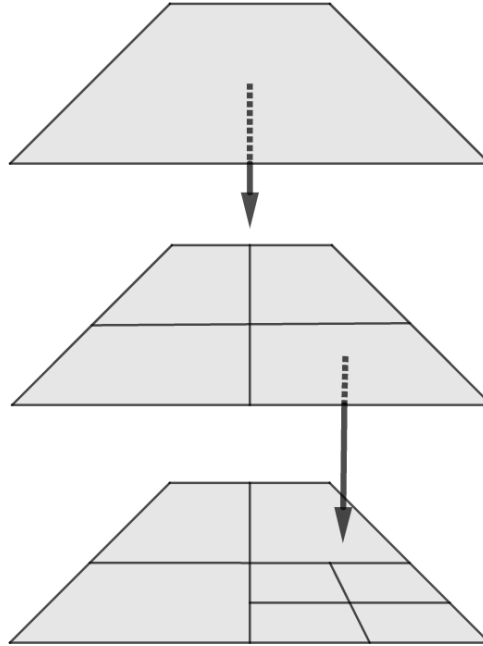
Figure 3.9: Hierarchical subdivision

mesh can be refined without any need of new tesselation (mainly for planes and square shapes). When dealing with concave and convex geometries, the improvements falls drastically: a curved surface already needs to be represented as small polygons and thus, the hierarchical radiosity will not bring a lot of improvements. Nevertheless, some techniques can be added to the radiosity method in the particular case of curved shapes [89, 25]. This algorithm has been used recently for the simulation of thermal radiative transfer inside houses and implemented on a central processing unit (CPU) [107].

> **Note**
>
> When parallelized, the adaptive subdivision becomes more complex; fortunately this can be overcome on current graphic hardwares with the use of quadtree textures [79].

For participating media, the classical radiosity method can be extended through voxels (*i.e.* pixels in 3D) to consider the radiative exchanges in 3D [138, 9, 161].

> **Note**
>
> To complete the previous note on quadtree textures to represent subdivided patches, the voxel representation in recent hardware is made possible by the use of octree [110]

Finally, please note that other methods also exists. For example meshless methods have been developed [112, 111] on which only a mathematical description of the domain is needed, without any need for meshes. In this case, a basis function is used to approximate the solution on the 3D objects at some points. Those points are then interpolated to get the final estimate.

Hybrid methods

To overcome the radiosity difficulty to take into account specular reflection, participating media and also dynamic scenes, hybrid methods that combine both ray tracing and radiosity have also been developed [157]. Two common methods called photon mapping and instant radiosity are presented first. Then more recent and advanced ones will be briefly presented.

Photon mapping The photon mapping approach introduced by Jensen and Chris-tensen [65] is a two-pass algorithm. First, the photons are traced from the light sources into the scene. Each time a photon intersects a surface, the intersection point and incoming direction are stored on a cache called photon map. Then, based on this photon map, the rendering can be performed using the radiosity equation. A

ray tracing method is used where the ray is divided into categories: direct illumination, specular reflection, transmission and indirect illumination. The direct illumination, specular reflection and transmission are ray-traced but the indirect illumination is computed using the previously cached photon map. For a given point, N nearest photons are gathered. The photon density is then computed by adding the flux of those photons and divided by the projected area of the sphere that contains those N photons [64]. Photon mapping is a simple and efficient method able to deal with complex geometries and multiple light phenomena such as caustics. However, as stated in [25], it suffers from various artifacts such as blurred shadows and requires a high amount of memory when a lot of light are present on the scene. In fact the photon map is a shortcut that allows the interpolation of photon density. However, if all surfaces are considered to emit such as in the radiation field, then the photon map may not be efficient.

Instant radiosity　　Instant radiosity is also a two-pass hybrid approach. The aim of the algorithm is to provide a real-time radiosity solution [99]. In this method, the indirect illumination is replaced by the direct illumination from new light source points (called virtual point light (VPL)). As for the photon mapping, a first set of photons are thrown on the scene and new point sources are placed where a trajectory intersects a surface. Then the direct illumination due to these point light sources is accumulated, resulting in an approximation of both direct and indirect illumination. This method provides actually fast results. However, the resulting image may have bias due to the VPL introduced into the scene. First, a too much intensity at the location of the VPL can occur. Then, if the VPL number is too small or not well sampled regarding the geometry, some interactions may be missed. A solution to this issue is given in [154].

Recent developments　　Based on the VPL, more recent developments have been made, in particular to improve the computation speed of the process and adapt it to modern hardware [188, 33, 106].

Also, efficient and nowadays often used methods for real-time global illumination algorithms are the *light propagation volume* method which is an extent of VPL where a hierarchical storing data is performed: every VPL stores which points in the world they light up on a 3D grid. Then the light is spread by using spherical harmonics for the directional reflectance function approximation. This method was developed by Crytek for their rendering engine CryEngine.

Finally, another method is the *voxel cone tracing*. Again the scene is represented by volume elements (voxels) and stored in an octree texture. Then, instead of tracing a ray represented by a line, the ray is represented by a cone. This cone can then propagates its energy through the octree across the different refinements scales [47].

Those advanced algorithms render a highly looking realistic scene in real-time but still resides in a view-dependent solution and approximation on the radiosity equation. The number of bounces or VPL in those algorithms will refine the solution but increase the computational time. Moreover, those

algorithms are also based on the fact that not all the surfaces on the scene are a source of light.

Softwares for radiative simulations

Softwares have been developed in the literature for computing radiative exchanges of 3D scenes. SOLENE notably, initially developed for architectural purposes, takes into account the radiative exchanges in urban scenes [81, 88]. It proposes thermal balance, building surface temperature, effect of vegetation on streets *etc.* This software has multiple modules to make different thermal interactions based on finite element models. The method used for solving the radiative exchanges is the progressive radiosity refinement method, for its rigorous physical approach [103]. Although the results are interesting and the models used are complete to simulate a micro-climate urban scene, this software is not available freely and suffers from a lack of online documentation.

Synthesis

The radiosity continuous equation have been discretized through a finite element method, inspired from the radiative transfer field. This equation has then been taken to the image synthesis field for rendering realistic 3D scenes. Then, a short overview of different techniques used in image synthesis have been shown. If the ray tracing based algorithms offer a way to render complex phenomena, those methods suffer from their view-point dependence, and the need for denoising the rendered image in post-process. Moreover, some of those methods may not strictly respect the physical equations. At the opposite, methods that aim at solving the radiosity equation are view-independent and respect the physical equations (relative to the spatial discretization). However, radiosity based methods are computationally intensive and are not able to take into account all physical phenomena easily. To conclude, many methods have been developed to simulate the light transport. Some algorithms will therefore perform better in specific situations: from a generic perspective, radiosity based algorithms will behave better in pure diffuse environment and ray tracing in highly specular scenes. In order to simulate the radiative exchange between surfaces for *in-situ* infrared thermography measurements, the progressive radiosity method has been chosen. It has been implemented on recent graphic hardware and slightly adapted with a GPGPU approach inspired from the image synthesis field and presented in Chapter 5.

Multiples solutions exist to improve the temperature estimation on *in-situ* infrared thermography. As shown in this chapter, the first way is to develop emissivity and temperature separation methods to get a more accurate estimate of the temperature, or at least a known uncertainty on this estimation. This can be achieved through the use of Bayesian methods. Secondly, the optimization of *in-situ* set-up due to the angle and view dependency of the measurement will also have an important impact on the final temperature estimation. As a consequence, a 3D radiative exchanges simulation tool can be used to optimize and visualize the measurement, before any setting up. This tool, can also be used

to test the previous temperature and emissivity separation methods. Finally, the estimation of the parameters other than emissivity and temperature can also help to improve the estimation accuracy. The next chapter aims at introducing the concepts of practical infrared thermography measurements and the impact of the environmental parameters on the final temperature estimation.

In-situ long-term thermal monitoring of structures: environmental measurements bias compensation

I N the previous chapters, the different equations that govern the thermal radiative transfer have been exposed. It has been shown that the simplified radiometric equation, used to derive the temperature from infrared measurements depends on many factors. To overcome the lack of knowledge about most of those parameters, one solution is to use other *in-situ* measurements, or at least find a way to estimate those secondaries variables. Practical instrumentation considerations for *in-situ* infrared thermal monitoring will be exposed in this chapter. In particular, the need for coupling multiple sensors with an infrared camera will be presented. Handling multi-sensors data over long-term thermal monitoring can lead to an important amount of information to store. Those data need to be well registered. In this case, this is done by using standardized file formats for the records that allow the writing of *metadata* (a description of the recorded data). Moreover, a specific tool has been developed during this thesis to visualize and process such amount of data.

Like any measuring tool, an infrared camera needs to be calibrated. In order to retrieve the temperature from the digital levels returned by the camera, the process of thermal calibration is shown. Also, due to the geometrical dependency between the camera and the observed surfaces, a spatial calibration is proposed. Such spatial calibration allows the retrieval of some geometrical parameters such as the angle and distance between the object and the camera.

Moreover, a study of the effects of the different parameters that influence the measurement are also exposed. Finally, it will be shown that it is still possible to estimate some parameters of the model when no coupling has been done. This is achieved by using open data available online.

4.1   Infrared camera calibration

This section introduces the thermal and spatial calibration processes, needed to make as trustful measurements as possible. The thermal calibration process is performed prior to the measurements in laboratory conditions. The spatial calibration can be done in different ways, exposed below.

4.1.1 Thermal calibration

Thermal calibration functions

The equations behind the measurements have been presented previously. Once the object radiation $L_{\Delta\lambda}^{\circ(i,j)}(T_{obj}^{(i,j)})$ has been estimated, it is still necessary to convert this radiation to temperature. The most popular calibration function used for radiation thermometers is the Sakuma-Hattori equation [128, 104, 175]. Introduced in 1982 [151], the Sakuma-Hattori equation provide the electromagnetic signal of thermal radiation from an object's temperature. This equation is then used during the thermal calibration process to relate the blackbody source temperature to the signal received at sensors. Multiple equations have been derived then from this main equation to fit different bandwidth and needs [153]. The Planckian form of the Sakuma-Hattori function is given in the following, where A, B and C are constant values and C_2 is the second radiation constant.

$$S(T) = \frac{C}{exp(\frac{C_2}{AT+B}) - 1} \tag{4.1}$$

where $S(T)$ is the measured quantity, in digital levels. An equivalent function is often found in the IR thermography field called the RBF-function.

$$S(T) = \frac{R}{exp(\frac{B}{T}) + F} \tag{4.2}$$

This equations are mainly used due to their close form to the integrated over spectral responsivity's Planck's law and the fact that it can be inverted.

Thermal calibration process

In order to calibrate thermally the camera, a black body calibration source is used. The infrared camera is placed in a climatic chamber where the temperature is controlled. Different measurements at different source temperature are made. Finally, the RBF function is fitted to the measurements through an optimization algorithm, deriving the R, B and F constants. An example of such calibration process is presented in the next part.

Thermal calibration example

A black body calibration source Mikron M310 ($\epsilon_{eff} = 1, \forall \lambda \in [8\mu m, 14\mu m]$) has been used for thermally calibrating the infrared camera Fig. 4.1. Experimental calibration at 10K sampling has been made for $T \in [293.15K, 343.15K]$. The quantity $\left(T_{\text{object}} - T_{\text{calibration}}\right)^2$ is minimized through the Levenberg-Marquardt algorithm.

The Fig. 4.2 shows the optimized parameters improvement on the conversion process from the measurements. The distance between the objective temperature and the measure one is reduced and

(a) Mikron M310
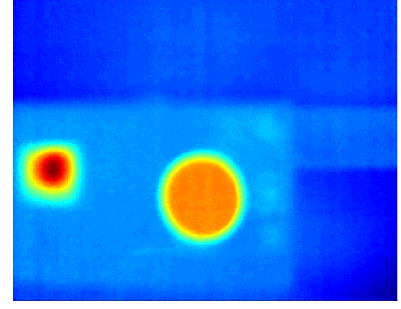
(b) Infrared view of the source

Figure 4.1: Black-body calibration source

the standard deviation is also reduced.
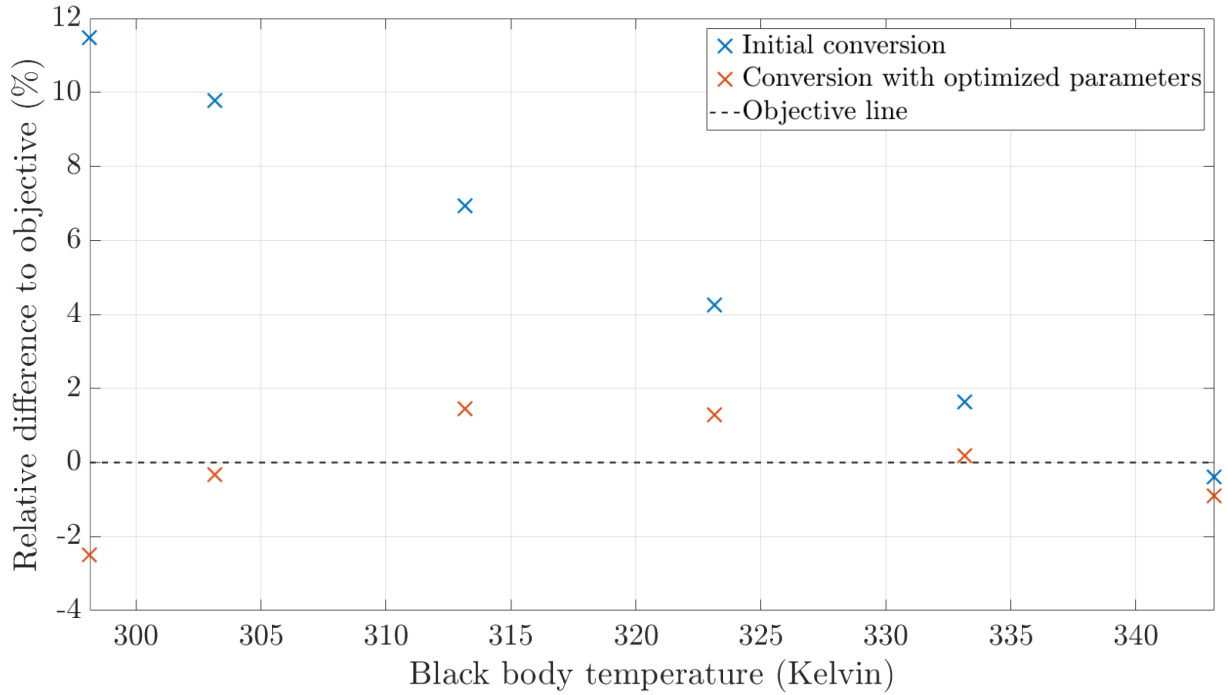
Figure 4.2: Result of thermal calibration

4.1.2 Spatial calibration

The necessity of knowing the geometrical information has been presented earlier. Some methodologies and algorithms to perform camera resectioning (*i.e.* calibrate spatially the camera) are proposed. Many algorithms and methods have been developed in the computer vision field for many different applications such as machine vision, mobile robot navigation, biomedical, *etc.* Some differences from the traditional algorithms that we can find in the literature can be noted. In the case of infrared

structure monitoring, the input images comes from other spectral ranges than the visible spectrum which leads to a different pixel range dynamics. Moreover, in some cases, it can be difficult to put a geometric reference on a test site for safety purposes. However, we usually use a 3D model of the structure we are monitoring that is assumed to be available.

Camera resectioning

Introduction The objective of camera resectioning is to find the new coordinates of the projected points into the image from the real scene (see Fig. 4.3). Such calibration gives the position information from an image to its corresponding points in the real scene. Different calibration methods exists, mainly depending of what we know about the observed scene.
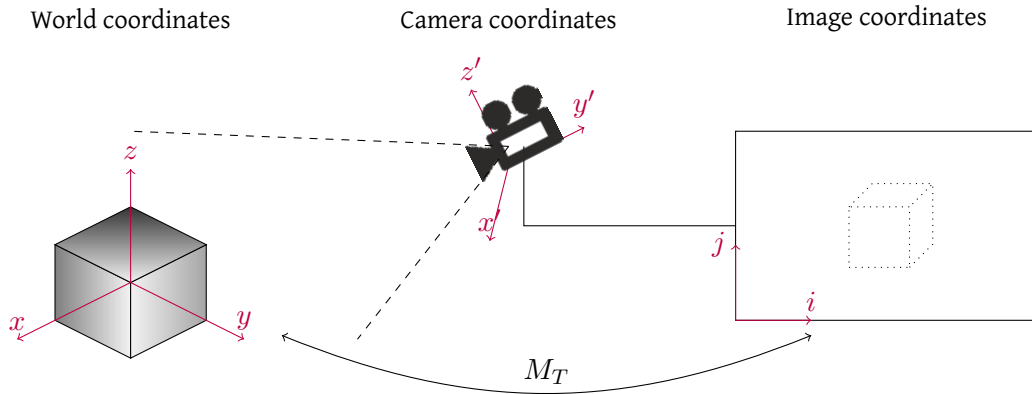
Figure 4.3: Principle of camera resectioning.

A camera resectioning model can be defined through two types of parameters, represented in Fig. 4.4:

- **Extrinsic parameters**, that transform the 3D coordinates of the scene into the 3D coordinates of the camera. These parameters mainly depend on the scene and the camera disposition relative to it.
- **Intrinsic parameters**, that transform the 3D coordinates of the camera into pixels coordinates. The intrisic parameters depend on the camera properties due to the lens used for example.

The simplest model of a perspective matrix is given below. We will review in this section how to extend this model and how to estimate the different parameters.

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} \tag{4.3}$$

74
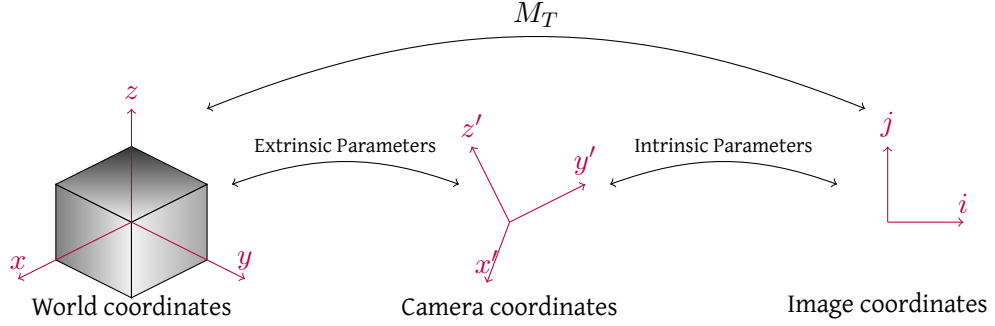
Figure 4.4: Extrinsics and intrinsics parameters over the camera resectioning.

Extrinsic parameters The first step for converting the real world coordinates into pixel coordinates is to translate the information of the camera position.

The camera can be translated and/or rotated from the real world. Therefore, a combination of rotation and translation is enough to define the camera position from the real world:

$$\overrightarrow{p}_c = R_{w \to c} \overrightarrow{p}_w + \overrightarrow{T}_{w \to c} \tag{4.4}$$

where $\overrightarrow{p}_c$, $\overrightarrow{p}_w$ respectively define the point from the camera frame and a point from the scene frame. $R_{w \to c}$ is the rotation matrix from the world to the camera and $T_{w \to c}$ is the translation one.

This transformation creates 5 degrees of freedom (DOF), due to the need of setting 3 points in $\mathbb{R}^3$.

- First point p_1: 3 degrees of freedom (x_1, y_1, z_1)
- Second point p_2: 1 degree of freedom $d = \|p_1, p_2\|$
- Third point p_3: 1 degree of freedom θ, angle around the line created by p_1, p_2.

In order to get from this new camera 3D frame to the pixel frame, we now need to get the intrinsic parameters.
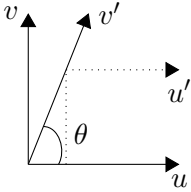
Intrinsic parameters With Eq. (4.3), the simple perspective transformation can be written as:

$$\begin{aligned} u &= f\frac{x}{z} + u_0 \\ v &= f\frac{y}{z} + v_0 \end{aligned} \tag{4.5}$$

where u_0 and v_0 are the camera origin coordinates in the pixel frame. However, by consediring the fact that pixels are in some arbitrary spatial units, and not always square:

$$\begin{aligned} u &= \alpha\frac{x}{z} + u_0 \\ v &= \beta\frac{y}{z} + v_0 \end{aligned} \tag{4.6}$$

Moreover, we can also consider a coefficient of non-orthogonality between x-axis and y-axis of our frame, called "skew" and represented by Fig. 4.5:

$$u = \alpha\frac{x}{z} - \alpha\cot(\theta)\frac{y}{z} + u_0$$
$$v = \frac{\beta}{\sin(\theta)}\frac{y}{z} + v_0 \qquad (4.7)$$

$$v'\sin(\theta) = v$$
$$u' = u - \cos(\theta)v'$$
$$= u - \cot(\theta)v$$

Figure 4.5: Frame skew

Such equation leads us to the following matrix in homogeneous coordinates:

$$\underbrace{\begin{pmatrix} zu \\ zv \\ z \end{pmatrix}}_{\overrightarrow{p'}} = \underbrace{\begin{pmatrix} \alpha & -\alpha\cot(\theta) & u_0 & 0 \\ 0 & \frac{\beta}{\sin(\theta)} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{=K} \underbrace{\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}}_{\overrightarrow{p}}$$

In order to have a more readable and understable matrix, let s be the skew, (c_x, c_y) the camera center and a the aspect ratio:

$$K = \begin{pmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{pmatrix} \qquad (4.8)$$

Parameters combination By combining Equation 4.4 and Equation 4.8:

$$\overrightarrow{p'} = K(R_{w\to c}\overrightarrow{T}_{w\to c})\overrightarrow{p}$$
$$= M_T\overrightarrow{p} \qquad (4.9)$$

With I_3 the identity matrix, M_T is:

$$M_T = \begin{pmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} R_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix}\begin{pmatrix} I_3 & T_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix} \qquad (4.10)$$

The equation 4.10 is called the pin-hole model. Therefore, a camera matrix is described by several parameters:

Extrinsics	Intrinsics
- Translation T from the origin of world coordinates - Rotation R from the world coordinates	- Focal length f - Principle points (c_x, c_y) - Aspect ratio a - Skew s

Finally, we have $11 = 5(\text{Projection}) + 3(\text{Rotation}) + 3(\text{Translation})$ degrees of freedom.

Therefore, we can estimate the transformation matrix if we have a prior knowledge of our observed scene. Which means having at least 6 points for which we know the position in the scene frame and in the image.

Camera resectioning algorithms

The methods can be divided into two main categories, traditional methods and self-calibration methods [145]. We use in the next part the non-linear based method that can take into account the distorsion of the camera. We assume that some reference are known in the image. At least, the Computer-aided design (CAD) is known, which is the case for most recent applications of civil engineering.

Classic camera calibration   In traditional methods, a reference on the scene or a set of correspondances points between the scene and the image are known. Then, an optimization algorithm is used to find the camera model parameters. Different methods exists such as direct linear transformation, non-linear methods, two-steps method (Tsai's algorithm), Zhang's method, *etc.*.

Camera auto-calibration   When there is no reference on the scene or that nothing on the scene is known, self-calibration techniques determines the internal camera parameters directly. Such calibration techniques can use multiple views or when the camera is moving perform an active camera self-calibration.

Direct Linear Transformation   The direct linear transformation (DLT) was introduced in 1974 by I.E Sutherland [174] for digitizing 2D drawings to 3D model with a pen. This algorithm solves a set of variables links to a set of similarity equations. Based on this algorithm, [85] have provided a "Gold standard" algorithm for calibration We have seen in the previous section that at least 6 points were needed to solve the equation system. If we have at least such points, we can create a homogeneous set of equations that defines a least squares problem when over constrained.

With $M_T = (m_{ij})_{\substack{1 \leq i \leq 2 \\ 1 \leq j \leq 3}}$ and by writting the system for n points, we have:

$$A_{2n\text{x}12} \begin{pmatrix} m_{00} \\ m_{01} \\ m_{02} \\ ... \\ m_{22} \\ m_{23} \end{pmatrix} = 0_{2n} \tag{4.11}$$

Where

$$A = \begin{pmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_1 X_n & -u_1 Y_n & -u_1 Z_n & -u_1 \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_1 X_n & -v_1 Y_n & -v_1 Z_n & -v_1 \end{pmatrix} \tag{4.12}$$

Therefore, we can minimize $\|Am\|$ to get a least squares problem when over constrained (6 or more points). Since the scale is arbitrary, we can also add the constraint $\|m\| = 1$. This can finally be formulated as a homogeneous least square problem:

$$\min_{\|x\|=1} \|Ax\| \tag{4.13}$$

(Get the eigenvector of $A^T A$ with the smallest associated eigenvalue).

Singular Value Decomposition

Theorem 1. *Let M be a $m \times n$ matrix for which coefficients belongs to the field K where $K = \mathbb{R}$ or $K = \mathbb{C}$. M can be decomposed such as:*

$$M = U\Sigma V^* \tag{4.14}$$

where U is a $m \times m$ unitary matrix on K, Σ is a $m \times n$ matrix for which the diagonal coefficients are reals positives or zero and all others are zeros. V^* is the conjugate transpose to V, a $n \times n$ unitary matrix on K. This decomposition is called the singular value decomposition of M.

The first way to get the parameters is to solve 4.13 by eigenvalue decomposition. This can be achieved by a singular value decomposition of A.

Proof. Let $A = UDV^T$ be the singular decomposition of A (with real coefficients) such as D is sorted by decreasing values.

$$\|Ax\| = \|UDV^T x\| = \|DV^T x\| \tag{4.15}$$

Let $y = V^T x$, therefore we want to minimize $\|Dy\|$ subject to $\|x\| = \|V^T x\| = \|y\| = 1$. However, since D is diagonal with decreasing values, the minimum of $\|Dy\|$ is found when

$$y = (0, 0, 0, ..., 0, 1)^T \tag{4.16}$$

Therefore, $x = Vy$ is the last column in V.

Moreover,

$$A^T A = (UDV^T)^T (UDV^T) = V D^T U^T U D V^T = V D^T D V^T \tag{4.17}$$

Since $D^T D$ is diagonal, this means that the singular value decomposition corresponds to the eigenvalue decomposition. $D^T D$ contains the square eigenvalues of A and V contains the eigenvectors. $\qquad\square$

Algorithm 3: Pseudo-code of the SVD computation

Objective: Given $n \geq 6$ image point correspondences $x_i \leftrightarrow X_i$, compute the singular value decomposition of A that will minimize the homogeneous least square problem:

$$\min_{\|x\|=1} \|Ax\|$$

Initialization *Generate matrix A with x_i and X_i inputs*

Compute SVD

     Perform eigenvalues decomposition $A^T A = V D V^T$
     Find the minimum value of D diagonal corresponding to the smallest eigenvalue
     return the corresponding eigenvector from V

Output: *Estimate of the calibration matrix M_T*

Nonlinear methods   Those methods are preferred to the previous one. The idea is to define an error function between the two sets of points (projected 3D points and image positions). Then minimize E with nonlinear optimization methods such as Levenberg-Marquardt. This concept has been introduced by [85] with the direct linear transformation (DLT) algorithm. The objective is to estimate the transformation matrix M by minimizing an error function f:

$$\min_{M_T} f(x_i, M_T X_i) \tag{4.18}$$

Prior to this minimization, a normalization is recommended since it provides an invariant behavior regarding the scale and coordinate origins of the input points. Moreover, it also improves the accuracy of the results (see section 4.4.4 of [85]). Different normalization are possible. We will present here the isotropic scaling that we apply to the 3D and 2D dataset of points:

- The points are translated so that their centroid will be at the origin.
- The points are scaled so that in average, they will be around the unit ball of there space in euclidian norm.

Moreover, multiple error functions can be chosen even if they are obviously based on a geometric error function.

1. **Error in one image** The simplest way is to minimize the geometric distance of the projeted points with the objectives points:

$$\sum_i d(x_i, M_T X_i)^2 \tag{4.19}$$

2. **Error in both images - symmetric transfer error** However, in a more realistic case where image measurement errors occur in both the images, we can minimize the error for the forward (M_T) and backward (M_T^{-1}) transformations:

$$\sum_i d(x_i, M_T X_i)^2 + d(X_i, M_T^{-1} x_i)^2 \tag{4.20}$$

3. **Error in both images - reprojection error** In this case, we want to get a perfect match of the forward and backward projected points. To do so, we determine new correspondences of our input points $x_i \leftrightarrow \tilde{x}_i, X_i \leftrightarrow \tilde{X}_i$ so that M will perfectly match the two points and minimize the distance between the correspondances.

$$\sum_i d(x_i, \tilde{x}_i)^2 + d(X_i, \tilde{X}_i)^2, \quad \text{with } \tilde{x}_i = M_T \tilde{X}_i \tag{4.21}$$

The illustration in Fig. 4.6 inspired from [85] shows the differences between the symmetric transfer error and the reprojection error.
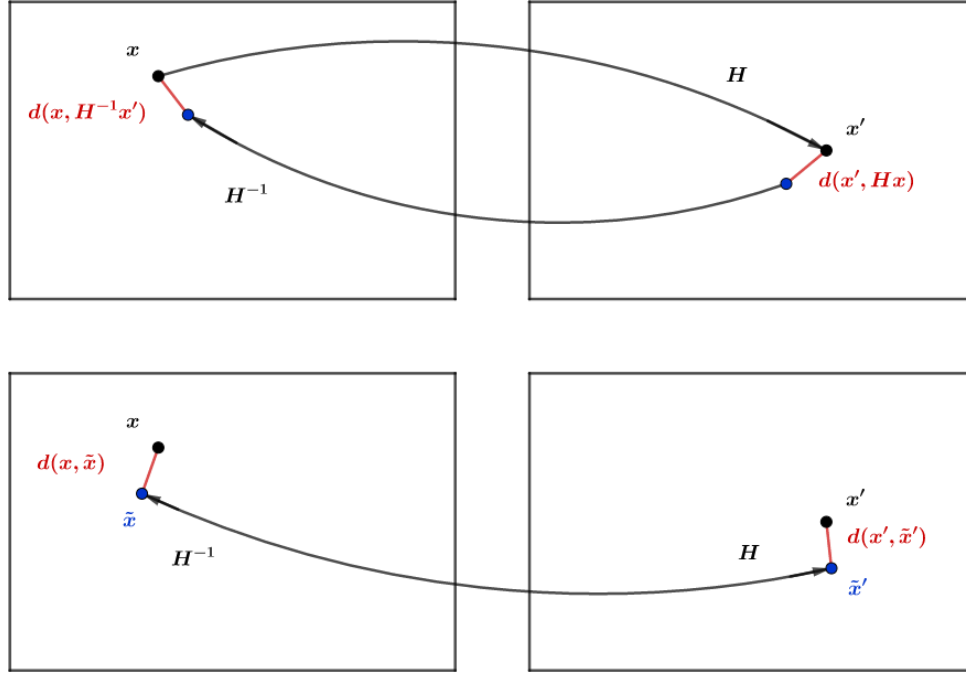
Figure 4.6: A comparison between the symmetric and reprojection error, as proposed in [85]

4. **Sampson error** Another cost function that can be used is called the Sampson error [85], based on the approximation used in [152] for conic fitting.

5. **"Gold Standard" Algorithm**: Algorithm 4

Algorithm 4: DLT based algorithm for calibration matrix estimation

Objective: Given $n \geq 6$ image point correspondences $x_i \leftrightarrow X_i$, determine the "Maximum Likelihood Estimation" of M

Initialization *Compute a linear solution to get an initial estimate*

    Normalization: Compute the normalization matrices U and V: $\tilde{X}_i = U X_i$, $\tilde{x}_i = V x_i$. We call $\tilde{M}_T$ the matrix such as: $\tilde{x}_i = \tilde{M}_T \tilde{X}_i$;

    Singular Value Decomposition: By stacking the equations, we get a $2n \times 12$ matrix A on which we solve $A\tilde{m} = 0$ by SVD where $\tilde{m}$ is the vector containing $\tilde{M}_T$ entries.

Levenberg-Marquadt

    By using the linear estimate as a starting-point, we minimize the geometric error:
    $\sum_{1 \leq i \leq N} d(\tilde{x}_i, \tilde{M}_T \tilde{X}_i)$

Denormalization

    The matrix is finally obtained by: $M_T = V^{-1} \tilde{M}_T U$

---

6. **Improving the estimation** It has been shown that at least 6 correspondences were needed to solve the system, meaning that the only source of error is in the measurement's point position. Most of the time this error is said to follow a Gaussian distribution. However, in practical, this assumption may not be verified since mismatch points will be outliers for the Gaussian distribu-

tion. Therefore, if we have $n > 6$ correspondences, a more robust estimation can be made by performing a Random sample consensus (RANSAC) [68] algorithm. A pseudo-code of a generalized RANSAC algorithm for this purpose is given on Algorithm 5.

Algorithm 5: Pseudo-code for RANSAC algorithm where the error function and the model are defined appart.

Function RANSAC(*data, fcnModel, fcnError, nData, nIteration, thres, nMinData*)

data	set of observations $\binom{Observations A}{Observations B}$
fcnModel	model to be adjusted to the data, returns the parameters p
	of the model ($p = \mathrm{fcnModel}(inputPoints)$)
fcnError	estimate error between a point and the parameters: $e = \mathrm{fcnError}(point, p)$
nData	number of data used to adjust the model
nIteration	number of maximum iterations
thres	a threshold value to adjust the data to the model
nMinData	number of minimal data necessary to consider this model

Initialization *iIteration = 0, finalError = ∞, other allocations here*

while *iIteration < **nIteration***

  rPoints = select **nData** random points from **data**;

  modelParameters = **fcnModel**(rPoints);   `/* Get those parameters */`

  setOfPoints = rPoints;   `/* We initialize the set of points */`

  foreach *iPoint ∈ **data** and iPoint ∉ rPoints* `/* For all other points        */`

    if ***fcnError**(iPoint, modelParameters) < **thres***

      add iPoint to setOfPoints;

  `/* If the number of elements is satisfying               */`

  if *numberOfElements(setOfPoints) > **nMinData***

    e = **fcnError**(setOfPoints, modelParameters); `/* Compute the error with all`
    `the points */`

    if *e < finalError* `/* Best model yet                      */`

      `/* Save the data                            */`

      finalModelParameters = modelParameters;

      finalSetOfPoints = setOfPoints;

      finalError = e;

  iIteration++;

Result: finalModelParameters, finalSetOfPoints, finalError

Summary of main resectioning algorithms

Type of view	Algorithm	Input Data	Output Data
One view	DLT	At least $n \geq 6$ correspondences	Transformation matrix M
	DLT + RANSAC	A bunch of correspondences	Transformation matrix
	Gold Standard	At least $n \geq 6$ correspondences	Model parameters
	Gold Standard + RANSAC	A bunch of correspondances	Model parameters
Two views	Zhang's method	Multiple images views with patterns and calibrated cameras	Essential matrix and parameters
	Zhang's method	Multiple images views with patterns and uncalibrated cameras	Fundamental matrix and parameters

---

Note

With the development of technologies to realize the 3D design of civil engineering structures, we can use those virtual object to help the camera resectioning. Some articles can be found in the literature for matching textured 3D models and 2D images ([29], [28]). [60] have proposed an algorithm for performing a calibration based on the CAD of the observed scene. This approach seems interesting but unfortunately has not been tested in this thesis.

Spatial sampling

Another topic that is also challenging is the extraction of profiles from infrared images. In fact, it can be interesting to extract thermal profiles from infrared images to compare them to theoretical models for example. In particular, the error made on the profile compared to the theoretical thermal one can be complicated to compute.

Monitoring a civil engineering structure requires a large angle of view which forces users to use a mast to place the camera. This situation results in images with varying spatial resolution and important angle between the object and the camera (*e.g.* Fig. 2.14). The underlying consequences are non-square pixels in the real-world and varying angles values along the same plane which has an impact on the emissivity values to be considered. One way to overcome this is to perform spatial calibration on images

to be able to recover the region of interest (ROI). However, such operation induces a new sampling obtained by an interpolation and illustrated in Fig. 4.7. The reference image in this figure shows the non-square and non-constant spatial sampling in the image.
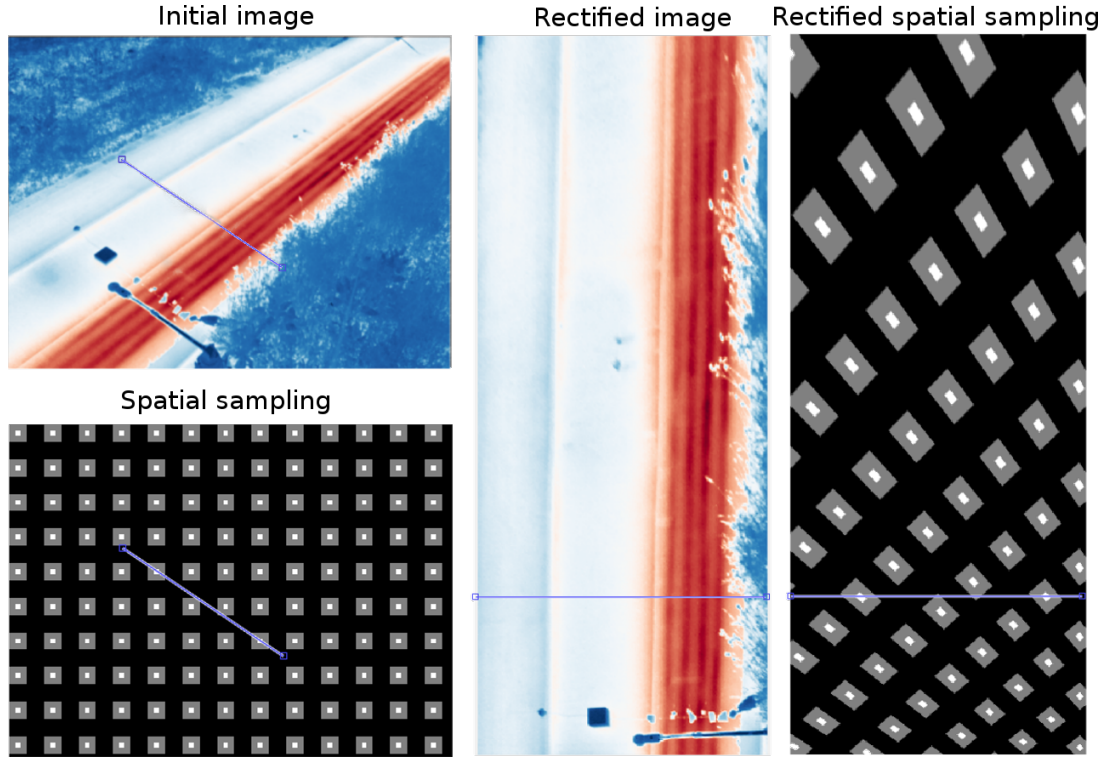
Figure 4.7: Result of interpolation after calibration and ROI recovering

When re-sampled, the extracted profile will be dilated and smoothed due to the interpolation, as we can see on Fig. 4.8a. Similarly, Fig. 4.8b shows that the dilation coefficient is not constant within the image and therefore, the error induced by the interpolation is varying depending on the pixels position. The common errors for such interpolations are a blurring of the edges and jagged artifacts.

Therefore, the profile extraction will induce errors from which an upper bound can be derived from Taylor's series in the case of polynomial interpolation. Of course, those errors are only from a mathematical point of view. Physically, it is even more complicated. In fact, considering the fact that one pixel in the image represents a varying surface in reality make the process of sampling more complex. A square pixel will potentially represent a non-square surface and all the pixels within the image may not represent the same surface in reality. Furthermore it is likely to have the mixed pixel effect, on which the pixel value does not represent a single surface material properties but multiple ones [96] (see also Fig. 2.15). This is actually a challenge often addressed in Remote Sensing [30].

Recent infrared thermography applications make use of unmanned aerial vehicles (UAV) to inspect large-scale structures [133]. The use of UAVs in the field of civil engineering has been investigated years
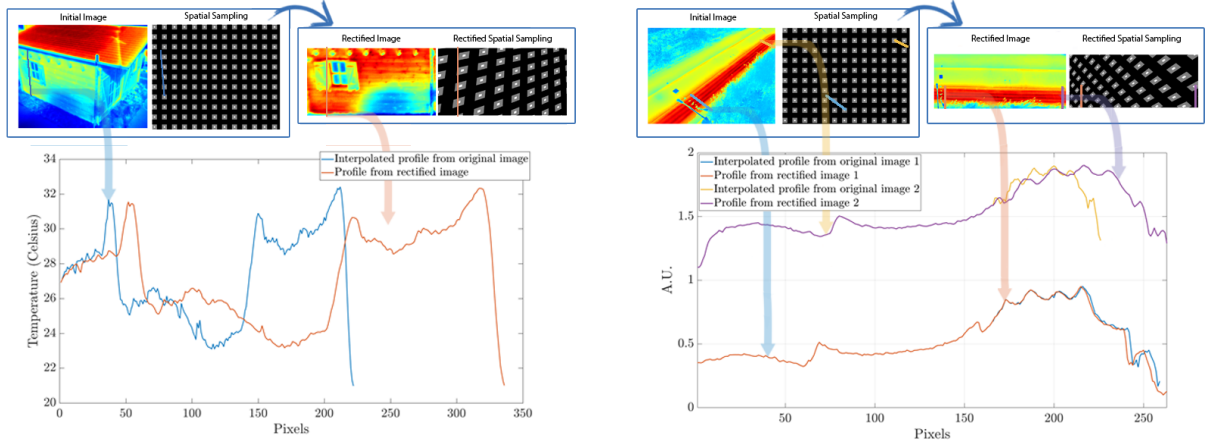
(a) The re-sampled profile is dilated and smoothed due to the interpolation

(b) The dilation coefficient depends on pixels position

Figure 4.8: Spatial re-sampling influence

ago [56] but thanks to the development of drones, UAVs offer new possibilities. Such technique enables an advanced inspection of the structure and can therefore overcome the former sampling issues. However, those applications are facing other technical difficulties such as motion blur, image stabilization, georeferencing, *etc.* Furthermore, UAVs may not be suitable for the live monitoring of running systems for security purposes such as bridges with operating cars and most of the employed methods are actives. Nevertheless, IR thermography combined with UAVs is an emerging promising method for many different applications [54].

4.2 Multi-sensor data exploitation

In order to perform a complete thermal monitoring of an infrastructure, one needs to set up a complex instrumentation based on various sensors. The aim of such instrumentation is to overcome the lack of knowledge about environmental and meteorological parameters. A reminder of the simplified radiometric equation to solve is given in Eq. (4.22). Thus, a multi-sensor instrumentation helps for estimating the different involved parameters.

$$
\begin{aligned}
L_{\Delta\lambda,\text{total}}^{(i,j)}(T) =& \tau_{opt}\left[\tau_{atm}\left(\epsilon_{\Delta\lambda,\text{obj}}^{(i,j)}L_{\Delta\lambda,\text{obj}}^{\circ(i,j)}(T_{obj}^{(i,j)}) + (1-\epsilon_{\Delta\lambda,\text{obj}}^{(i,j)})(L_{\Delta\lambda,env}^{\circ} + L_{\Delta\lambda,sun}^{\circ})\right)\right.\\
&\left. + (1-\tau_{atm})L_{\Delta\lambda,atm}^{\circ}\right] + (1-\tau_{opt})L_{\Delta\lambda,opt}^{\circ}
\end{aligned}
\tag{4.22}
$$

Therefore, a need for connected sensors and a flexible architecture with different types of measurements (punctual, *in-situ* full field thermal images, distributed measurements, etc.) appeared. Even if some tools have already been studied and developed for gathering and recording all the data from such instrumentation [51, 50], a software for exploiting and helping to process all this amount of data was

lacking. In such context, a thermal infrared images post-processing tool has been developed under MATLAB® in order to visualize infrared images and process data from various sensors. The developed software had to be user-friendly to help for importing data, to navigate between different time stamps of the experiment and also export data to other formats.

4.2.1   Input data and standard formats

The need for interoperability between previously developed tools from the community is primary. In this sense, the developed software makes use of OGC standards. The *Open Geospatial Consortium* (OGC) is an international not for profit organization committed to developing open standards. The advantage of using standards is a certain guarantee of the interoperability with other developed tools. Therefore, one of the main feature of the developed software is its ability to read and convert infrared images from various files formats; from proprietary ones to OGC ones such as HDF. *Hierarchical Data Format* (HDF) is a digital container format used to structure and handle large amount of data. It is supported by numerous programming languages and offers possibilities to compress data. The data is structured around groups that handles different type of data and *datasets* that represents a single type of data. Such type of format is necessary for *in-situ* large scale monitoring. In fact, it allows recording and gathering multiple type of data in one single file (see Fig. 4.9).

- GPS: location, time (up to synchronization)
- Local Weather: humidity, pressure, temperature, hail data, rain data, wind data, …
- Regional data: METAR network data or alternative accessible open source
- Shortwave radiation data
- Longwave radiation data
- Full field IR camera: 16 bits data, 640×480
- Metadata: Full description and remarks for each recording data
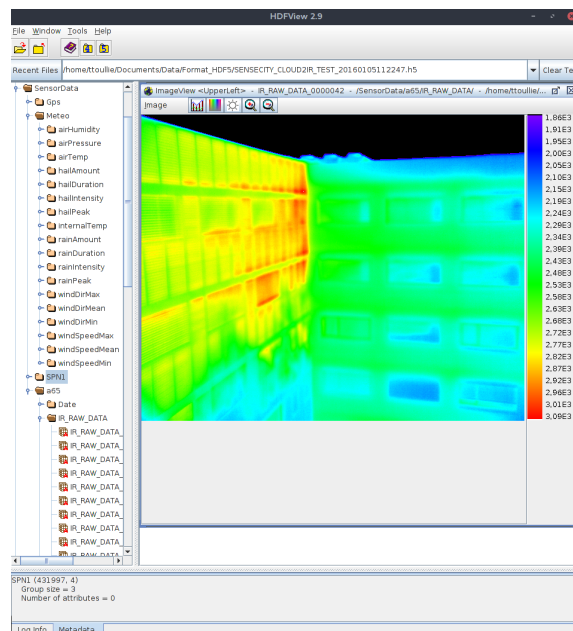
(a) Description of one HDF5 file record   (b) View of the recorded file through HDFView software

Figure 4.9: HDF file recording format: different data type from various sensors can be registered and then read back fastly due to the HDF internal architecture.

The software can therefore read and import different data through multiple file formats (.hdf; proprietary IR: .seq, .fff; binaries IR: .irl; common ones: .csv, .txt etc.) and make the conversion between any of those formats. The importation and formatting is facilitated by a custom importation module (see Fig. 4.10).
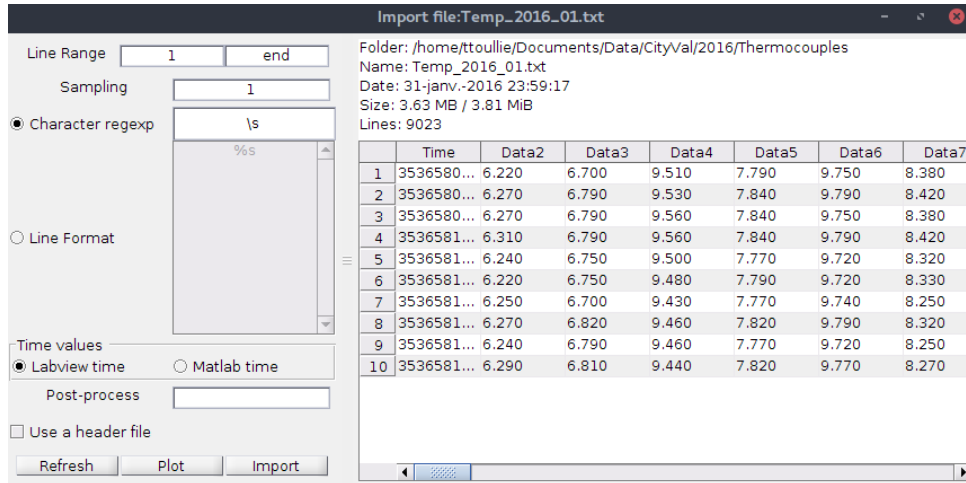
Figure 4.10: Any text file format can be imported and formated easily

4.2.2 Data processing

Conversion process

The tool supports different data processing algorithm. The conversion model from digital levels to temperature can be customized and the parameters derived from the thermal calibration can be sets. The defaults values are the one stored into the metadata of the input file. Even though the conversion from digital levels to temperature is not a difficult computation to perform for MATLAB®, it may be too long for fast visualization purposes. In fact, when displaying the images and browsing from them, any latency can make the user's experience complicated. As a consequence, the computation speed has been improved by the parallelization of the conversion model. The digital levels to temperature conversion is then quickly done and can be customized depending on user's hardware. Three choices are available, from the slowest to the quickest: CPU only, GPU through `gpuArray`'s built-in MATLAB® functions and CUDA using custom GPU kernel. *Compute Unified Device Architecture* (CUDA) is a *General-Purpose Computing on Graphics Processing Units* (GPGPU) technology. Due to the constant improvements of graphic hardware for 3D graphics compute-intensive real-time applications, the GPUs became really efficient for performing highly parallel tasks on large blocks of data. Therefore, this design led to increase the number of applications and therefore derive the original purpose of 3D graphics to scientific general computations. CUDA was therefore developed by NVidia for their hardware to achieve such goals and interface directly with the GPU.

To develop such CUDA kernel on MATLAB®, a C program linked with the CUDA kernel is developed and interfaced with MATLAB® through a MEX compilation [78]. In order to be able to compare the three implementations, the processing time is calculated from the beginning of the conversion call process to the final converted image (it means that it includes the transfer communication CPU ↔ GPU). Moreover, this amount has been averaged over 60 different images conversions. The standard `gpuArray` represents an improvement of a factor 10 compared to the CPU conversion whereas the custom CUDA kernel represents an improvement of a factor 40. The difference between the `gpuArray` and the CUDA kernel can be explained due to the fact that `gpuArray` offers an easy way for developers to parallelize their algorithm. As a result, it also need to perform multiple checking to be able to handle any user code. Please note that with the built-in functions, developers only need to declare their variables as `gpuArray`, the code does not change. At the opposite, the CUDA kernel development needs some knowledge on C, CUDA and compilation. Fig. 4.11 shows the comparison of those implementations.

(a) CUDA kernels in MATLAB® (b) Time performances for the different methods
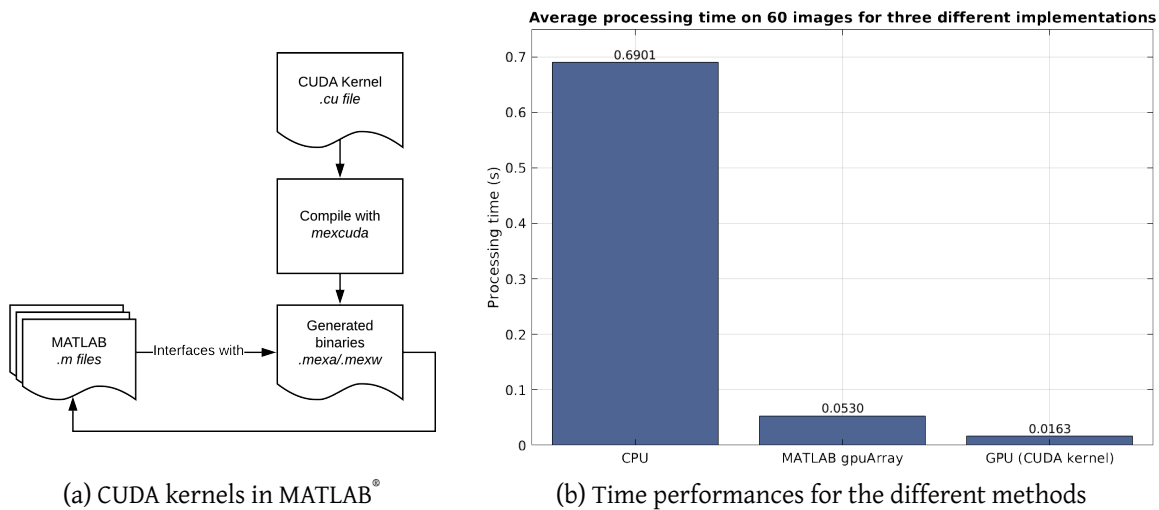
Figure 4.11: Parallelization of the temperature conversion process onto the GPU. Comparison of the different implementations

Graphical User's Interface (GUI)

To be able to visualize the data efficiently, an interface has been developed to facilitate the processing and be able to detect easily errors or aberration. Fig. 4.12 shows the user's interface.
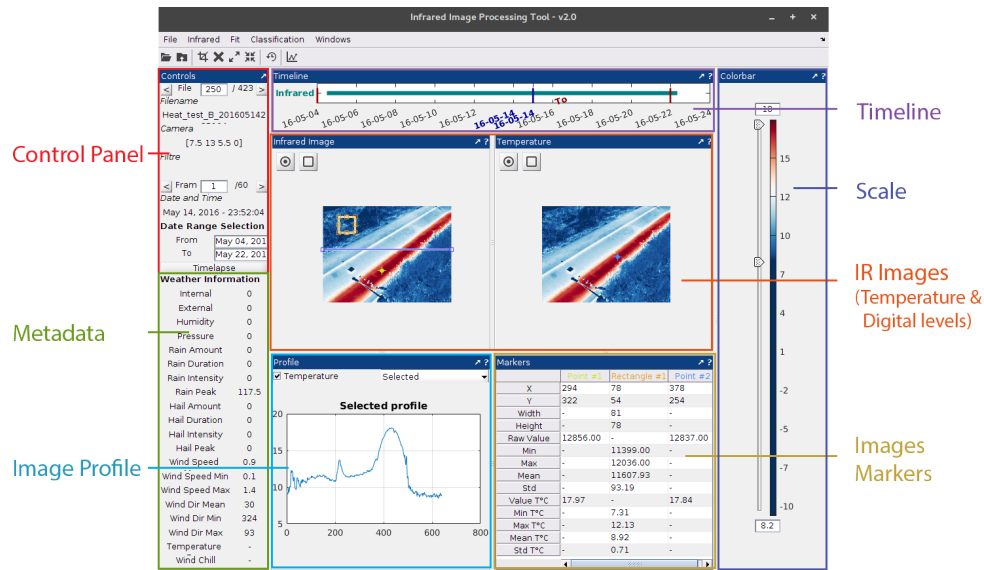
Figure 4.12: GUI of the tool

Batch processing

Even though a GUI is user friendly, it is not efficient to process a large amount of data at once. Instead, in order to apply some specific processes on a large amount of data, a batch process module have been created. This module relies on *YAML Ain't Markup Language* (YAML) language [178]. It is a markup language that has the advantage of being readable for humans, representing data through lists. Furthermore, its non-hierarchical structure but relational based one avoid redundancy: identical data can be referenced multiple times in the same file without the need to repeat the data. The process works as follow, as shown in the schematic in Fig. 4.13:

1. The user describe the input data,
   - Folders / files paths,
   - Begin date and end date,
   - Optionally how to extract the date from the file.
2. The output that will be stored into a global structure with its data type.
3. The actions to be executed,
   - Name of the action.
4. Definition of the action,
   - Name of the action (as in 3.),
   - When the action is executed (Begin / For every file or given frame rate / End),
   - Function to call for executing the action.

An example of such YAML configuration file is given in Appendix A.
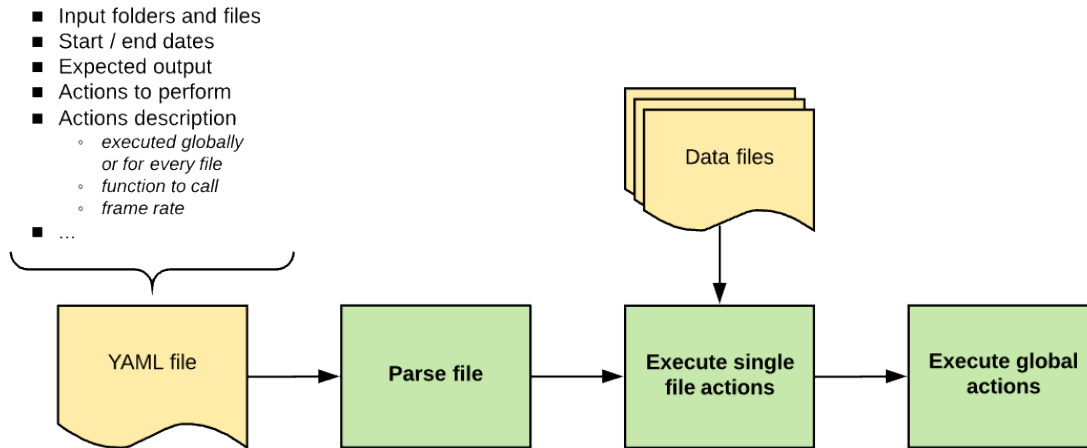
Figure 4.13: YAML batch processing flowchart

4.3   Tests sites and use cases

In order to illustrate the bias through practical use-cases, long-term *in-situ* measurements made on two mock-up are used: an instrumented wood house and an instrumented road section that are presented in the following.

4.3.1   Instrumented road section
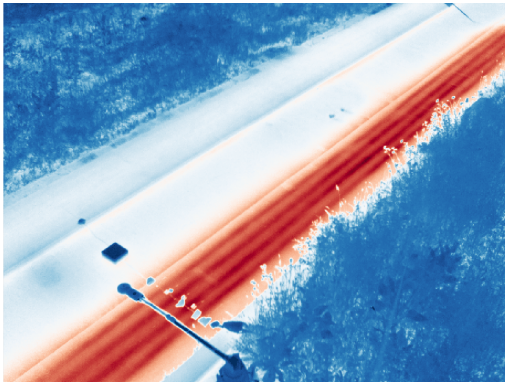
(a) Image of the road in the infrared spectrum

(b) Image of the road in the visible spectrum

Figure 4.14: Road section instrumentation: IR image (a) and test site overview (b)

A road section that can be internally heated has been instrumented with different sensors. The infrared thermal monitoring is done with a FLIR SC655 thermal infrared camera (640×480 LWIR FPA, pitch $17\mu m$, and a $13.1mm$ optical focal length). A weather station (Vaisala WXT 520), a net radiometer (CNR4) and a pyranometer (SPLite2) provides complementary measurements with meteorological data (rain, wind, temperature, etc.) and radiative heat fluxes. All those data are synchronized in time thanks

to our multi-sensor data acquisition system Cloud2IR. Due to the particular angle of view, distance and planar properties of the scene, those data are interesting to process to observe spatial sampling issues, mixed pixels and transmissions effects. Moreover, the presence of thermocouples on the road enables us some comparison with infrared measurements.

4.3.2 Instrumented wood house

A wood house has been built and instrumented for multiple years in Paris under the SenseCity project of IFSTTAR. The house is monitored with a FLIR A65 thermal infrared camera (640×512 LWIR FPA, pitch $17\mu m$, $13mm$ optical focal length), a weather station (Vaisala WXT 520) and a sunshine pyranometer (SPN1) for the total and diffuse solar radiation. Cloud2IR is also used here for recording synchronized data.
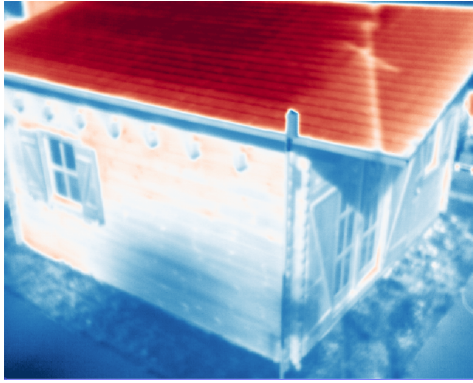
(a) Image of the house in the infrared spectrum   (b) Image of the house in the visible spectrum

Figure 4.15: Wood house instrumentation: IR image (a) and test site overview (b)

4.4 Parameters sensitivity

4.4.1 Sensitivity to emissivity

Different analysis of emissivity sensitivity have been previously done in the literature [34, 35, 141, 102, 71] and [128] especially. Those analyses are extended to the case of the *in-situ* model presented previously and focus on the emissivity, environment temperature and atmospheric transmission effects. In particular, the use of multi-sensors data for practical use cases to improve the temperature

estimation is discussed. By considering that the object's temperature and emissivity are independent:

$$
\begin{cases}
\frac{\partial L^{(i,j)}_{\Delta\lambda,\text{total}}(\theta,\phi)}{\partial \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi)} = \tau_{atm}\tau_{opt}\left(L^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi,T^{(i,j)}_{\text{obj}}) - L_{\Delta\lambda,\text{env}} - L_{\Delta\lambda,\text{sun}}\right) \\[2mm]
\frac{\partial L^{(i,j)}_{\Delta\lambda,\text{total}}(\theta,\phi)}{\partial T_{env}} = \left(1 - \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi)\right)\tau_{atm}\tau_{opt}\frac{\partial L_{\Delta\lambda,\text{env}}}{T_{env}} = \left(1 - \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi)\right)\tau_{atm}\tau_{opt}\frac{RBe^{B/T_{env}}}{T^2_{env}(e^{B/T_{env}+C})^2} \\[2mm]
\frac{\partial L^{(i,j)}_{\Delta\lambda,\text{total}}(\theta,\phi)}{\partial \tau_{atm}} = \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi)\tau_{opt}L^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi,T^{(i,j)}_{\text{obj}}) + (1 - \epsilon^{(i,j)}_{\Delta\lambda,\text{obj}}(\theta,\phi))\tau_{opt}\left(L_{\Delta\lambda,\text{env}} + L_{\Delta\lambda,\text{sun}}\right) - \tau_{opt}L^{\circ}_{\Delta\lambda,\text{atm}}
\end{cases}
\tag{4.23}
$$

Those equations will be used in the following in order to observe the impact of the different model's parameters to the temperature retrieval.

Emissivity measurements have been made on the wood house (test site #2) in order to get an emissivity map, represented in Fig. 4.16. Table 4.1 shows the associated emissivity measurements.

Material	Color	Emissivity
Wood wall (west)	Blue	0.95
Wood wall (east)	Yellow	0.90
Grass	Green	0.95
Sidewalk	Red	0.89
Tar	Grey	0.95
Roof	Purple	0.92
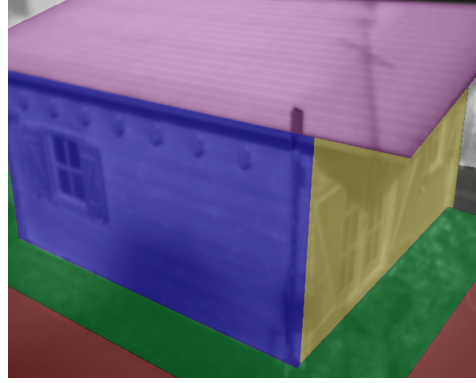
Table 4.1: Emissivity measurements values in the $8\mu m - 14\mu m$ band.

Figure 4.16: Emissivity map for the wood house, colors represents one particular emissivity measurement

Fig. 4.17 shows the effect of adjusting the emissivity to a particular material. Unfortunately, the comparison can only be made relatively due to the lack of ground truth temperature measurements at the surface of the different materials.
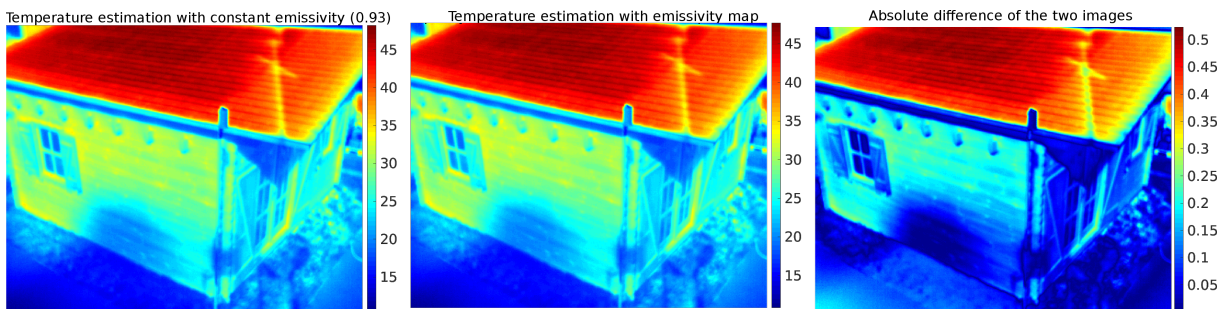
Figure 4.17: Effect of adjusting the emissivity for different material compared to a constant emissivity

However, one can see that adjusting the emissivity will make the temperature estimation more accurate. In fact, in this example, we have chosen the best candidate for a constant emissivity value over all the image because the average emissivity value is 0.93 with a standard deviation of 0.025 (derived from Table 1). As a consequence, we can see that even when the best constant emissivity value is chosen, this is still leading to, at most, $0.5K$ difference compared to the use of an emissivity map.

4.4.2 Sensitivity to sky temperature and atmospheric transmission

The sky temperature (T_{sky}) will also have an impact on the final temperature estimation ($\tilde{T}_{obj}$), as shown in Fig. 4.18 where the difference between the estimated sky temperature ($\tilde{T}_{sky}$) is compared to the difference made on the estimated object temperature, for different values of T_{obj} and T_{sky}. The R, B and F values used in the model to compute this figure are derived from a thermal calibration, as explained previously [109]. One can note that those coefficients are only valid for a limited range of object temperature ($283.15K - 343.15K$ in this case) and could expect some more important error out of this range. The following figures compare the differences between the actual object temperature and the estimated one by using estimates on T_{env} ($\tilde{T}_{env}$) and ϵ_{obj} ($\tilde{\epsilon}_{obj}$).
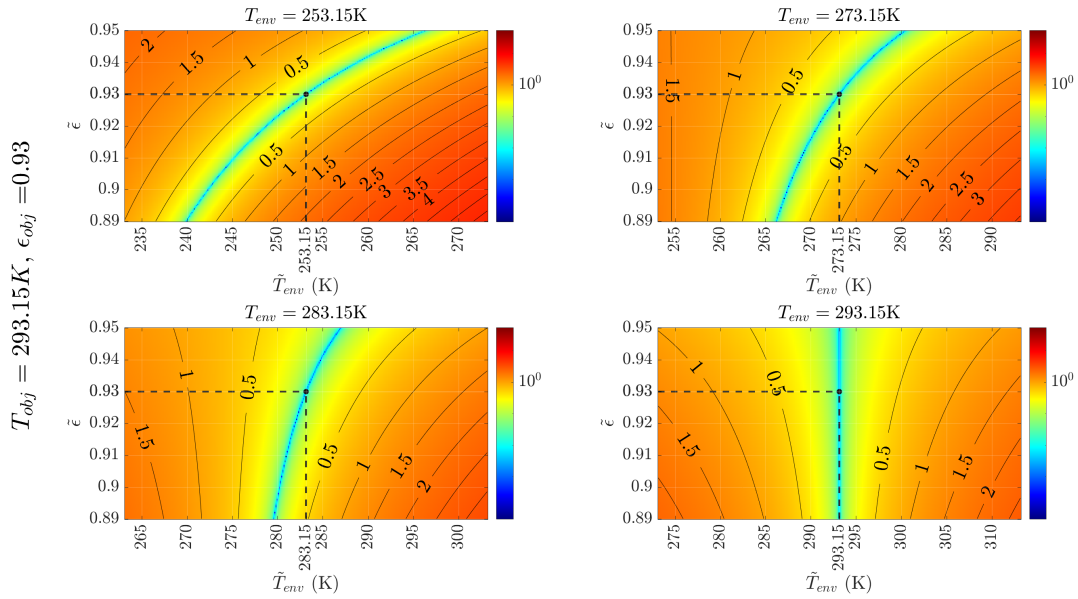
Figure 4.18: Difference between true object temperature and estimated one depending on the error made on the environment temperature and emissivity. In this case, the true object is at $293.15K$ and its emissivity is 0.93.

<u>How to read:</u> on the left the true object's temperature and emissivity. A point on the image represents the difference between the true object temperature and the estimated one with $\tilde{T}_{env}$ (x-axis) and $\tilde{\epsilon}_{obj}$ (y-axis) values. The images represent different environment temperatures values.

To overcome this, one can use the measurements made *in-situ* to get the sky temperature. In

93

Fig. 4.19, the ground truth temperature obtained from thermocouples data is compared to the temperatures estimated with and without sky correction. In the case of the sky correction, the sky temperature is derived from a pyrgeometer data whereas a constant sky temperature is taken in the other case. The Fig. 4.19 shows that taking into account the sky temperature from local measurements gives a better estimate of the object temperature. In this particular example, we compare the estimation of the temperature with and without the sky temperature correction to thermocouples measurements at the surface of a concrete pavement structure. The sky correction reduces in average the difference between the thermocouples measurements and the estimated temperature from infrared data.
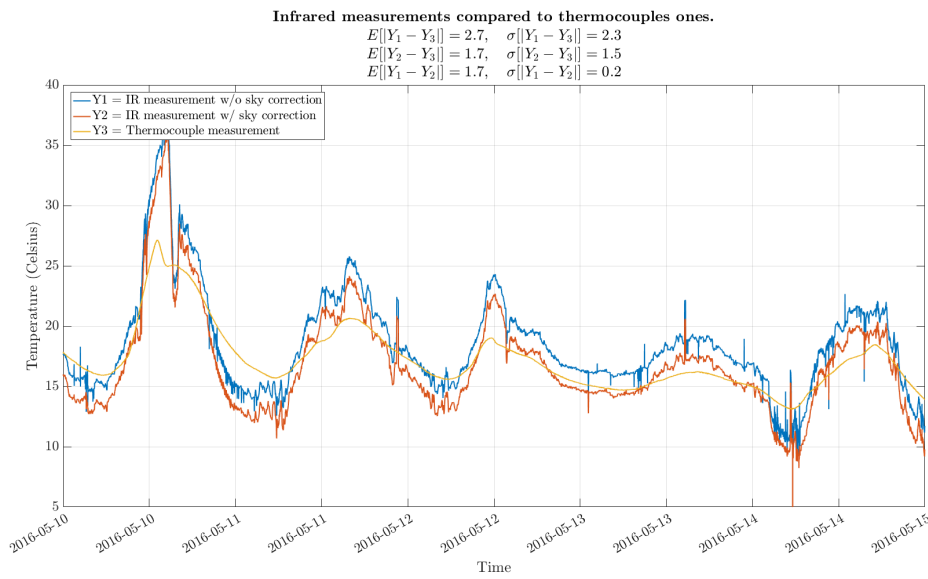
Figure 4.19: Comparison of ground truth temperature values to temperature estimation with and without sky correction

Another possibility is to use correlations from air temperature to estimate the sky temperature. Such correlations can be found in the literature [4]. Fig. 4.20 shows the effect of the Swinbanks correlation for two different periods: in december and august. Moreover, one can see that the sky temperature derived from the correlation will have an impact of the same magnitude's order as if using *in-situ* measurements.

If local measurements data are not available, the Swinbanks' formula can be used from meteorological air temperature data. The following figure shows the sky temperature derived from air temperature from three different data sources:

- Local weather station
- Copernicus' data store [44]
- METAR data

Even if differences exists in term of sky temperature (see Fig. 4.21), it represents a small quantity of thermal radiation, regarding the involved temperatures. Therefore, it is always possible to enrich the
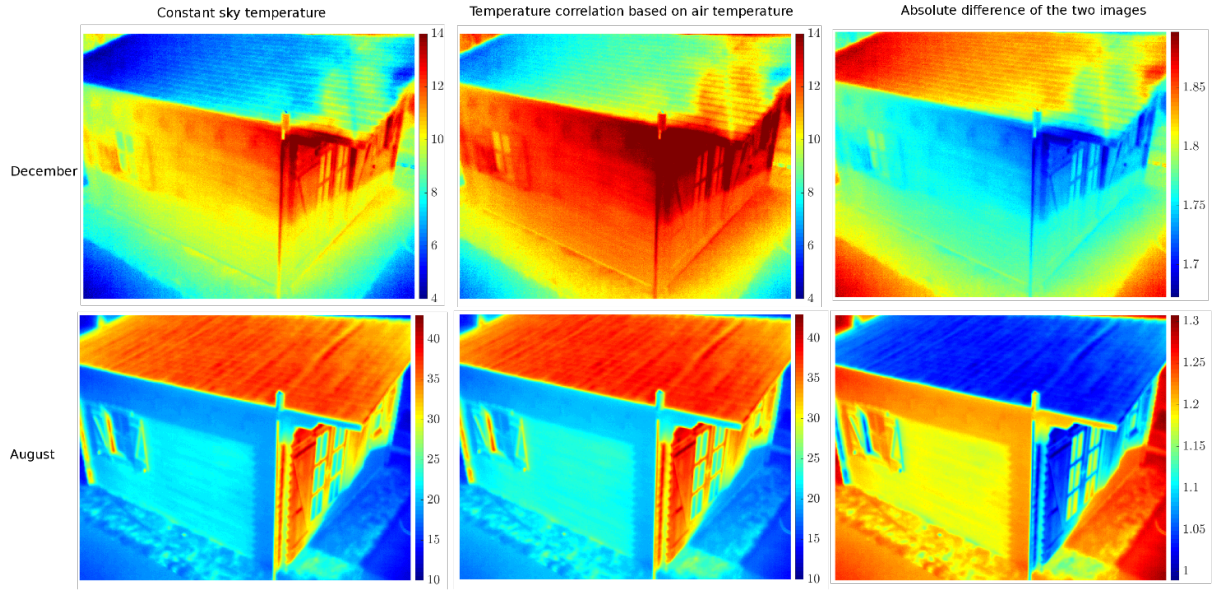
Figure 4.20: Effect of using a correlation for the sky temperature based on the air temperature

radiative model conversion thanks to online and open data.
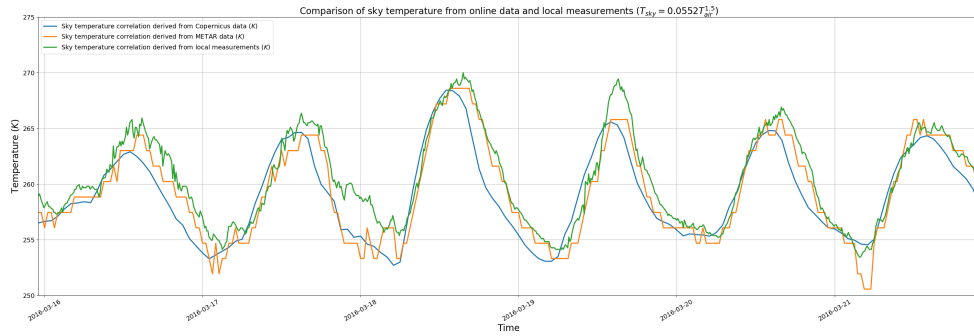
Figure 4.21: Sky temperature derived from different sources of air temperature

Based on Eq. (4.22), the differences maps can also be made for the atmospheric transmission, illustrated in Fig. 4.22.

Moreover, the atmospheric transmission is also influenced by the involved distances. In particular, when dealing with infrastructures monitoring applications, great distances differences may occur due to large field of views (*e.g.* during the monitoring of a bridge). In such case, a spatial calibration is necessary to compute a distance map and take into account those distances. As for example, the Fig. 4.23 shows the effect of the distance to the atmospheric transmission in the $7.5\mu m - 13\mu m$ band and the need for considering the distances due to the camera view projection within the image. As shown in this figure, considering the same distance between the observed surface and the camera may lead to inaccuracies.
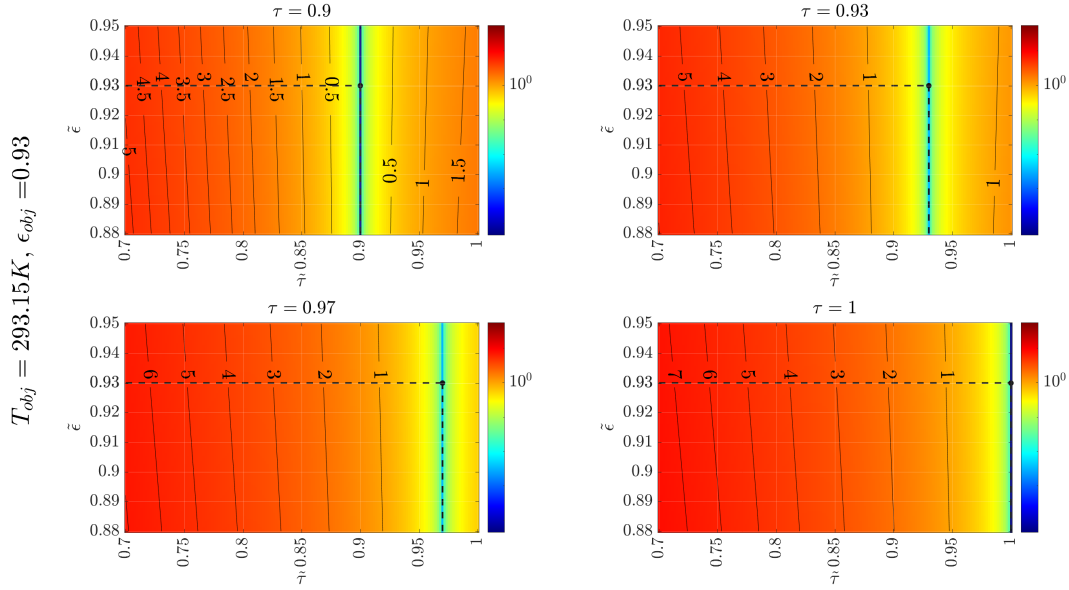
Figure 4.22: Difference between true object temperature and estimated one depending on the error made on the atmospheric transmission. Here is the difference map for an object temperature of $293.15K$ and emissivity of 0.93.
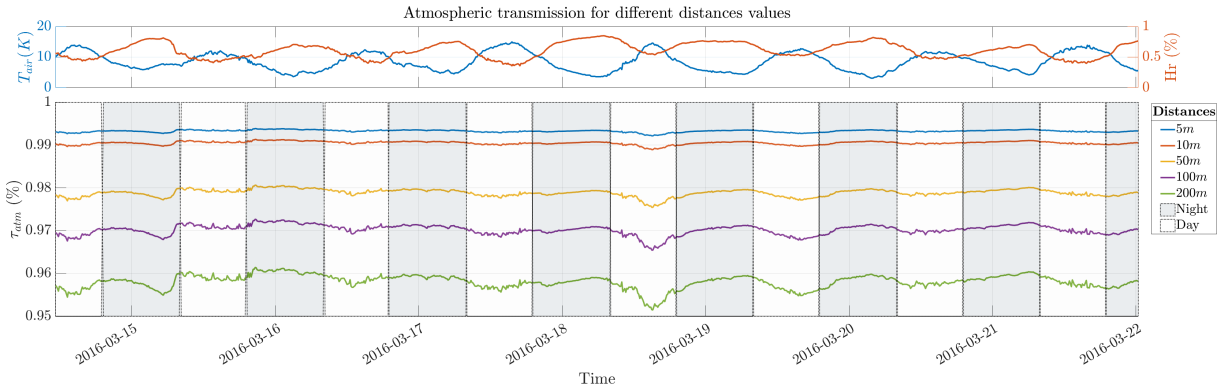
Figure 4.23: Different distance values for the atmospheric transmission, through time.

4.4.3 Sensitivity to sun contribution

The sun's irradiance will also have an impact on the final temperature estimation. In fact, depending on the emissivity of the observed object, the irradiance at camera's sensor will contain the sun's reflected contribution, both specular and diffuse. An estimation of the solar irradiance can be obtained thanks to a pyrgeometer for example. In the case of the LWIR ($7.5\mu m - 13\mu m$), even though the sun contribution may be small compared to the NIR, SWIR and MWIR bands, it stills represents a source of error in the conversion process. In particular for low object's temperatures as illustrated in Fig. 4.24,

we can see that the direct sun irradiance contribution can be responsible for up-to 3% of the object's emittance, leading to incorrect temperature estimation if not considered. In the case of SWIR-MWIR band the sun contribution should not be neglected, in particular when observing surfaces with low temperatures (Fig. 4.25).
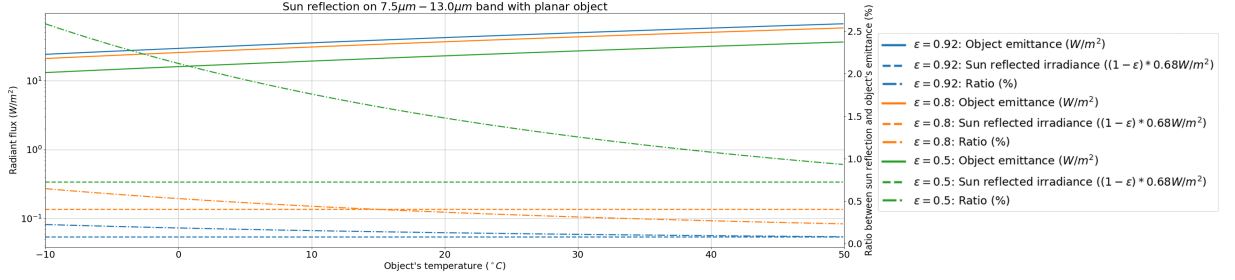
Figure 4.24: Sun irradiance reflection on $7.5\mu m - 13\mu m$ band for a normal object and different emissivity values
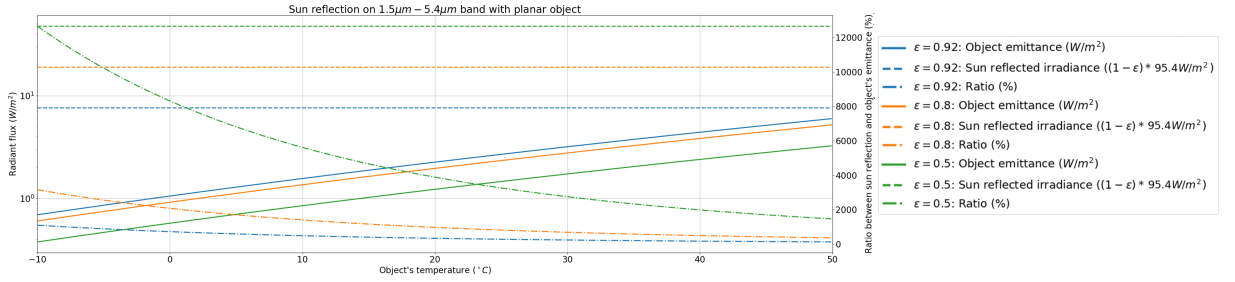
Figure 4.25: Sun irradiance reflection on $1.5\mu m - 5.4\mu m$ band for a normal object and different emissivity values

Moreover, one can see that the error made on the temperature estimation will therefore vary depending on the time of the day. When local measurements are not available, one can use literature's models. The Fig. 4.26 shows the SMARTS2 [165] (see next chapter) irradiance's simulation compared to a local pyranometer measure in the SWIR band.

4.4.4 Other parameters

Finally, rain and fog conditions will have an important impact on the flux received at cameras sensors. Fig. 4.27 illustrates the weather variation during a measurement campaign. Such variation will produce measurements that are difficult to exploit for accurate temperature estimation (see Fig. 4.28). This may not lead to accurate temperature measurements. However, such effect could be compensated by using a black-body source and perform automatic thermal calibration in case of rain or using adequate radiative transmission model (for instance in case of fog [62]).
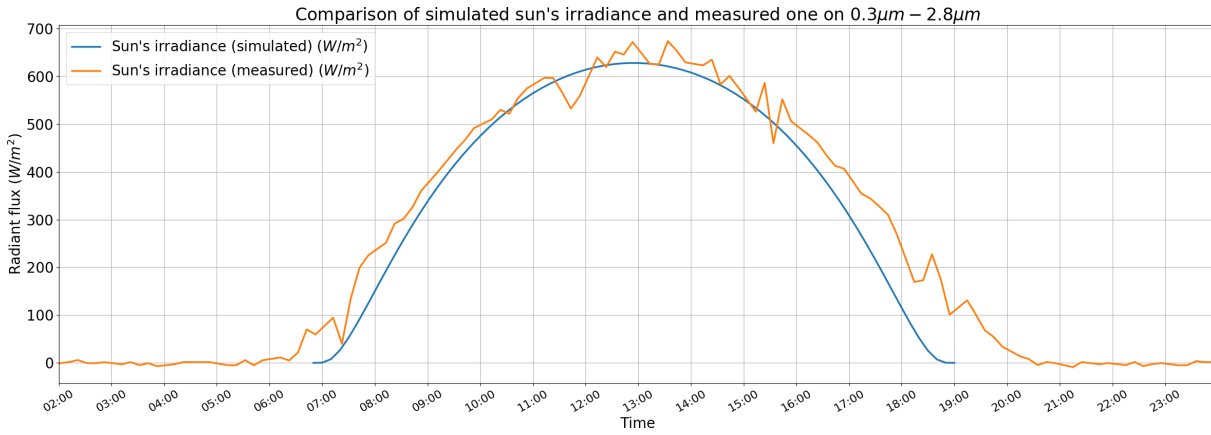
Figure 4.26: Comparison of sun's irradiance measured and simulated on may 5th in the $0.3\mu m - 2.8\mu m$ band
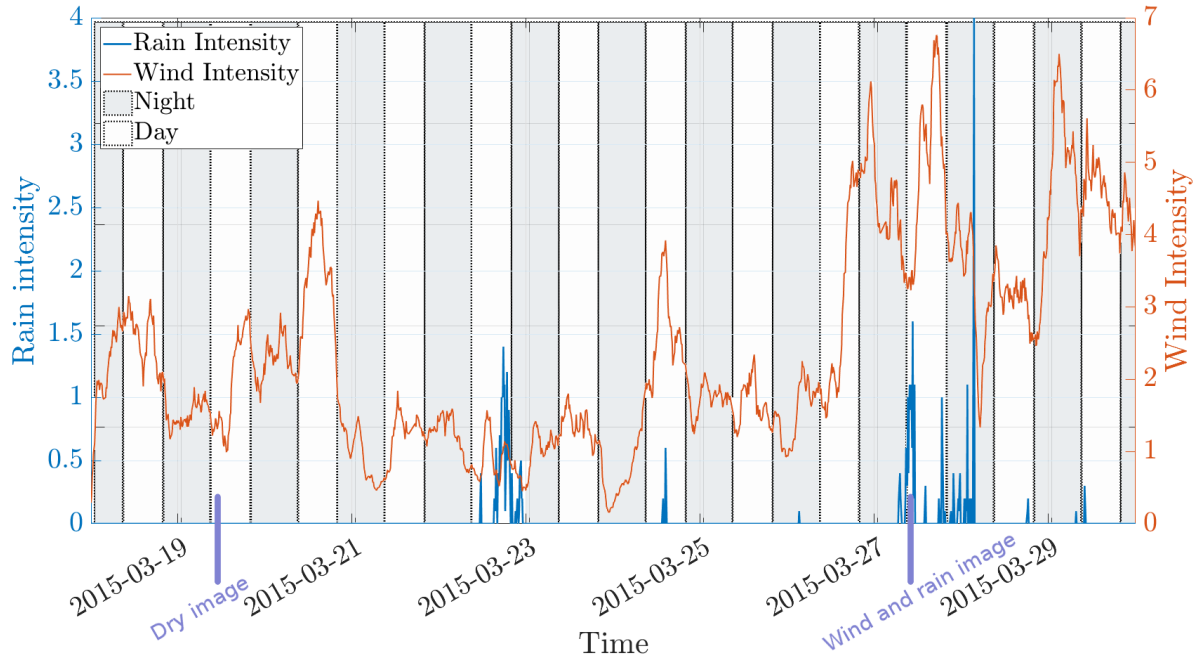
Figure 4.27: Rain and wind *in-situ* measurements

4.4.5 Summary

A review of the different parameters that may affect the temperature estimation has been made. Temperature estimation based on *in-situ* infrared thermography measurements needs to consider the geometry of the scene and the environmental parameters to get a better estimate. When local data are not available, it is possible to find online open data thanks to meteorological reports or forecasts. The Fig. 4.29 shows an example of the amount of data that can be gathered during a long-term experiment

(a) Dry situation

(b) Raining and optic issues
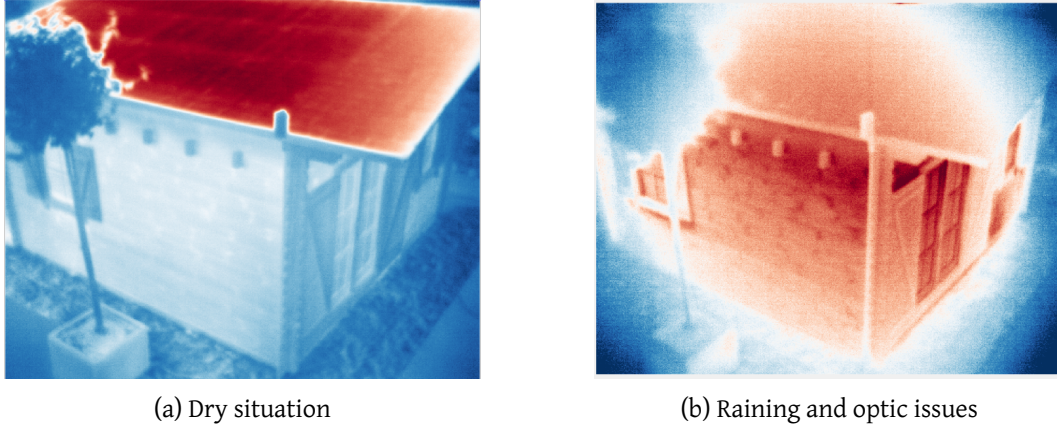
Figure 4.28: IR image view: in dry environment (a), rain with optic issues (b)

with missing data in red due to various experiment issues (sensor failure, current shortage *etc.*). This study showed that even if those measurements are missing, it is still possible to estimate those data thanks to open access online data to enable the usage of a refined model.
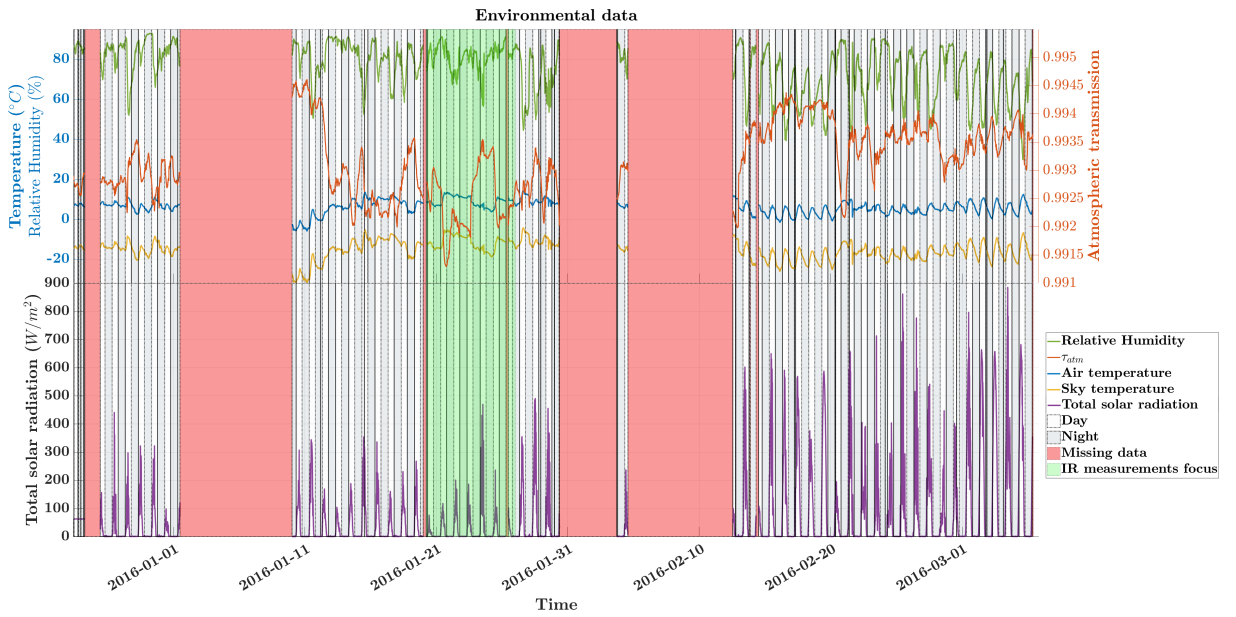
Figure 4.29: A set of environmental data for long-term thermal monitoring

Fig. 4.30 shows a comparison between the temperature conversion with and without the consideration of the environmental parameters for two measurement points on the house: on the roof and on the wall. The measurements are made on the $7.5\mu m - 13\mu m$ band for which the sun contribution is relatively low (see Fig. 4.24). In this particular case, the final temperature estimation difference with the multi-sensors conversion and without it can be up-to $2K$ and could be even more in a NIR band due to the sun influence. In this example, the error made on the sky contribution and the emissivity
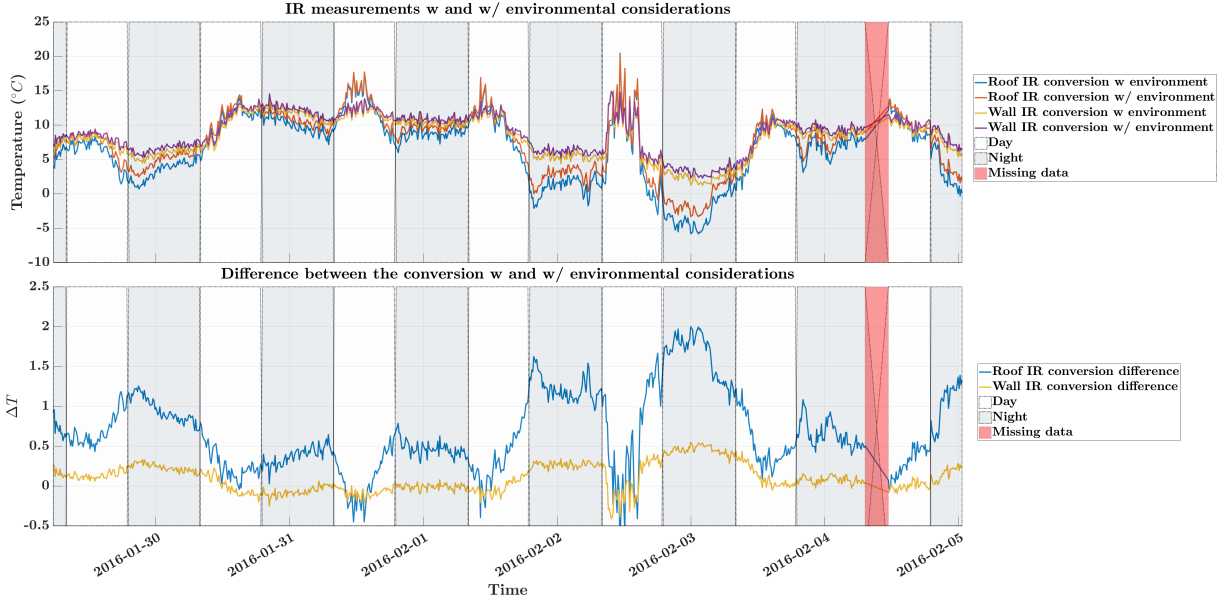
Figure 4.30: Comparison between the temperature estimation by considering $M_{tot} = \epsilon_{obj} M_{obj}$ or using the presented model and multi-sensor data

one will be the main parameters that will influence the temperature estimation. Here, the open access online data to fix the missing ones are not used since the IR measurements are also lacking.

4.5   Synthesis

The long-term infrared thermography measurement process has been reviewed in this chapter: from the calibration of the sensors to the actual data processing tools. Moreover, the sensitivity of the simplified radiometric equation to its parameters has been studied. It was shown that estimating the parameters through multi-sensors data is efficient. When local sensors data are not available, it is still possible to use online data from various sources (METAR, Copernicus, *etc.*). After such compensation, one of the most important source of temperature retrieval error is the emissivity. This is why temperature / emissivity retrieval methods are fundamental in the infrared thermal monitoring process. The objective of the last chapter is to propose simultaneous temperature / emissivity separation methods for long-term infrared thermal monitoring. However, finding back the presented *in-situ* data is a challenging task, particularly because there is no ground truth to trust. As a consequence, an infrared multi-spectral images simulation tool has been developed. This simulation tool is a way to test the developed methods on simple cases. Then, noise can be added by the simulator by considering various environmental parameters. The next chapter will present this simulator based on the progressive radiosity algorithm. Then, developed methods will be presented and finally tested via simulated images.

Study and development of an infrared multispectral images simulator

Aforward solution simulation software has been implemented in order to generate data based on 3D object models. The objectives of this software are to simulate radiative diffuses exchanges between surfaces on a 3D scene as well as integrating atmospheric and infrared camera models for *in-situ* infrared thermography. The spectral bands used in the software can be adjusted and the computation time has been optimized by using graphical hardware acceleration. Different applications can be done with such simulation tool: simulation's generated data are an efficient and quick way to test temperature and emissivity separation methods robustness or infrared thermal monitoring systems sizing for example. The proposed software gives a forward solution to the diffuse radiative transfer equation through the radiosity method, more detailed in this chapter. The radiosity equations and the form factors computation will be firstly presented. The progressive radiosity method used in the developed software and its implementation on graphic hardware is introduced and discussed. Then, models to simulate solar spectral irradiance at ground and camera's detectors are conferred to finish with some practical applications examples.

5.1 Radiosity method

The radiosity method has been chosen for its physical equations solving approach. The idea is to solve a simplified linear system representing all the contributions of the elements present in the scene. Such algorithm can be combined with Monte-Carlo methods [157] or the equations to be solved can be extended to consider multiple physical phenomena [162, 186] for considering advanced reflections, participating medium, Fresnel effect, *etc.* Contrary to Monte-Carlo algorithm, such approach renders the 3D scene globally, *id est* the generated image does not depend on the viewpoint. Once the scene has been rendered, the camera can be moved inside to inspect each surface. Moreover, Monte-Carlo algorithms suffer from sampling issues and noise. The advantage of Monte-Carlo methods are their ability to consider participating medium, specular reflections and directional reflectance functions easily. However, in order to get accurate results, multiple bounces are needed which means casting a lot of rays which make the complexity and the time processing more important.

5.1.1 Radiosity equations

The different radiosity methods are all based on the radiosity equation. The aim of the different methods is to solve this linear equation as fast as possible either by providing improved algorithms or by giving an approximation of the solution. The assumptions made here are nonparticipating medium, with orthotropy and lambertian diffuses materials.

Let A_1 and A_2 be the surface of two elements denoted 1 and 2. The form factor $F_{1\rightarrow2}$ which characterizes the geometric interaction between those two diffuse elements is defined in Eq. (5.1) [93]

$$F_{1\rightarrow2} = \int_{A_1}\int_{A_2} \frac{\cos(\theta_1)\cos(\theta_2)}{\pi r^2}\mathrm{d}A_1\mathrm{d}A_2 \qquad (5.1)$$

For a given spectral band $\Delta\lambda_i$, the radiosity $B_{k,\Delta\lambda_i}$ of a patch k is given by its self emission $E_{k,\Delta\lambda_i}$ added to the contribution of all others patches j present on the scene. Therefore, after discretization (see Eq. (3.60)):

$$J_{k,\Delta\lambda_i} = M_{k,\Delta\lambda_i} + (1 - \epsilon_{k,\Delta\lambda_i})\sum_{j=1,j\neq k}^{j=N_{elements}} V_{kj}F_{k\rightarrow j}J_{j,\Delta\lambda_i} \qquad (5.2)$$

Figure 5.1: Form factors for two infinitesimal elements

where V_{kj} represents an occulting factor

$$V_{kj} = \begin{cases} 1 & \text{if } k \text{ sees } j \\ 0 & \text{otherwise} \end{cases} \qquad (5.3)$$

One can see that three main steps appear from equation 5.3. First, the visibility between the different patches must be solved. Then, the form factors must be computed and finally the system has to be solved. The next part will expose the different methods used for solving those three main steps.

5.1.2 Form factors computation

The direct calculus of the integral of the form factors is computationally intensive, in particular when an important amount of patches is used. Therefore, different methods have been developed for improving the computation speed of the form factors. Many tables of pre-computed form factors can be found in the literature and used as-is in lookup tables [93]. However, such technique cannot be generalized and can only be used on simple cases. In the following, some of the main methods to compute or approximate the form factors are presented.

Contour integration

Instead of being evaluated through the areas, the integral can be evaluated through a contour integration, by applying Stokes' theorem. As a consequence, the multiple integration over a surface area is reduced to a single integration around the boundary of the area [131, 170]. In such context, the form factor from an infinitesimal element to a finite area $F_{d1 \to 2}$ can be expressed as a sum of single integrations. Such method improves both accuracy and computing time as shown in [155].

Nusselt analog

The form factors can also be computed through the unit sphere method introduced by Nusselt [132]. The idea is to build a hemisphere around the differential element $\mathrm{d}A_1$ and project onto the hemisphere plane the area to be integrated. In such case, the form factor from $\mathrm{d}A_1$ to A_2 becomes:

$$F_{d1 \to 2} = \frac{1}{\pi} \int_{A_2} \cos(\theta_1) \frac{\cos(\theta_2)\mathrm{d}A_2}{r^2} = \frac{1}{\pi} \int_{A_2} \cos(\theta_1) d\Omega \tag{5.4}$$

($d\Omega = \mathrm{d}A_p = \frac{\cos(\theta_2)\mathrm{d}A_2}{r^2}$). Moreover, $\cos(\theta_1)\mathrm{d}A_p$ is the projection of $\mathrm{d}A_p$ onto the hemisphere plan (see Fig. 5.2). By integrating the previous quantities, we can derive the form factor such as:

$$F_{d1 \to 2} = \frac{1}{\pi} \int_{A_2} \cos(\theta_1) d\Omega = \frac{A_n}{\pi} \tag{5.5}$$
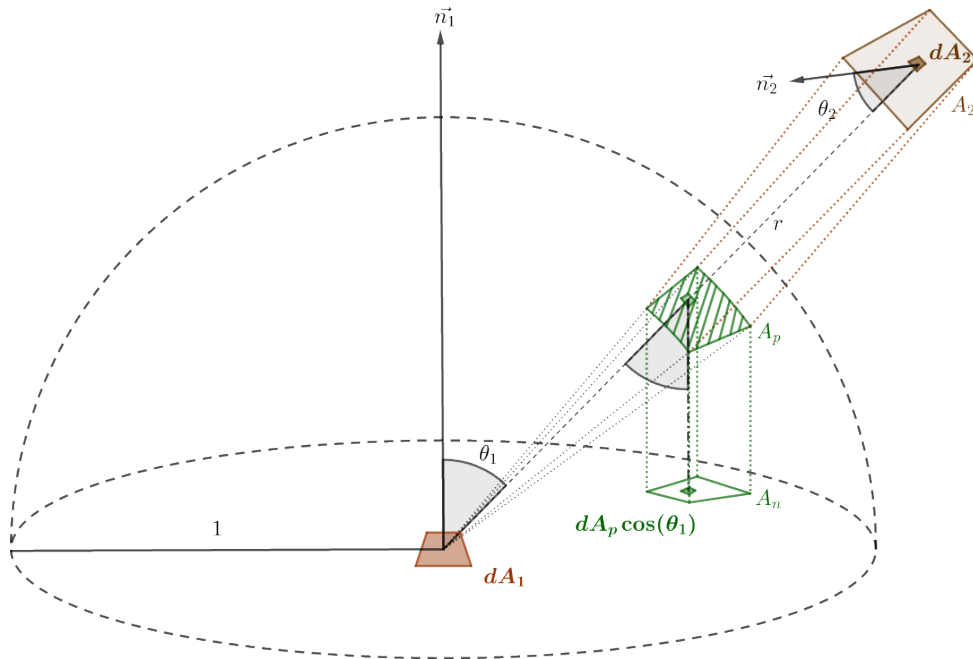
Figure 5.2: Nusselt's analog

103

The relation Eq. (5.5) can be used to improve the computation speed of the form factors by using the hardware acceleration [79]. In practical, the Z-Buffer hardware algorithm can be exploited to compute the form factors and patches visibility at the same time. However, one drawback of this method is that although polygon edges project to curves on a hemisphere, rasterization produces only straight edges. As a consequence, tessellation should be important in the image. More details for such projection are given in the next section.

Hemicube method

One of the most common method used for computing the form factors is the hemicube approach. Rather than using a unit-sphere, the projection is done on a cube centered over the receiving element [39]. Such method has been developed for its simplicity and efficiency to be implemented on hardware. However, this method needs to render each face of the hemicube (5 rendering) and depends on the resolution of the chosen hemicube. Small elements that fall into a too large hemicube's resolution will have an important form factor approximation. The Z-Buffer algorithm can also be used for determining the visibility and form factors at the same time. Moreover, in recent hardwares, a special hemicube buffers are available to render hemicubes efficiently. One drawback of the hemicube method is its linear projection that induces issues at edges of the hemicube, particularly for coarse meshing [40].

Single plane method

Different methods have been developed through years for increasing the computation speed such as the planar projection proposed by F. Sillion [162]. In this method, a projection plane just above the surface of interest on which other elements are projected is used. This plane is then subdivided adaptively depending on the size of the projected elements to improve sampling. The advantage of this method is to reduce the number of rays to be thrown for the computation of form factors. In particular, in this study, F. Sillion introduces extended form factors for computing specular component.

Monte-Carlo method

Similarly to the ray tracing algorithms for rendering presented in Chapter 3, a ray casting approach is possible to compute the form factors. One of the main advantages of this technique is its ability to handle many surface types, including curved ones. Moreover, it is also possible to get *a priori* upper bound as described in [140]. However, one drawback is that the expense can be more important depending on the distribution to approximate [40]. In Monte-Carlo methods, the full area-to-area form factor integral can be approximated or the Nusselt's analog can be derived [40].

5.2   Numerical solution to the linear radiosity system

Once the form factors have been computed, the radiosity system can be solved. The Eq. (5.2) shows that the radiosity system is a set of linear equations which can be summarized by:

$$Ax = b \tag{5.6}$$

where A is the interactions matrix, x a vector of radiosity values and b a vector of emission terms. By combining Eq. (5.2) and Eq. (5.6), we can get the shape of A:

$$A = \begin{pmatrix} 1 - (1 - \epsilon_{1,\Delta\lambda_i})F_{1\rightarrow 1} & -(1 - \epsilon_{1,\Delta\lambda_i})F_{1\rightarrow 2} & \dots & -(1 - \epsilon_{1,\Delta\lambda_i})F_{1\rightarrow n} \\ (1 - \epsilon_{2,\Delta\lambda_i})F_{2\rightarrow 1} & 1 - (1 - \epsilon_{2,\Delta\lambda_i})F_{2\rightarrow 2} & \dots & -(1 - \epsilon_{2,\Delta\lambda_i})F_{2\rightarrow n} \\ \dots & \dots & \dots & \dots \\ (1 - \epsilon_{n,\Delta\lambda_i})F_{n\rightarrow 1} & (1 - \epsilon_{n,\Delta\lambda_i})F_{n\rightarrow 2} & \dots & 1 - (1 - \epsilon_{n,\Delta\lambda_i})F_{n\rightarrow n} \end{pmatrix} \tag{5.7}$$

> **Note**
>
> Most of the time, the diagonal coefficient of A are equals to 1. In fact, one assumption is to consider that one element cannot "see" itself (in practical, the meshing is supposed to be refined enough and the involved patches are often planar surfaces). Therefore, in the following for all $i \in \mathbb{N}$, $F_{i\rightarrow i} = 0$.

Many numerical methods for solving linear equation's systems can be found in the literature. It should be noted here that the system involved in the radiosity is a large matrix most of the time. Furthermore, when dynamic scenes are involved, the main difficulty is that the matrix will be changed and that it should be inversed again. As a consequence fast algorithms are needed. Solving such system is a large active field on its own, so only a review of the main algorithms used in the literature and applied to the radiosity's equation is proposed here. The next section focuses on methods for solving the linear system Eq. (5.6) of unknown x.

5.2.1   Solving the linear system

Once the system has been solved, the radiosity values are found and the rendering is completed. As a consequence, finding the exact solution in a fast way seems to be the best option. However, finding the inverse of A is non trivial and simple inversions methods (*e.g.* Gauss-Jordan elimination or LU-decomposition) will not work to solve the system in a reasonable amount of time. Those algorithms are usually in order of $O(n^3)$ which is not suitable for the large matrix that is involved in the radiosity equation system. Instead, iterative methods that provide an approximation of the solution will be preferred.

5.2.2 Approximate solution to the linear system

Some iterative methods that could be used for solving the linear radiosity system of equations are presented in this section and translated to the radiosity problem [40]. Most of these methods are available in standard numerical texts books. In this part, the idea is to find the solution to the linear system $Ax = b$, with $A = (a_{ij})_{1\le i,j\le n}$, $x = (x_i)_{1\le i\le n}$. Then at the end of the proposed method, the algorithm is developed for the radiosity problem.

Relaxation methods

The basic idea behind relaxation methods is to compute iteratively the result of the system $Ax = b$ for large systems.

Fixed point methods In order to introduce the Jacobi and Gauss-Seidel methods, let decompose A such as $A = G - H$ with G invertible:

$$Ax = b \Leftrightarrow x = G^{-1}Hx + G^{-1}b \tag{5.8}$$

The previous linear equation system is therefore transformed to a problem fixed point search [1]:

$$\begin{cases} x_0 & \in \mathbb{R}^n \\ x^{(k+1)} & = G^{-1}Hx^{(k)} + G^{-1}b \end{cases} \tag{5.9}$$

If this sequence converges, then its limit is the solution to the previous linear system $Ax = b$.

Jacobi method The Jacobi method has been introduced by C.G.J. Jacobi in the XIXth century. Based on the previous equation, A can be decomposed into two components, a diagonal matrix G (which is invertible provided that its coefficients are also invertibles, assumed here) and a remainder H constituted of the others elements of A:

$$A = G - H \quad \text{with} \quad G = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix} \quad \text{and} \quad H = - \begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & 0 \end{pmatrix} \tag{5.10}$$

---

[1]The superscript $^{(k)}$ denotes the kth complete iteration. The superscript $^{(0)}$ corresponds to a prior given as input.

The matrix $G^{-1}H$ is then called the Jacobi matrix and the system can be solved iteratively:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1, j\neq i}^n a_{ij}x_j^{(k)}}{a_{ii}} \tag{5.11}$$

A sufficient condition for this method to converge for all $x^{(0)}$ is to have a diagonally dominant matrix. Let now r be the residual vector: $r = Ax - b$. In that case, we have:

$$r_i^{(k)} = b_i - a_{ii}x_i^{(k)} - \sum_{j=1, j\neq i}^n a_{ij}x_j^{(k)}$$

$$\Rightarrow x_i^{(k+1)} = x_i^{(k)} + \frac{r_i^{(k)}}{a_{ii}} \tag{5.12}$$

Where the residuals are updated after each iteration.

Algorithm 6: Pseudo-code for Jacobi radiosity solver

Initialization $x_i^{(0)} = emission_i, \quad r = b - Ax$

while *not converged*

    for $i \in Faces$

        $x_i^{(k+1)} = x_i^{(k)} - \frac{r_i^{(k)}}{a_{ii}}$

    $r^{(k+1)} = b - Ax^{(k+1)}$

return x

Gauss-Seidel The Gauss-Seidel method is a slight variation of the Jacobi one. Instead of taking the diagonal matrix as G, the Gauss-Seidel method uses the lower triangular part of the matrix:

$$A = G - H \quad \text{with} \quad G = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & 0 & \dots & a_{nn} \end{pmatrix} \quad \text{and} \quad H = -\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \tag{5.13}$$

By using forward substitution, a sequential formula is obtained:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}}{a_{ii}} \tag{5.14}$$

The differences between the Gauss-Seidel and the Jacobi methods reside in the amount of elements that are needed to be stored. For a very large problem, Gauss-Seidel is preferred due to the fact that only one storage vector is required due to the fact that elements can be overwritten as they are computed.

Similarly to the Jacobi method, one can express the iteration based on the residuals:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{r_i^{(k)}}{a_{ii}} \tag{5.15}$$

Where this time the residuals are updated after each step.

Algorithm 7: Pseudo-code for Gauss-Seidel radiosity solver

Initialization $x_i^{(0)} = emission_i$

while *not converged*

    for $i \in$ *Faces*

        $x_i^{(k+1)} = x_i^{(k)} + \frac{r_i^{(k)}}{a_{ii}}$

        $r_i^{(k)} = b_i - \sum_{j=1}^{n} A_{ij} x_j^{(k)}$

return x

By looking more closely to the Gauss-Seidel algorithm and its physical interpretation in terms of radiosities, one can see that at each step a surface radiosity value i is updated by summing all the contribution of the others surfaces on the scene. Physically speaking, this step is equivalent to take a single patch and consider the radiation from all the surrounding patches which is called a gathering step in the global illumination field, as illustrated in Fig. 5.3.
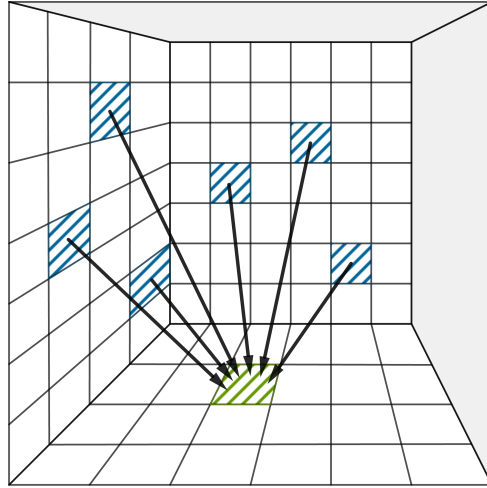
Figure 5.3: Gathering step as in the Gauss-Seidel algorithm, figure inspired from [40]

Southwell method A variant of the Gauss-Seidel method is the Southwell algorithm. Developed during the 40', the Southwell algorithm was made for solving linear systems for humans. In this method, instead of relaxing in the ith order, the relaxation is performed on the element with the greatest residual. Such ordering is called a Southwell iteration [77, 70]. However, in order to know which element has the

greatest residual at iteration k it seems that all the r_i should be computed. To overcome this, a trick is to make use of the changes $\Delta x^{(k)}$ between one approximation to the other:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} \tag{5.16}$$

The residuals are given by:

$$r^{(k+1)} = b - A\left(x^{(k)} + \Delta x^{(k)}\right) = r^{(k)} - A\Delta x^{(k)} \tag{5.17}$$

Since for a given iteration, only $\Delta x_{\arg\max r_i^{(k)}, \forall i \in [1,n]}$ is not zero, we have:

$$\forall j \in [1, n], \quad r_j^{(k+1)} = r_j^{(k)} - \frac{a_{ji}}{a_{ii}} r_i^{(k)} \tag{5.18}$$

In such case, the point fixed algorithm is applied to the residuals. The initialization is straightforwardly adapted:

$$r^{(0)} = b \quad x^{(0)} = 0 \tag{5.19}$$

The pseudo-code for the Southwell relaxation is given in Algorithm 8

Algorithm 8: Pseudo-code for Southwell radiosity solver

> **Initialization** $r_i = emission_i \quad x_i = 0$
>
> **while** *not converged*
> > $i = \arg\max\limits_{j \in [1,N]} r_j$
> > $x_i = x_i + \frac{r_i}{a_{ii}}$
> > $temp = r_i$
> > **for** $j \in Faces$
> > > $r_j = r_j - \frac{a_{ji}}{a_{ii}}$
>
> **return** x

If each step of the Gauss-Seidel algorithm is $O(n)$, it is more complicated to characterize the complexity of the Southwell algorithm since one given element may be visited multiple times before visiting some other element. The physical comparison with the Gauss-Seidel method can be made. In the Southwell solver, the greatest residual is taken and then the residual of the other surfaces are updated, at each step. On the contrary of Gauss-Seidel, a given element updates the others: it is called a shooting step.
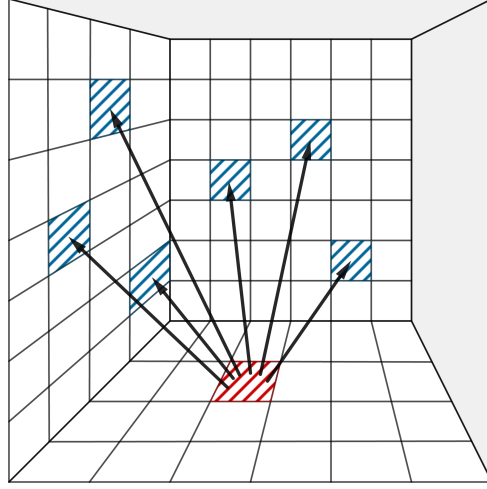
Figure 5.4: Shooting step as in the Southwell algorithm, figure inspired from [40]

> **Note**
>
> The Southwell method is actually a successive over-relaxation (SOR) method. A SOR method is a variant of the Gauss-Seidel one on which a relaxation factor is incorporated and the A matrix is defined by $A = D + L + U$ where D is diagonal, L lower and U upper triangulars matrices.
>
> $$(D + \omega L)x_{k+1} = \omega b - (\omega U + (\omega - 1)D) x_k \qquad (5.20)$$
>
> For a positive definite matrix, the algorithm converges for $\omega \in [0, 2]$. For $\omega = 1$, the equation is equivalent to Gauss-Seidel.

Progressive Radiosity

Based on the Southwell algorithm, [41] proposed a similar approach to render the radiosities which enables both computation efficiency and rendering called *progressive radiosity* or sometimes *progressive refinement*. As for the Southwell method, the progressive radiosity method processes the patches in a sorted order according to their energy contribution to the environment. This approach avoid computing all the form factors at the beginning of the algorithm. Thus, each iteration needs the computation of the form factor from the current patch to all other patches which can, however, lead to redundant computations. The pseudo-code for the progressive refinement method is presented in Algorithm 9. The Southwell method is slightly changed so that the result can be shown progressively. To achieve this, an additional variable is introduced: *unshot* radiosities which can be stored during the process. Taking into account the unshot radiosities is a way to take into account what part of energy remains to

be distributed. At each iteration, the current approximation can be displayed.

Algorithm 9: Pseudo-code for progressive radiosity

Initialization $x_i = emission_i \quad \Delta x_i = x_i$

while *not converged*

  $i = \underset{j \in [1,N]}{\arg \max} (x_j \text{Area}_j)$ `//Maximum of energy`

  for $j \in$ *Faces*

   $\Delta u_i = \Delta x_i a_{ji}$ `//Unshot energy to be distributed to j`

   $\Delta x_j = \Delta x_j + \Delta u_i$ `//j will have to distribute this part of energy`

   $x_j = x_j + \Delta u_i$ `//Update the approximation of j`

  $\Delta x_i = 0$ `//i has shot its unshot energy`

  dispImage(x)   `//Display current approximation`

return x

By looking at the algorithm, one can see that by changing x in the progressive radiosity algorithm by $x+r$ in the Southwell one gives the same algorithm. One difference though is that in the progressive radiosity, the largest energy to shoot is taken instead of the largest residual. Fig. 5.5 shows an example of the progressive radiosity algorithm, for which the implementation is detailed in the next section.
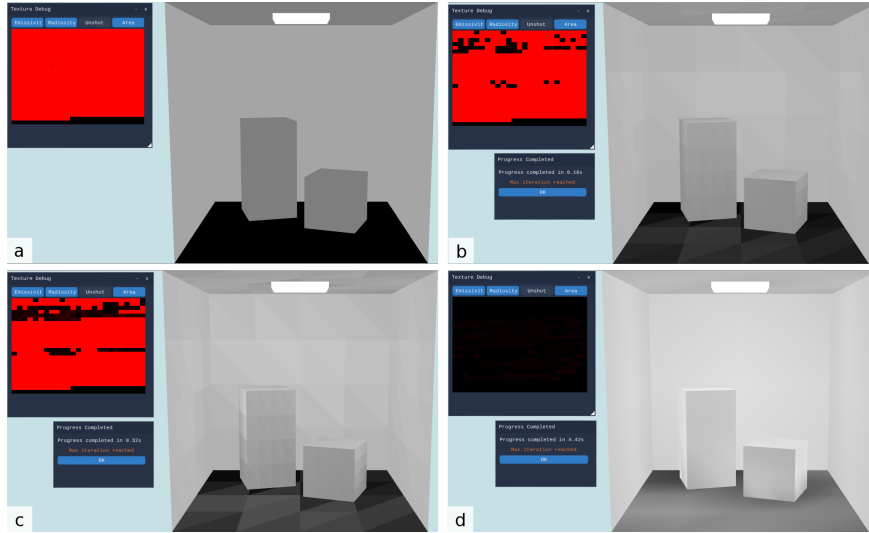
Figure 5.5: Progressive radiosity examples with a Cornell's Box where all surfaces emit in $[1.5\mu m - 5.4\mu m]$ band. (a) Initial state. The ceiling source is at $353.15K$ and the objects at $293.15K$ with different emissivities. (b) After 10 iterations ($\approx 0.15s$). (c) After 100 iterations ($\approx 0.3s$). (d) After 1000 iterations with interpolation to smooth the result ($\approx 4.4s$). Almost all surfaces have shot their unshot radiosity (red texture).

5.3   Implementation on accelerated hardware

The progressive refinement approach described previously has been implemented in C++. Parallel radiosity is difficult to have in practice since the computation of the visibility and the form factors in the scene implies many links between the different patches of the scene and therefore to be able to know at a given time the value of other patches. The objective of the software is to provide a fast way to render the radiosities of a 3D scene thanks to recent hardware and software technologies as long as returning physical quantities that can be exploited for infrared thermography. The OpenGL®'s API [134] has been used for the graphics rendering, Assimp's library [11] for loading 3D objects and Nuklear's library for rendering the GUI (Graphical User Interface) elements [126].

5.3.1   Hardware acceleration and OpenGL®

OpenGL® (Open Graphics Library) is an API for rendering 2D and 3D vector graphics. This library defines a set of abstract functions that are normalized among the Khronos Group consortium. Graphics processing unit vendors provides an implementation of OpenGL® (and additional functionalities specifics to each vendors/cards) with their graphic cards so that softwares can interact with the GPU thanks to OpenGL®. OpenGL® is used in many types of applications from video games (see Fig. 5.7) to CAD.

Initially, the rendering of any object in 3D was normalized through a rendering pipeline (see Fig. 5.6) that includes automatically and efficiently the tesselation, the clipping, the rasterization, the memory management, the buffers management, *etc.* Such architecture was designed to be handy, so that the user only had to create 2 type of scripts : one for handling the geometry and manipulating vertices (called *vertex shader*) and one for handling the rendered image (called the *fragment shader*). However, with the development of new technologies and new needs for imaging and

Figure 5.6: OpenGL® pipeline. Blue boxes are programmable steps and dashed lines correspond to optional stages [191].

rendering, the pipeline has been extended so that any part of the pipeline could be customized through scripts. More recently, with the apparition of GPGPU needs for scientist and computing resources, *compute shaders* appeared with OpenGL® version 4.3. This time, the *compute shaders* replace entirely the pipeline and users can perform data manipulation on the GPU with OpenGL®. This feature has been used in our implementation in order to optimize the computation time of the algorithm.

Figure 5.7: Mirror's Edge Catalyst driven by Unreal Engine 3, which can interface OpenGL® - *Credits to deadendthrills.com*

5.3.2 Progressive radiosity GPGPU implementation

The *compute shader* enable a GPGPU approach within the OpenGL® API which appears to be an interesting architecture for the radiosities. The computation of the radiosities is done in a texture through a *compute shader*. This texture is read for the rendering through a classical OpenGL® pipeline by using a *vertex shader* and a *fragment shader*. This architecture minimizes CPU/GPU transfers. The size of the texture depends on the number of bands and number of patches. The actual texture stores radiosities values as-is and should be improved in the future by using mipmap textures or even hierarchical storage. A flowchart of the developed architecture is presented in Fig. 5.8.

The main steps of the developed software are summarized below:

1. The 3D geometry of the scene is loaded thanks to a 3D file model description
2. Textures are initialized (radiosities values, residuals, areas, etc.) and stored on the GPU memory through 32bits 2D textures
3. A first projection/rendering is done on a hemisphere onto a buffer
4. The highest patch is selected on the GPU
5. The progressive radiosity algorithm is executed with all the previous computed data on a compute shader

The algorithms presented previously are derived from realistic images synthesis that mainly work in the visible spectrum (generally 3 bands). In this particular case, sources (that emits light) must be distinguished from other elements. As a consequence, in the classical approach the radiosities are initialized to a floating-point (usually between 0 and 1), representing the light intensity if it is a source and to 0 otherwise. The values are then kept on $[0;1]$ interval since only the rendering view aspect really matters. However, in the infrared spectrum any patch will emit radiation and in order to retrieve
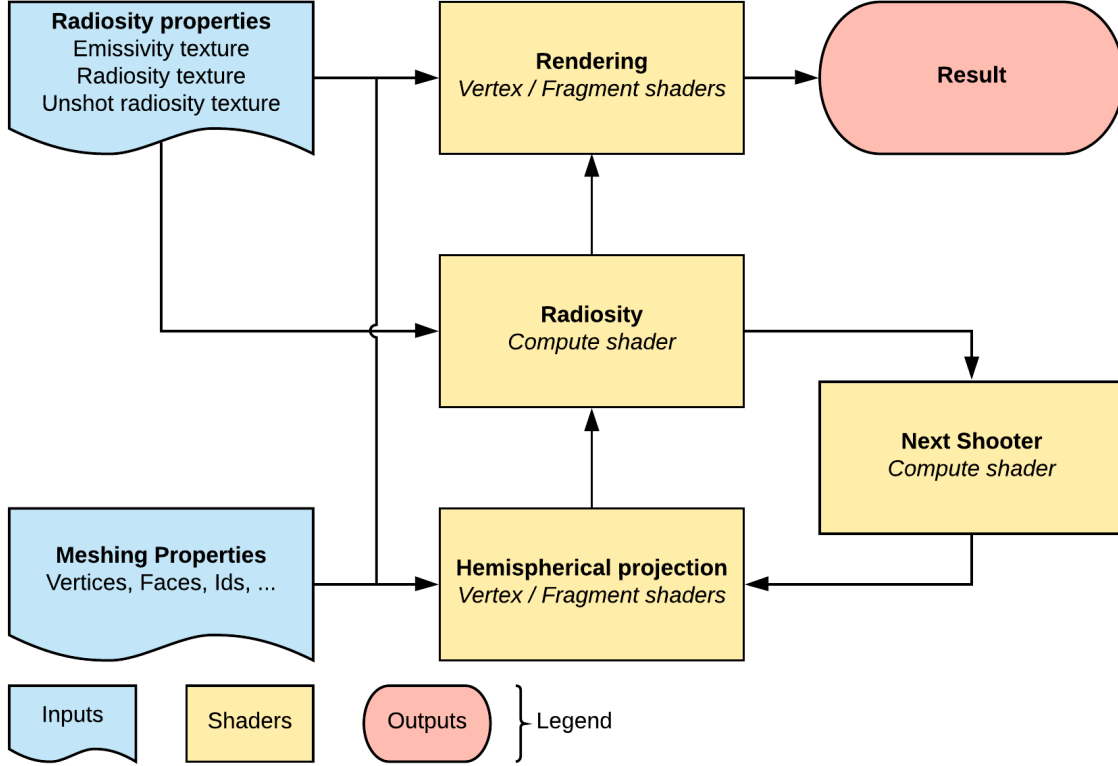
Figure 5.8: Principle of the progressive radiosity GPGPU implementation

physical data, the values are on $\mathbb{R}$ not on $[0;1]$ anymore which makes the algorithm more difficult to implement on the GPU. Due to the fact that all objects are considered as sources, the initialization needs to compute the spectral emittance of all patches:

$$M_{k,\Delta\lambda_i} = \epsilon_{k,\Delta\lambda_i} \int_{\Delta\lambda_i} L^\circ(\lambda, T_k)d\lambda \tag{5.21}$$

Those computed spectral emittances are the initialization of the radiosity and unshot radiosity textures, which corresponds to the first picture of Fig. 5.5.

> **Note**
>
> The initialization step is performed at the beginning, once the 3D model has been loaded. If the meshing is finely refined, the process can be quite long. The arrays are filled on the CPU and then transfered to the GPU textures. To overcome this, one possible way (not yet implemented) is to parallelize the texture filling. In any case, the textures are saved onto OpenEXR textures files to avoid computing those values again. OpenEXR is a widespread format developed by George Lucas' ILM company for saving 32-bits values textures up-to 64 different channels.

Once the textures have been initialized, some parameters such as atmospherics, solar, or camera parameters and described in the next sections can be tuned. Then, the parallelized progressive radiosity algorithm can be started.

5.3.3 Form factors computation

Instead of using the classical hemicube algorithm, the Nusselt's Analog hemisphere has been used to reduce the number of renderings and avoid cubemaps usage. In fact, hemicube rendering needs to render the scene from the five sides of the hemicube. Instead, a non-linear projection is performed and used with the Nusselt equations to derive an approximation of the form factors [79]. A square texture is used in offline rendering to render the view from the current's element of interest. Surrounding elements are then projected onto a unit hemisphere by using a non-linear transformation, analogous to a fisheye lens as shown in Fig. 5.9. Even though a non-linear projection is more costly than a linear one, this extra-cost is manageable over 5 renderings.

A comparative study of form factors computation with stereographic projections is done in [7]. This article points out the fact that using a stereographic projection in the vertex shader accelerates the creation of visibility textures but has many drawbacks. Among those drawbacks, the information of overlapped surfaces is distorted and leads to incorrect shadow boundaries. In fact, since only vertices are affected by the projection, straight edges in the scene are mapped onto straight edges in the projected surface instead of curved segments. This issue becomes critical when the meshes are too coarse. Extreme deformations may lead to incorrect visibility classification and therefore incorrect distribution of energy. This last drawback leads to inconsistent values and increases the algorithm computation time [7] due to the fact that the algorithm may not converge correctly if the visibility is not well computed. Instead, a real non-linear projection is shown to overcome most of those issues and to decrease the rendering time considerably. This algorithm has been developed by [69] and propose actual non-linear projection for coarse meshes. However, this method has not been implemented in the software. First, most of the 3D scenes used can be refined thanks to the important amount of available memory

on recent hardware, reducing the approximation of elliptic edges to straight ones. The drawback is that the meshing has to be refined everywhere and adaptive subdivision is therefore not possible otherwise distortion will lead to wrong refinement. It seems that adaptive refinement would be possible by using a real non-linear projection (including edges) as described in [69].

When loading the 3D mesh, each surface is given a color-id. This unique id is then used to retrieved back corresponding elements on the projection texture.

> ### Note
>
> In the actual implementation a limit of $256^3 - 2$ faces can be used due to the color-id limit (2 colors (black and white) are kept special to check the background for example). This number could be extended by using a hierarchical method or a specific texture containing the ids. In such case, no limit on the number of faces would exist.

The Eq. (5.5) is then used by counting the number of pixels for each element seen on the projection texture. By dividing by the total number of pixels in the hemispherical base, an approximation of the form factors can be obtained. The algorithm is implemented onto the GPU. The OpenGL® rendering pipeline is used for projecting the scene at every iteration from the point of view of the element that needs to distribute its energy. To keep the texture onto the GPU and avoid back and forths exchanges between the CPU and GPU, the counting of the pixels is performed by using compute shaders, memory barriers and atomic counters [159]. Even though memory barriers break the parallelism essence of the method, it enables the process to be kept onto the GPU without any additional memory transfer.

> ### Note
>
> By taking the equations of the form factors and using the Nusselt analog:
>
> $$F_{1\to2} = \frac{1}{A_1} \int_{A_1} \int_{A_2} dF_{d1\to d2} \mathrm{d}A_1 = \frac{1}{A_1} \int_{A_1} F_{d1\to2} \mathrm{d}A_1$$
> $$= \frac{1}{A_1} \int_{A_1} \frac{A_n}{\pi} \mathrm{d}A_1$$
>
> By assuming that the meshing is refined enough and that the surfaces dk are small enough regarding the involved scene distances; the quantity $F_{d1\to2}$ can be considered constant over A_1:
>
> $$F_{1\to2} \approx \frac{A_n}{\pi} \frac{1}{A_1} \int_{A_1} \mathrm{d}A_1 = \frac{A_n}{\pi}$$

The radiosity equation is therefore adapted to the implemented Nusselt hemisphere form factor algorithm. Let N_b be the number of pixels in the hemisphere basis of the projection texture and $\text{el}(i)$ the function that returns the element index based on the pixel color id. Since $A_n = N_n/N_b$

$$J_{k,\Delta\lambda_i} = M_{k,\Delta\lambda_i} + (1 - \epsilon_{k,\Delta\lambda_i})\frac{1}{\pi N_b}\sum_{p=1}^{N_b} J_{\text{el}(p),\Delta\lambda_i} \tag{5.22}$$

Please note that by using the projection, the visibility is automatically computed through a Z-buffering algorithm in the rendering pipeline. Therefore, the summation performed in the previous equation is done only on the visible elements of k. Furthermore, the sum of the form factors is always equals to 1 by design which prevents the algorithm from diverging. In order to validate the implementation of the form factors, a particular unit-test has been implemented in the simulator. Identical and parallel directly opposed rectangles have been taken to compute the form factors and compare to analytical solutions [1, 83, 91]. The model has been created with the 3D modeling software Blender. The unit test is passed with an error smaller than 0.1%
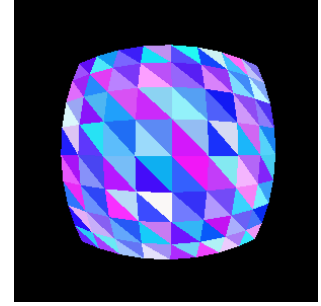
Figure 5.9: Projection image of one of the plane from an element of the other plane

5.3.4 Progressive radiosity implementation

Now that the form factors can be computed, the full algorithm can be implemented and its more detailed pseudo-code is given in Algorithm 10. The distribution of energy is made through a GPGPU approach by using compute shaders. The rendered projection texture is the input of the compute shader which processes each pixel separately on a single grid element. The purpose of this method is to parallelize the `For each` loop in Algorithm 10 and is applicable thanks to the `coherent` memory access of OpenGL®'s `Image Load Store` functions property.

Next element function

As in the standard progressive radiosity algorithm, the next shooting element is the one with the greatest energy. From a practical point of view, it means that the largest unshot radiosity times area element textures must be found. If finding the maximum value in an array through the CPU is a standard thing it is more difficult on a GPU. In fact, looking into a GPU array the maximum value can be done in parallel but may fail due to concurrent writings and readings.

Concurrency: occurs when two asynchronous processes try to write / read to the same location at the "same" time which can lead to memory corruptions and wrong data written or read. Concurrency

---

Algorithm 10: Proposed code for progressive radiosities algorithm on hardware

---

Function `RADIOSITY()`

    Initialisation *radiosity = residuals = emission, stopCriteria = ∞, i = nextElement()*

    While *stopCriteria unsatisfied*

        hemiProjection = nonLinearProjection(*i*);

        For each *pixels $j \in$ hemiProjection*

            ;   `//performed on the compute shader`

            F_{ij} = computeFormFactor(*i,j*);

            $\Delta J = (1 - \epsilon_j) * F_{ij} * residuals_i * \frac{A_i}{A_j}$;

            $radiosity_j = radiosity_j + \Delta B$;

            $residuals_j = residuals_j + \Delta B$;

        $residuals_i = 0$;   `//`*i*` contribution is reset`

        i = nextElement();   `//choose `*i*` such as `$B_i * A_i$` is larger`

---

can be avoided by different methods such as message-passing where the requested are queued.

Many possibilities can be investigated to overcome the concurrency on the GPU. The most efficient one would probably be to interface with CUDA or OpenCL libraries and a specific kernel. One drawback of this method is that the code becomes more complicated and an overload of includes libraries just to find a maximum value. Another possibility is to get the texture back to the CPU and find the maximum on the CPU but the efficiency of this method may vary between hardwares and breaks the GPU-only process. Instead, the chosen solution is to use again atomic counters thanks to the `GL_ARB_shader_atomic_counter_ops` extension which incorporates an atomic `max` operation.

Atomic counter: a specific openGL variable type which can be seen as a limited buffer image variable on which atomic memory operations can be performed on them. Therefore, no memory barriers are needed to synchronize their access: each read/modify or write operation is completed in its entirety before other operations are permitted.

The current implementation uses two atomic counters, one for keeping the maximum value and one for keeping the index of the maximum. Such design leads to a lack of synchronization in-between the counters:

1. `atomicCounterMax` function replaces the value of the max atomic counter if the new value is greater than the oldest one.

2. `atomicCounterExchange` exchanges the old value in the index atomic counter by the new index

Therefore, if the `atomicCounterMax` function always guarantees to store the maximum, the `atomicCounterExchange` does not guarantees that the index actually corresponds to the max found. In fact, even if the queues of the counters are synchronized by themselves, the maximum and exchange queues are not synchronized together, as shown in Fig. 5.10.

The algorithm is different from the progressive refinement on which the maximum of energy is not always chosen but *one of the maximum patch energy* is instead. Concurrency **can** occur when two

Figure 5.10: Concurrency example. At first, the maximum value is 10 at index 0. The concurrency makes the orders in the queues not the same. The maximum value is always guarantee but not its corresponding index

elements are both greater than the current atomic counter which has more probabilities to happen at the beginning of the algorithm (even though the greatest energy is the same on multiple elements) but less and less while the algorithm progresses, as shown in Fig. 5.11. Trying to add synchronization and barriers into this part of the algorithm may lead to a lack of efficiency due to the induced slowdown.

Post-process

Finally, once the radiosity have been computed, the rasterized triangles cannot be directly used for texturing so missing values are linearly interpolated.

5.4 Camera model

Camera parameters such as focal length, position, resolution or even acquisition frequency for example are inputs to the software that are used to visualize the 3D scene through this specific camera from one side and also to compute the infrared image from the other part. For instance, the focal length, position and camera resolution are used for the projection of the scene through the pinhole model for the visualization. In the current version, there is no optical transfer function to simulate advanced effects such as diffraction, opening shape and various aberrations but such feature could be added in the future. However, the camera is also defined by a matrix of detectors that can be customized to take into

Figure 5.11: Energy shot at a given time of the algorithm as a percentage of the first shot

account the acquisition process, from the irradiance at the camera to the digital image. As for example, the quantum CCD/CMOS detectors defined in [101] have been implemented directly into the GPU through shaders for a real-time rendering and visualization. The parameters of the detectors can be customized and visualized in live. This model considers the following conversion process: photons → electrons → voltage → digital levels. Different noises parameters can be adjusted such as shot noise, dark current, gains, analog to digital converter *etc.* Fig. 5.12 shows an example of a camera impulse response.
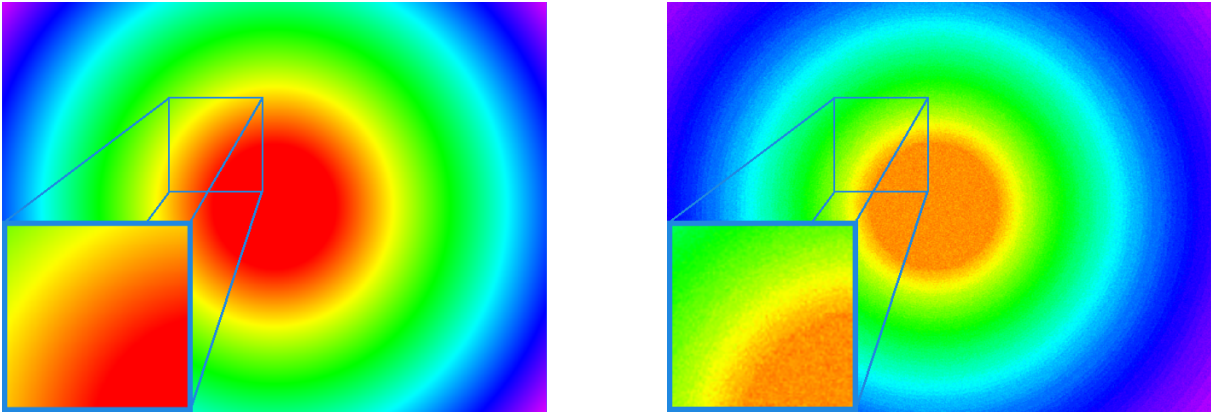
Figure 5.12: Signal received at the camera (left). Generated image by the camera (impulse response), the digitization can be observed as well as the noises impact on the image.

5.5   Solar spectral irradiance

In order to consider the *in-situ* characteristic of the infrared thermography structure monitoring, environmental models have been implemented to simulate the sun contribution, in particular in the NIR band. Such simulation needs the knowledge of the location at a given time to get the solar position

in the sky and an atmospheric model to take into account the different absorption, reflection and diffusion effects of the extraterrestrial radiation through the atmosphere. Many radiative transfer codes have been developed through years (a Wikipedia page with a non exhaustive list even exists [12]) for various applications such as climate models or solar photo-voltaic systems efficiency optimization for example. In this thesis, the concern was about finding a solar spectral model that could give the solar spectral irradiance at ground and fully documented to be integrated into the software. Those requirements were met by the SMARTS2 (Simple Model for the Atmospheric Radiative Transfer of Sunshine) algorithm described in [82]. This model is derived from SPCTRAL2 and SMARTS1, improved by the comparison and fitting of MODTRAN model for adjusting look-up tables coefficient values [181]. By using look-up tables, the SMARTS2 algorithm provides a fast way to simulate the solar spectral irradiance at ground by considering various atmospheric factors.

5.5.1 Sun position

The sun position can be derived thanks to the knowledge of the longitude (l), latitude (φ), date and time considered [180]. The declination angle δ that represents the angular position of the sun at solar midday from the equatorial plan can be written, from [43]:

$$\delta = 23.45 \sin\left(\frac{284 + n}{365} \times 360\right) \tag{5.23}$$

Where n is the nth day of the year. Solar hour can be expressed as a function of the legal hour that depends on the time zone UTC+X by:

$$h_{sol} = \frac{4}{60} \times (15 \times X - l) + \frac{E}{60} + h_{leg} \tag{5.24}$$

where the minute correction E that comes from the variation of the Earth self-rotating speed is given by [172] (see footnote in [180]).

Therefore, the zenith angle is obtained from [61]:

$$\cos(\theta_z) = \sin(\delta)\sin(\varphi) + \cos(\delta)\cos(\varphi)\cos(\omega) \tag{5.25}$$

Finally, the azimuth angle, from the same reference:

$$\gamma_s = \text{sign}(\omega) \left| \arccos\left(\frac{\cos(\theta_z)\sin(\varphi) - sin(\delta)}{\sin(\theta_z)\cos(\varphi)}\right) \right| \tag{5.26}$$

An example of the change of the sun position over the sky vault in the software is shown in Fig. 5.13.
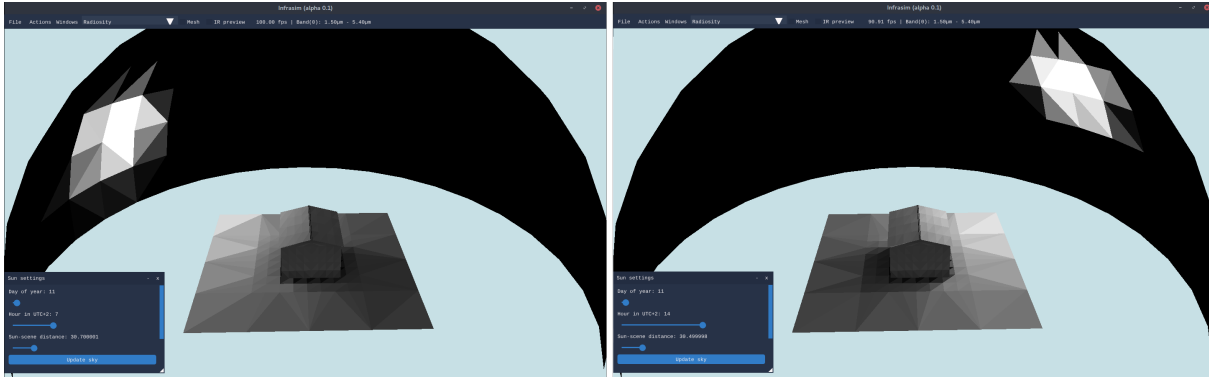
Figure 5.13: Sun position over the sky vault for a 3D wood house scene

5.5.2   Atmosphere

The atmosphere of Earth is composed of multiple gases layers. Depending on the composition of those layers, the solar radiation received from the sun and that goes through the atmosphere will be scattered, absorbed, polarized and transmitted before reaching the ground. Furthermore, the atmosphere will emit itself some radiation, particularly in the infrared spectrum due to the involved temperatures. Many models exist in the literature, from advanced and complex ones using finite elements and/or ray tracing solutions to simpler ones using fitted formulas from experiments.

Now that the zenith angle is known based on the day of the year, time and location the solar spectral irradiance can be derived from SMARTS2 algorithm. The SMARTS2 algorithm gives direct beam and diffuse beam radiations. For more details about how those quantities are derived, please refer to the original article [82]. The SMARTS2 algorithm uses 10 reference atmospheres included 6 used in MODTRAN. Those reference atmospheres take into account

Figure 5.14: Atmospheric model parameters in the software

different vertical profiles of air temperature, pressure, relative humidity, pathlengths of O_2, H_2O, *etc.* based on altitude variation. Also, 4 aerosol models (urban, rural, maritime and tropospheric) can be chosen. Fig. 5.14 shows the configuration window in the software with the possibility to plot the resulting transmittance and solar spectrum.

Extraterrestrial solar spectrum

SMARTS2 uses a reference extraterrestrial solar spectrum with $1nm$ resolution in $[280nm; 1700nm]$ range and $5nm$ in $[1705nm; 4000nm]$ range. This spectrum was derived from previous studies and adjusted to satellite data. Since this spectrum is interpolated in the current implementation to fit the user's requested resolution, this input spectrum could be changed or adapted to more recent data; if needed.

Direct beam radiation

The direct beam radiation by a surface normal to the sun is obtained by using the following formula:

$$E_{bn,\lambda} = E_{on,\lambda}\tau_{R,\lambda}\tau_{o,\lambda}\tau_{n,\lambda}\tau_{g,\lambda}\tau_{w,\lambda}\tau_{a,\lambda} \tag{5.27}$$

with

- $E_{on,\lambda}$ the extraterrestrial irradiance reaching the Earth,
- $\tau_{R,\lambda}$ transmittance due to rayleigh scattering,
- $\tau_{o,\lambda}$ transmittance due to ozone absorption,
- $\tau_{n,\lambda}$ transmittance due to nitrogen dioxide absorption,
- $\tau_{g,\lambda}$ transmittance due to mixed gases absorption (mainly O_2 and CO_2),
- $\tau_{w,\lambda}$ transmittance due to water vapor absorption,
- $\tau_{a,\lambda}$ transmittance due to aerosol extinction.

Some examples are provided in the following figures. Fig. 5.15a shows the impact of the zenith angle to the received direct irradiance component whereas Fig. 5.15b shows the impact of the visibility.
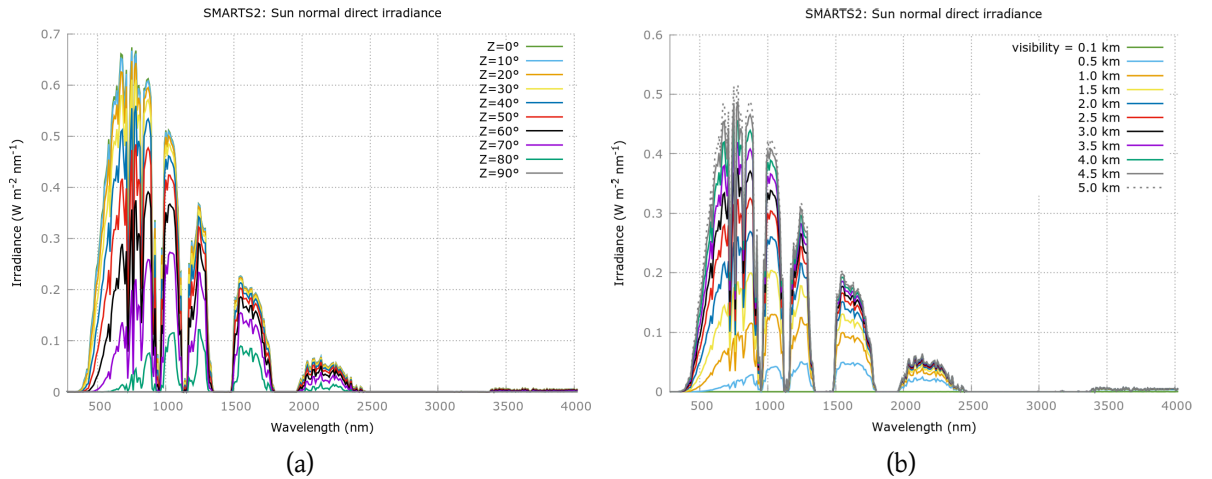
Figure 5.15: Spectrum of the solar irradiance received by a surface normal to the sun for different visibility and altitude values

An example of the effect of zenith angle over the transmittance is shown in Fig. 5.16.
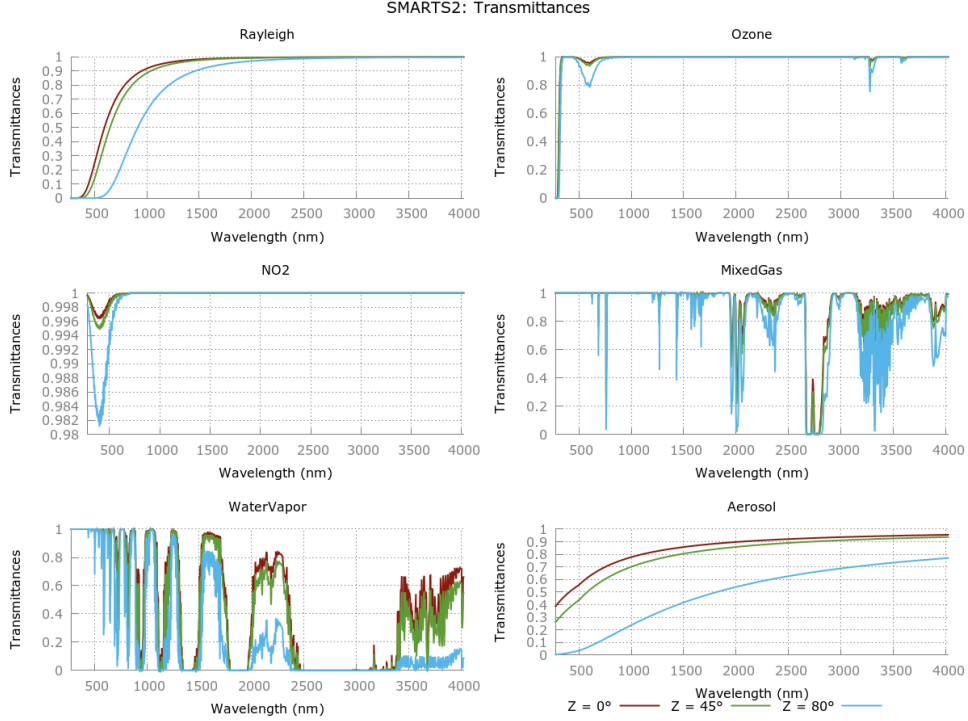
Figure 5.16: Transmittance values for different zenith angles

Diffuse beam radiation

The diffuse component of the sun spectral radiation is approximated by using the same transmittance functions used in the direct component in order to simplify the computation. A physical justification can be that the photons that are not directly transmitted are scattered in all directions for which a roughly predictable fraction will be directed downwards [82]. The diffuse irradiance is then given by:

$$E_{d,\lambda} = E_{d,R,\lambda} + E_{a,R,\lambda} + E_{b,R,\lambda} \tag{5.28}$$

with

- $E_{d,R,\lambda}$ due to Rayleigh scattering,
- $E_{a,R,\lambda}$ due to aerosol scattering,
- $E_{b,R,\lambda}$ due to backscattering (interreflections between the ground and the atmosphere).

Implementation and units tests

The SMARTS2 algorithm and its corresponding look-up tables have been implemented in C++ providing a model to simulate the radiation from the sun. The different parameters can be customized

in a specific window as shown in Fig. 5.14. The sun and sky are considered as a hemisphere over the scene and contribute to the radiation through the standard radiosity algorithm for which the patches only emit. The diffuse component is spread over the entire sphere whereas the direct one affects the patches based on the zenith angle. Currently, the circumsolar radiation part of the SMARTS2 algorithm is not implemented yet.

Again, units tests have been implemented to check the reliability between the expected article results and the implemented ones. The results in the article have been reproduced (see Fig. 5.17) and units tests for the different look-up tables interpolations algorithms performed.
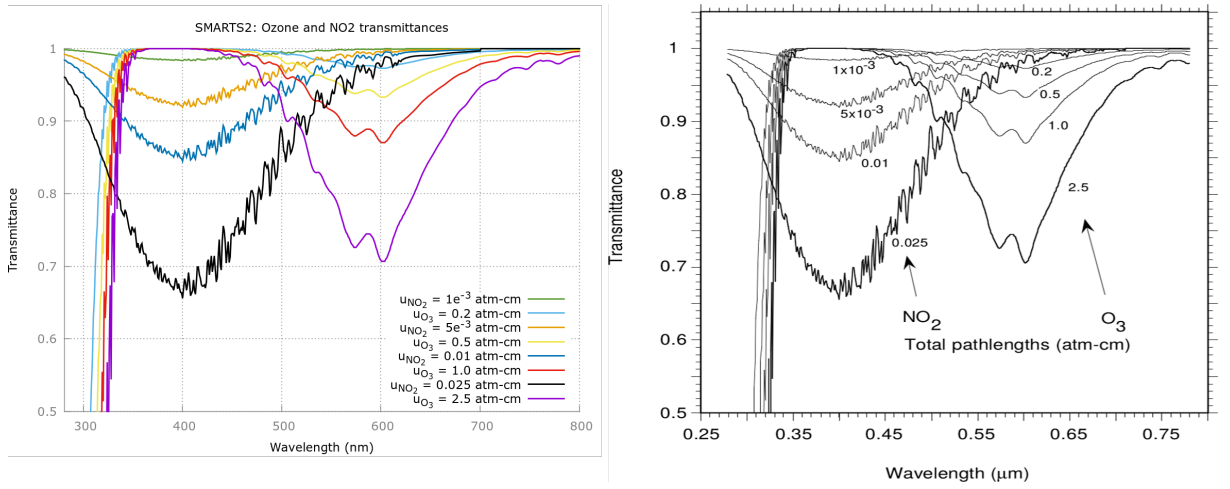
Figure 5.17: Effect of the pathlengths of ozone and NO_2 to their transmittance and comparison with the original figure.

5.6   Results and examples

The different algorithms and models used to simulate the radiative exchanges on a 3D scene have been exposed and their implementation commented in the previous section. In this section, results are provided and some examples of usage are shown to illustrate the user-friendly part of the developed software and its possible applications.

5.6.1   Graphical User Interface (GUI)

Once the radiosity algorithm has been computed - or during the computation -, the user can switch between multiple rendering shaders for displaying different available results. The radiosites values on every surface on the scene can therefore be observed and more information can be obtained by clicking on a specific surface. Another rendering shader enables the visualization of the energy received at camera's sensor that simulates the irradiance at the camera. This is achieved through Bouguer's law

Eq. (2.24) and a model of the detectors of the camera as explained in the next section.

$$I_{k,\Delta\lambda_i} = \frac{J_{k,\Delta\lambda_i}\cos(\theta)}{d^2} \tag{5.29}$$

Where I is the irradiance received at the camera, d is the distance and θ the angle between the pixel of the camera and the observed surface of radiosity B.

5.6.2   YAML files configuration

The simulation settings can be customized by a YAML (YAML Ain't Markup Language) configuration file. The YAML syntax [178] allows readable and easy to write syntax. A user can therefore customize the simulation settings and such configuration file is a good way to save and restore different simulation parameters. Variables can be defined locally on the YAML file, according to the `YAML 1.2` specification. Moreover, a specific feature has been added to the YAML parser of the software so that it understands formulas through the ExprTk library [24] and the tag `!!formula`. An example and more details about those configuration files are given in Appendix B

5.6.3   Python interpreter and module

Even if the graphical user interface enables interactions with the software, it may be complicated to batch process multiple simulations. Moreover, extending the software or writing `C++` code in it to fits the expectations may be difficult for most users. That is why a Python interpreter (`CPython`) has been included and a specific Python module has been developed so that the user can interface with the software from Python. Python is a widespread high-level programming language among scientist with a strong and dynamic typing. `CPython` offers a `C` implementation of Python. Interacting with the software is made easier thanks to the Python interpreter. The user has access to the 3D scene and the radiosity renderer programmatically, which can be useful to perform multiple simulations with different parameters or even animate the 3D scene (see Fig. 5.18).

```python
import time
for i in range(0,200):                              # For 200 iterations
    infrashell.rotate("floor_Plane", (0,4,0), True) # Rotate the object
    infrashell.draw()                               # Update the screen
    time.sleep(0.1)                                 # Pause to see the update
```

Figure 5.18: Animation of an object in the 3D scene through Python: an object on the scene is rotated of 4 degrees and the scene is displayed on screen at each iteration.

5.6.4 Static results

Fig. 5.19 shows an example of radiosity map that can be obtained in a static configuration with a building (near infrared band to view the effect of the sun). The sun contribution can be observed among with the created shadow due to this illumination. The meshing impact on the final resolution and the interpolation quality is also visible, and very documented in the literature [40]. The Python script used to generate the simulation is given in Appendix C
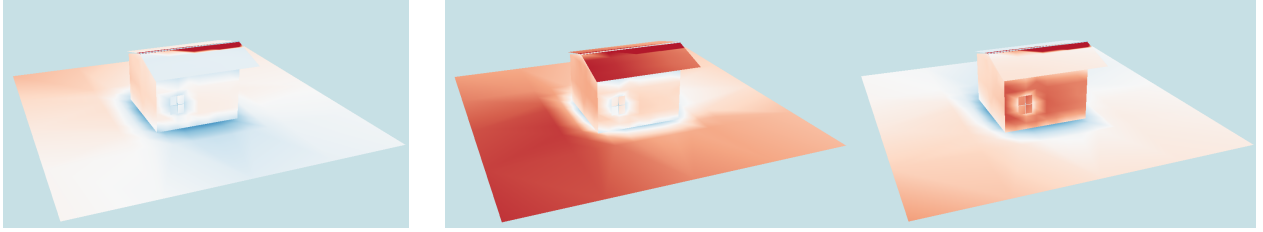
Figure 5.19: Radiosity map of a building for a given day, at different time (from left to right: 6AM, 12PM and 4PM)

5.6.5 Dynamic results

The simulation enables to have a first sense of the infrared measurement on site, which can be viewed as a decision making tool for infrared measurements systems sizing. Simulating data can help to choose the right measurement system, prior to the exploitation. In the following use-case, this tool has been used in this philosophy to study the feasibility of measuring train's hot boxes [179] through two different infrared cameras. The first camera used was an uncooled FLIR SC655 at 100Hz in the spectral band [7.5μm13μm] (LWIR) and the second one a cooled FLIR XC6540 at 250Hz in the [1.4μm5.4μm] (SWIR-MWIR) band. A 3D model of a train boogie has been used as shown in Fig. 5.20
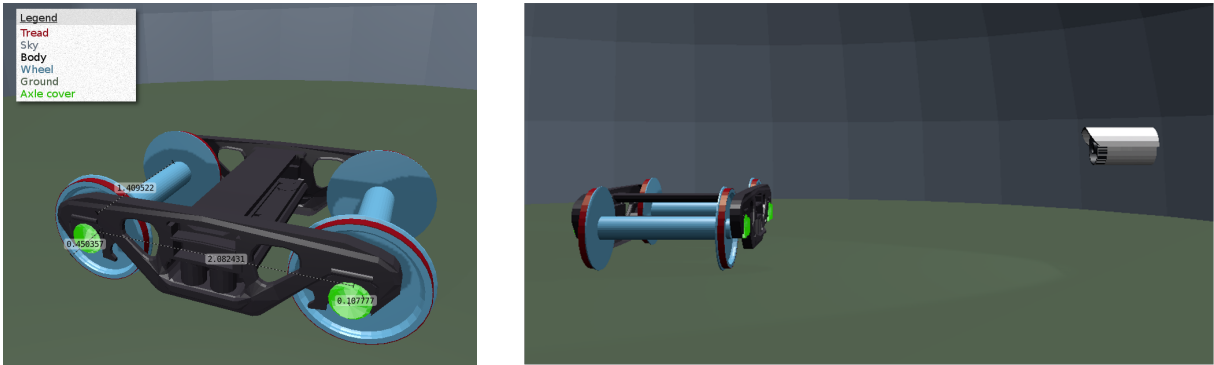
Figure 5.20: 3D model of the boogie in its scene environment

Then, based on the radiosity rendering equation, the irradiance received at the camera has been computed by using the Bouguer's law. An example of the boogie in the software is shown Fig. 5.21.
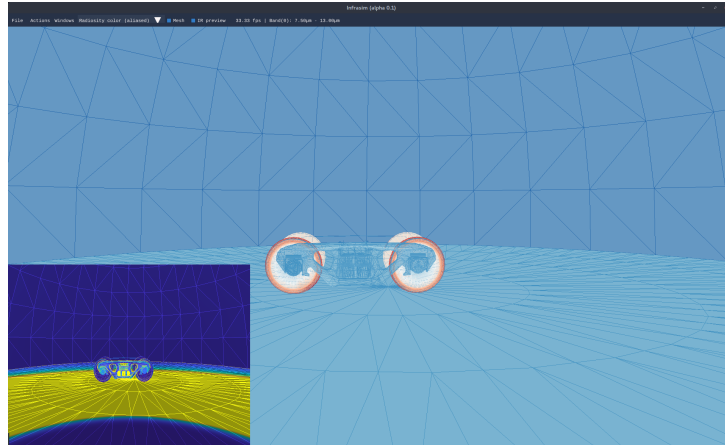
Figure 5.21: Boogie and environment in the simulation software. The preview of the rendered infrared image is done in real-time at the bottom left corner.

Then, experimental measurements have been performed in order to make a visual comparison between experiment and simulation. As shown in Fig. 5.22.
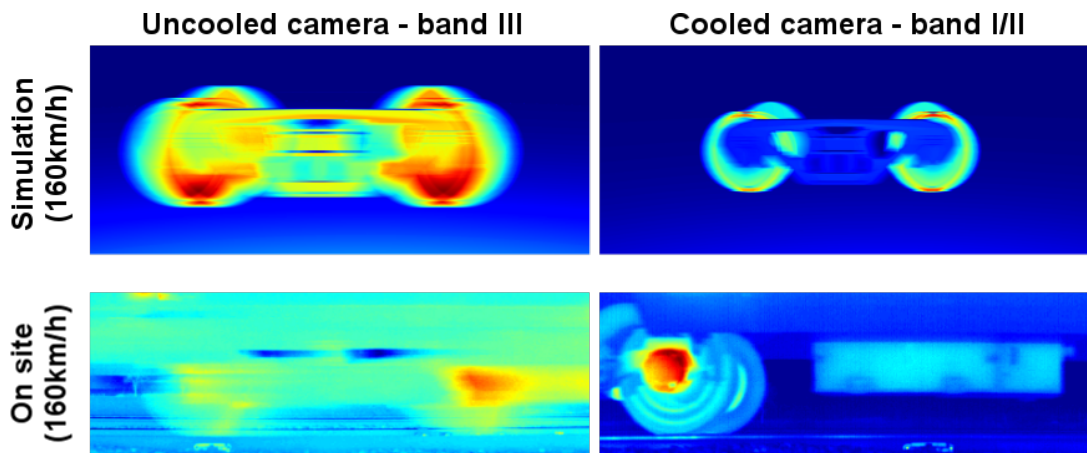
Figure 5.22: Comparison of simulation data with experimental ones. The motion blur is well represented in the simulation. The temperature levels however are not well represented due to a lack of information about the initial temperatures values of the boogie.

5.6.6 Comparison with literature

In order to validate the software, a comparison with a common literature sample called the Cornell Box has been made. The Cornell Box is a famous model which aims at determining the accuracy of rendering software. The model with the data found in [45] has been used. In order to simplify the image creation, the rendering has been made on 3 bands: [400nm; 495nm], [495nm; 590nm], [590nm; 700nm] assuming that those bands correspond approximately to blue, green and red, respectively. Therefore,

the reflectivity spectrum given by the original Cornell Box has been averaged on this bands, leading to the dashed spectrum in Fig. 5.23.
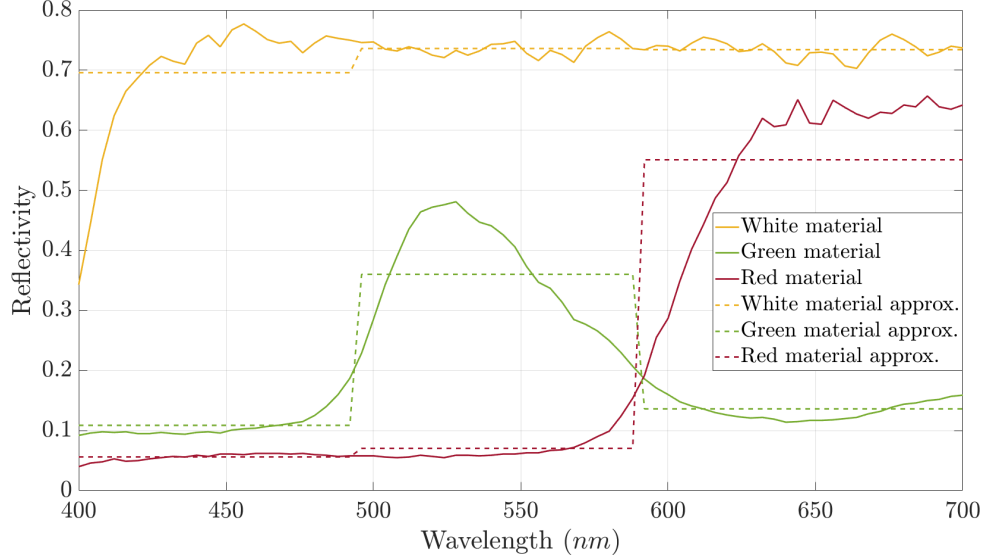
Figure 5.23: Original and approximated reflectivity spectrum used for the validation

The following results are shown below, for two different light temperature, one light white and another one that goes onto the red. In Fig. 5.24, the first top left zoom shows the diffusion that occurs between the red wall to the tall box, as expected. However, the classical issues of the radiosity method can also be found and are well addressed in the literature [163]. An example is the wrong shadows, on the second zoomed box at the lower right corner of Fig. 5.24. Such issue is due to the too coarse mesh in this region and the aliasing algorithm. It can be prevented by modifying the aliasing method and/or modify the mesh so that it is more refined where the gradient is important [86, 184, 136].
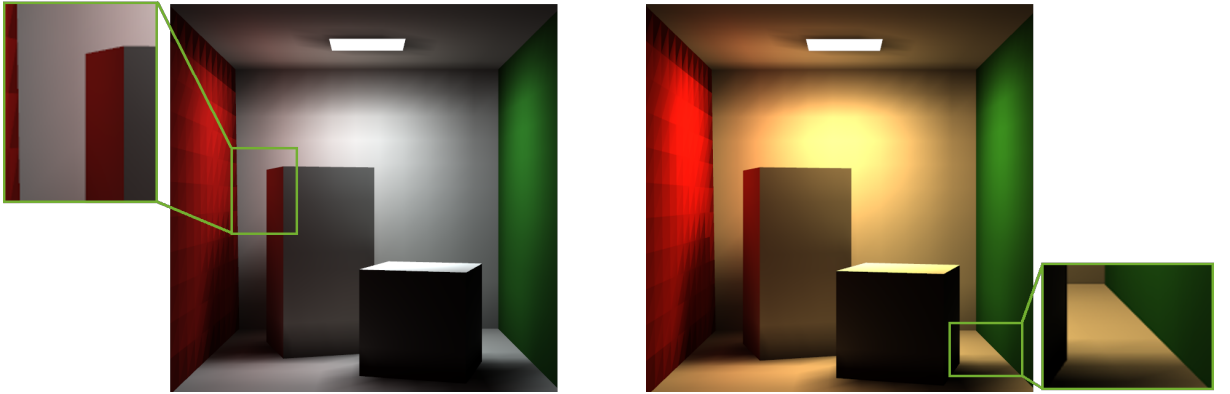
Figure 5.24: Cornell's box result for two different lights. Such result is computed on a professional laptop - M1000M Nvidia graphic card, Intel i7-6820HQ CPU @ 2.70GHz. The computation time is approximately 60 seconds with a progressive rendering and aliasing on screen every 50 iterations.

5.7   Synthesis

An infrared imaging software that simulates radiative exchanges of 3D scenes statics and dynamics have been produced. The implementation of the progressive radiosity method on GPU by using OpenGL® enables a parallelized rendering and visualization at the same time. Thanks to detectors and solar irradiance spectral data models from the literature, advanced *in-situ* multi-band infrared thermography simulations are possible. Settings different parameters onto the models through multiple simulations helps in setting infrared measurements systems before the exploitation. Moreover, thanks to the Python module, batch processing and advanced dynamic use-cases can be automatically rendered. However, improvements can be made in the simulation tool. In fact, the parallelized radiosity method could be improved by using an adaptive meshing that makes some refinement based on the energy gradient [79] and therefore modify the projection algorithm [7, 69]. Moreover, an overshooting method is not performed but could be incorporated to speed-up the convergence [77]. Also, to overcome the orthotropy, a BRDF (Bidirectional reflectance distribution function) could be used [163] and specular reflection as well as transmission could be incorporated as in [187].

The developed software provides a way to study different parameters for *in-situ* infrared thermography measurements. Its parallel implementation on graphics hardware uses advances from the image synthesis and computer science fields to enables fast interaction and visualization of *in-situ* 3D spectral scene which has not been done previously according to our knowledge. Moreover, the use of solar and camera's detectors models from the literature and parallelized through the graphics hardware offer a tool to simulate IR *in-situ*ăacquisition from the radiative exchanges to the generated image in digitial levels.

A flowchart that summarizes the developed software is given in Fig. 5.25. The software will be used in the following to simulate data for the evaluation of different emissivity and temperature retrieval methods.
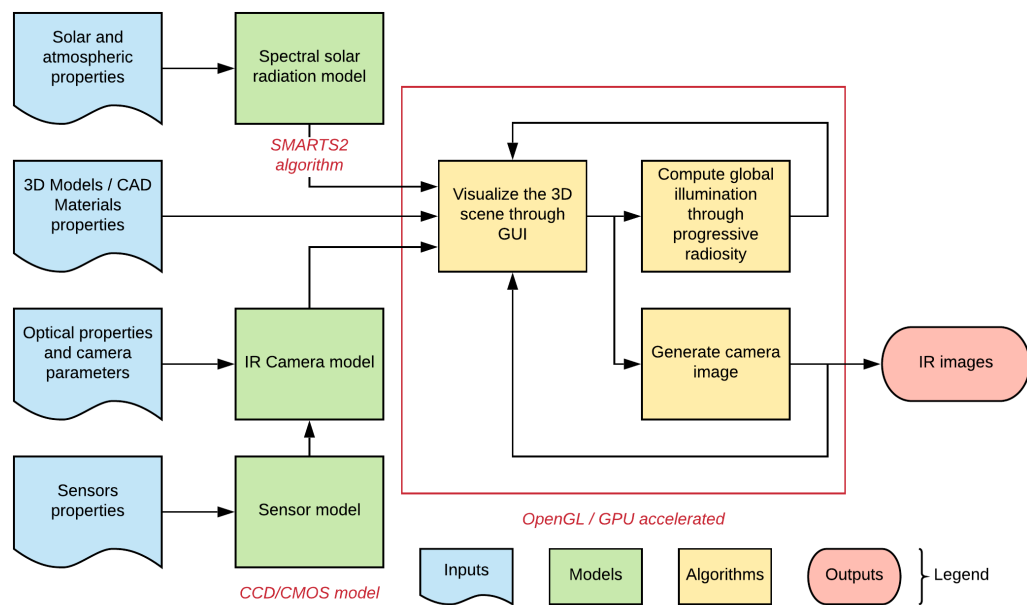
Figure 5.25: Flowchart overview of the developed software showing the different steps of the numerical simulation

Proposed and studied methods for the simultaneous estimation of temperature and emissivity

I N the previous chapter, different emissivity and temperature estimation methods were presented. Those methods are either restricted to a range of applications and materials or computationally intensive. In particular, no specific method for *in-situ* IRT structure monitoring was introduced. In this chapter, a presentation of the developed methods for the simultaneous estimation of temperature and emissivity through IRT is made.

Firstly, the state-space representation of the problem is derived from the equations presented in Chapter 3. In order to track the dynamic component of the measured signal (*i.e.* the temperature under our assumptions), a Kalman filter has been implemented. However, the lack of knowledge about the emissivity and the non-linear characteristics of the model lead us to extend this Kalman filter to a particle filter. Such approach exploits the temporal and multi-bands aspects of the signal but does not take into account the spatial one. Therefore, a Kriging step is finally added to the particle filter which serves two purposes. The first purpose is to integrate into the model a spatial description for the measured signal. The second purpose is to use the Kriging as an interpolator and thus avoid having the particle filter to run on all the pixels of the image but only on a selection of the pixels. This last step reduces the required computation needs.

Results of the different implementations are presented and discussed around a theoretical test case simulation scenario.

6.1   Linear time varying (LTV) systems in state-space representation

The methods presented in the following make use of a state-space representation in their formulation of the problem. Let define the state-space model of a linear system with $i \in \mathbb{N}$ inputs, $o \in \mathbb{N}$ outputs and $n \in \mathbb{N}$ so called state variables. A general state-space representation is given in the fol-

lowing equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t)$$
$$\boldsymbol{y}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \boldsymbol{D}(t)\boldsymbol{u}(t)$$

(6.1)

where:

- $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector,
- $\boldsymbol{y} \in \mathbb{R}^o$ is the output vector,
- $\boldsymbol{u} \in \mathbb{R}^i$ is the control vector,
- $\boldsymbol{A} \in M_n(\mathbb{R})$ is the state matrix,
- $\boldsymbol{B} \in M_{n,i}(\mathbb{R})$ is the control matrix,
- $\boldsymbol{C} \in M_{o,n}(\mathbb{R})$ is the observation matrix,
- $\boldsymbol{D} \in M_{o,i}(\mathbb{R})$ is the feedthrough matrix.

with $M_n(\mathbb{R})$ the set of matrices with coefficients in $\mathbb{R}$. The first equation is usually called the dynamic equation and the second one the measurement equation.

A discrete version of the previous state-space model can be derived from Eq. (6.1):

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k$$
$$\boldsymbol{y}_k = \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{D}_k\boldsymbol{u}_k$$

(6.2)

Based on the linearization of the temperature-emissivity problem for IRT (Eq. (3.20)), a state-space representation of the emissivity and temperature estimation problem for IRT can be derived. By assuming that the emissivity is constant through time and using the notation defined in Chapter 3, the measurement equation can be defined as:

$$\boldsymbol{y}_k = \begin{pmatrix} 1 & \Delta\lambda_1 & \Delta\lambda_1{}^2 & \ldots & \Delta\lambda_1{}^m & -\frac{C_2}{\Delta\lambda_1} \\ 1 & \Delta\lambda_2 & \Delta\lambda_2{}^2 & \ldots & \Delta\lambda_2{}^m & -\frac{C_2}{\Delta\lambda_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta\lambda_N & \Delta\lambda_N{}^2 & \ldots & \Delta\lambda_N{}^m & -\frac{C_2}{\Delta\lambda_N} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \\ \frac{1}{T_k'} \end{pmatrix}$$

(6.3)

for a given discrete time k. The associated state-space representation can be defined as:

$$x_k = \frac{1}{T_k} \quad \boldsymbol{y}_k = (ln\,(\gamma_{\Delta\lambda_i}))_{1 \le i \le N} \quad \boldsymbol{u}_k = \boldsymbol{u} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} \quad \boldsymbol{B}_k = 0$$

$$\boldsymbol{C}_k = \boldsymbol{C} = \begin{pmatrix} -\frac{C_2}{\Delta\lambda_1} \\ -\frac{C_2}{\Delta\lambda_2} \\ \vdots \\ -\frac{C_2}{\Delta\lambda_N} \end{pmatrix} \quad \boldsymbol{D}_k = \boldsymbol{D} = \begin{pmatrix} 1 & \Delta\lambda_1 & \Delta\lambda_1{}^2 & \dots & \Delta\lambda_1{}^m \\ 1 & \Delta\lambda_2 & \Delta\lambda_2{}^2 & \dots & \Delta\lambda_2{}^m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta\lambda_N & \Delta\lambda_N{}^2 & \dots & \Delta\lambda_N{}^m \end{pmatrix} \tag{6.4}$$

The state matrix A_k that represents the dynamic evolution of the state vector is undefined, yet. In fact, the evolution of the temperature remains difficult to predict. In the following, different dynamic equations will be considered for characterizing the evolution of the temperature. Even if some models can be found in the literature, a perfect dynamic clear sky ground's temperature evolution may not be suitable for having a good A_k estimate. On the other hand, choosing a complex model will lead to add more information onto the system that may not be always available. For the Kalman filter, the dynamic model will be taken as $A_k = 1$, leading to a random walk model. In the particle filtering based methods, the A_k will be considered as a parameter to be estimated as well as the emissivity. However, even if it has not been implemented in this work, an auto-regressive based dynamic model is possible. In fact, by using daily temperature measurements, an auto-regressive model can be fitted and then applied to the dynamic equation. Such model is proposed in Appendix D. Finally, such notations lead to this simplified state-space representation:

$$\begin{aligned} x_{k+1} &= A_k x_k \\ \boldsymbol{y}_k &= \boldsymbol{C} x_k + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{6.5}$$

The $\boldsymbol{C}$ matrix can therefore be defined as in Eq. (3.32) for polynomial approximation or in Eq. (3.41) for spline approximation.

This state-space representation hides many assumptions. The first one is that only the temperature and its evolution are supposed to vary through time (or the emissivity is assumed to vary slowly regarding the variation of temperature). The emissivity and the temperature are assumed to be independent. Those assumptions will be discussed later on. Please note that the presented model will evolve through this chapter to fit the developed methods. In the following, Bayesian filters that will be used to develop the estimation methods are presented.

6.2 Optimal Bayes filter

6.2.1 Prediction

The idea is to estimate the next state of the system given past measurements (next and past are related to current time). Let note $\boldsymbol{\mathcal{Y}}_t$ the measurement up to time t, including the past $t-1$ measurements. Consider the state-space system $(\boldsymbol{x}_t, \boldsymbol{y}_t)$. The objective is to find the prediction $p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})$ given the belief $p(\boldsymbol{x}_{t-1}|\boldsymbol{\mathcal{Y}}_{t-1})$. By using the law of total probability, we have:

$$
\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1}) &= \int p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1} \\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\mathcal{Y}}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1} \\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1}
\end{aligned}
\tag{6.6}
$$

The last line of the equation is obtained by assuming that the phenomena are independent: state $\boldsymbol{x}_t$ depends only on state $\boldsymbol{x}_{t-1}$. Such assumption means that only the immediate past matters: $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots, \boldsymbol{x}_0) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

6.2.2 Update

The update step involves computing $p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t)$ given a prediction $p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})$ and a measurement $\boldsymbol{y}_t$. By applying Bayes rule:

$$
\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t) &= \frac{p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\mathcal{Y}}_{t-1})}{p(\boldsymbol{y}_t|\boldsymbol{\mathcal{Y}}_{t-1})} \\
&= \frac{p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{\mathcal{Y}}_{t-1})}
\end{aligned}
\tag{6.7}
$$

Again, the last line is obtained through independence assumptions. That is to say, the observations only depends on the current state: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_0, \boldsymbol{\mathcal{Y}}_{t-1}) = p(\boldsymbol{y}_t|\boldsymbol{x}_t)$. By applying Chapman-Kolmogorov equations, one can get

$$
p(\boldsymbol{y}_t|\boldsymbol{\mathcal{Y}}_{t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_t
\tag{6.8}
$$

which appears to be a normalization factor in Eq. (6.7).

6.2.3 Synthesis

Assuming that $p(\boldsymbol{x}_0)$ is known, the recursive computation of $p(\boldsymbol{x}_t, \boldsymbol{\mathcal{Y}}_t)$ is based on the optimal filter Eq. (6.6) and Eq. (6.7):

$$\begin{cases} p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1}) = \displaystyle\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1} & \text{prediction} \\[2mm] p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t) = \dfrac{p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_{t-1})\mathrm{d}\boldsymbol{x}_t} & \text{update} \end{cases} \tag{6.9}$$

To conclude, the Bayesian filter is based on two main steps:

- Make prediction based upon previous belief
- Update belief based upon Measurement

Three algorithms (Kalman filter, particle filter and interacting Kalman filter) based on those equations are now introduced and then written for the particular case of the conjoint estimation of temperature and emissivity.

6.3 Kalman Filter

The Kalman Filter is an infinite impulse response filter that estimates the current state of a linear system based on the previous one and the current measurements. The filter can therefore track linear dynamical models in Gaussian noise. This filter involves two main steps, the prediction and the correction based on the observations.

6.3.1 Equations

White noises are now added to the dynamic and measurement equations and the state-space model is considered as a stochastic process with stochastic variables. The purpose of the Kalman filter is to reconstruct the state variable through time by taking into account this state-space model when the system to observe is assumed to be linear. Initial conditions are assumed to be relatively well-known in order to allow a reliable linearization.

The estimate of the state at time k is denoted by $\hat{\boldsymbol{x}}_{k|k}$ and $\hat{\boldsymbol{x}}_{k|k-1}$ represents the estimate of the system's state at time step k before k-th measurement. Let $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ be random independent Gaussian noises white in time (*i.e.* $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are independent one another for all $i \neq j$, same for $\boldsymbol{w}_k$). The covariance matrices of $\hat{\boldsymbol{x}}_{k|k}$, $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are respectively denoted by $\Sigma_{k|k}$, Σ_v and Σ_w.

> **Note**
>
> Usually normal distributions are taken as *a priori*. Apart from the fact that this distribution is widely observed in nature, such assumptions comes mainly from its mathematical properties:
>
> - Stable with finite variance
>
> $$\mathcal{N}\left(x, \Sigma_x\right) + \mathcal{N}\left(y, \Sigma_y\right) \sim \mathcal{N}\left(x + y, \Sigma_x + \Sigma_y\right)$$
>
> - Convenient to generate isotropic search points
> - Maximum entropy relative to all probability distributions covering $\mathbb{R}$ but having finite mean and variance [46].

Prediction

Given the measurements $(\boldsymbol{y}_i)_{0 \leq i < k}$, the expectation $\hat{\boldsymbol{x}}_{k|k-1}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{k|k-1}$ can be written:

$$
\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1} &= \boldsymbol{A}_k \hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_k \boldsymbol{u}_k \\
\boldsymbol{\Sigma}_{k|k-1} &= \boldsymbol{A}_k \boldsymbol{\Sigma}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{\Sigma}_v
\end{aligned}
\tag{6.10}
$$

Correction

The Kalman filter alternates two stages: the prediction and the correction of the prediction based on input measurements at step k: $\boldsymbol{y}_k$. The expectation $\hat{\boldsymbol{x}}_{k|k}$ and covariance matrix $\boldsymbol{\Sigma}_{k|k}$ are given by using the equations of the best unbiased linear estimator.

$$
\begin{aligned}
\tilde{\boldsymbol{y}}_k &= \boldsymbol{y}_k - C_k \hat{\boldsymbol{x}}_{k|k-1} - D_k \boldsymbol{u}_k && \text{Innovation} \\
\boldsymbol{S}_k &= \boldsymbol{C}_k \boldsymbol{\Sigma}_{k|k-1} \boldsymbol{C}_k^T + \boldsymbol{\Sigma}_w && \text{Covariance of the innovation} \\
\boldsymbol{K}_k &= \boldsymbol{\Sigma}_{k|k-1} \boldsymbol{C}_k^T \boldsymbol{S}_k^{-1} && \text{Kalman gain} \\
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k && \text{Corrected estimation} \\
\boldsymbol{\Sigma}_{k|k} &= \boldsymbol{\Sigma}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{C}_k \boldsymbol{\Sigma}_{k|k-1} && \text{Corrected covariance}
\end{aligned}
\tag{6.11}
$$

The Kalman filter aims at correcting the vector's state based on measurements. By taking Eq. (6.9):

- The prediction is obtained by using the previous corrected estimate $p(\boldsymbol{x}_{t-1}|\mathcal{Y}_{t-1})$ and the dynamic model linking the previous state to the next one in time: $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.
- The correction at step t is obtained by using the prediction and an observation model $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$.

As a consequence, on a Kalman filter, the Gaussian is propagated through time. The mean and standard deviation associated to the dynamic noisy model can be predicted and corrected. The Kalman filter is

therefore a feedback system on which the state vector expectation and covariance matrices are updated as new measurements are made. The Kalman filter gives an estimation of x_k but also an appreciation of the result through the estimation of the covariance matrix Σ_k.

6.3.2 Test case

In order to test this method, emissivity profiles have been taken from the database provided in [57]. The emissivity profile is decimated to get a smaller number of samples. A standard temperature 1st-order time evolution has been considered (see Fig. 6.1).
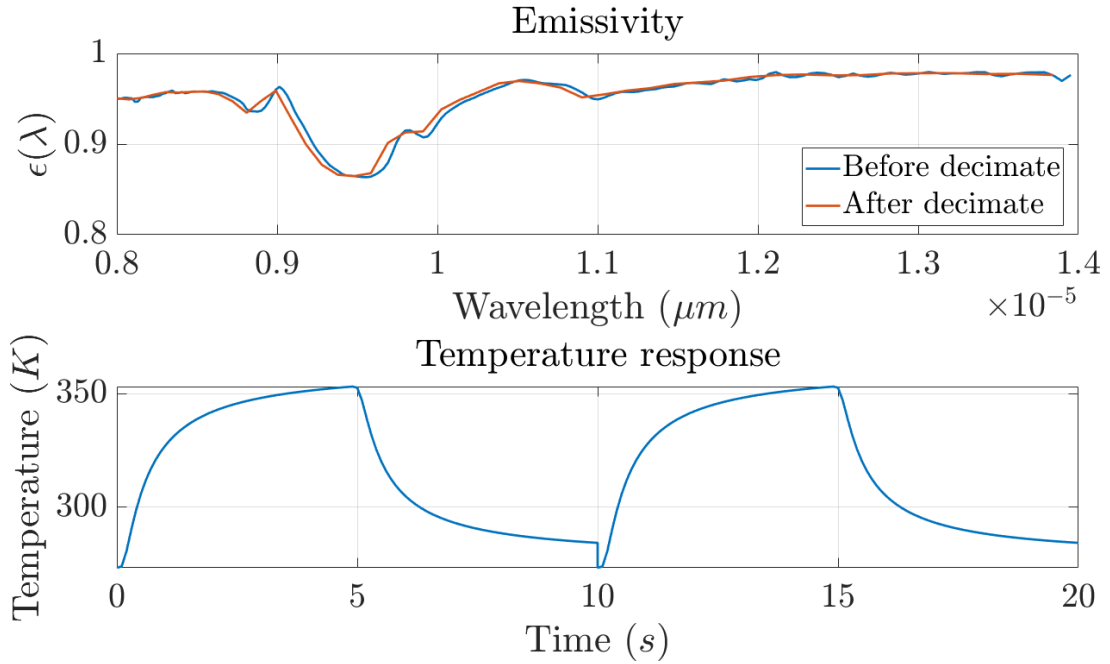
Figure 6.1: Temperature and emissivity profiles

Firstly, the model defined in Eq. (3.32), by using a 6-order Tchebychev polynomials basis has been taken. The results are presented in Fig. 6.2. One can see that the polynomial does not fit the emissivity profile. This is mainly due to a local minimum that is found in the process at the beginning and that will not change through the Kalman evolution. However, the temperature evolution does follow the original time profile.

To overcome the issue of the emissivity profile, the Kalman filter is combined with a gradient descent. As a consequence, the emissivity profile is updated at each step, following the temperature evolution as in a Levenberg-Marquardt algorithm. The prediction step in the Kalman is replaced by a gradient descent (with a dumping factor like in the Levenberg-Marquardt algorithm). Let J be the Jacobian matrix of the vectorial function defined by Eq. (3.20) and β the dumping factor. The prediction
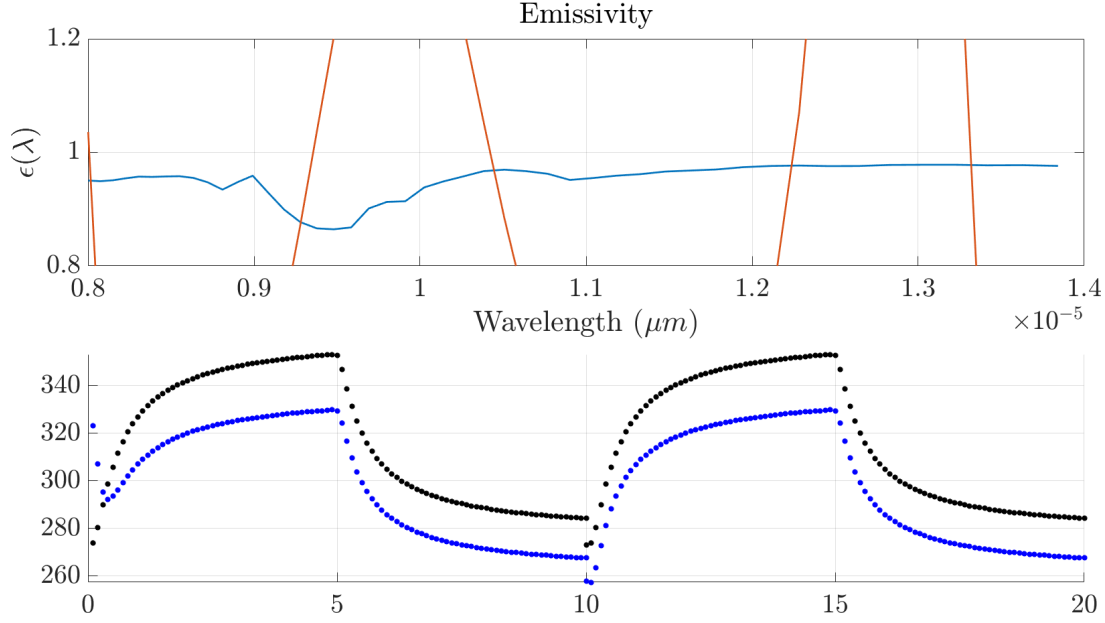
Figure 6.2: 6-order Tchebychev approximation through Kalman filtering

step in Eq. (6.10) is replaced by:

$$
\begin{cases}
\hat{\boldsymbol{x}}_{k|k-1} &= A_k\left(\left(\boldsymbol{J}^T\boldsymbol{J} + \beta\mathrm{diag}(\boldsymbol{J}^T\boldsymbol{J})\right)^{-1}\boldsymbol{J}^T(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k-1|k-1})\right) + \boldsymbol{B}\boldsymbol{u} \\
\boldsymbol{\Sigma}_{k|k-1} &= \boldsymbol{A}_k\boldsymbol{\Sigma}_{k-1|k-1}\boldsymbol{A}_k^T + \boldsymbol{\Sigma}_v
\end{cases}
\tag{6.12}
$$

Let $\mu \in \mathbb{R}$, the dumping factor is then updated once the correction has been made:

$$
\beta = \frac{\beta}{\mu}, \quad \text{if } \left\|\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1}\right\|_2 < \left\|\boldsymbol{y}_{k-1} - \boldsymbol{C}\hat{\boldsymbol{x}}_{k-1|k-1}\right\|_2
\tag{6.13}
$$

Fig. 6.3 shows the effect of the gradient descent. The first approximations are rough but once more data arrive into the system, the polynom fits more and more to the original profile. Of course, the same input parameters have been used for a fair comparison. As a consequence, the difference between the temperature estimate and the original one is reduced. Even if those results are encouraging, two main issues still remain. The first one is the fact that the emissivity profile is not bounded inside the $[0; 1]$ interval. The second one is that the 6-order polynomial seems insufficient for fitting the emissivity profile. The first issue can been tackled by using a modified parameterization to fit [121]:

$$
e_{\Delta\lambda_i} = \ln\left(\frac{\epsilon_{\Delta\lambda_i}}{1 - \epsilon_{\Delta\lambda_i}}\right)
\tag{6.14}
$$

The advantage is that $e_{\Delta\lambda_i}$ is defined on $\mathbb{R}$ while $\epsilon_{\Delta\lambda_i}$ is kept into $[0; 1]$. The second part could be
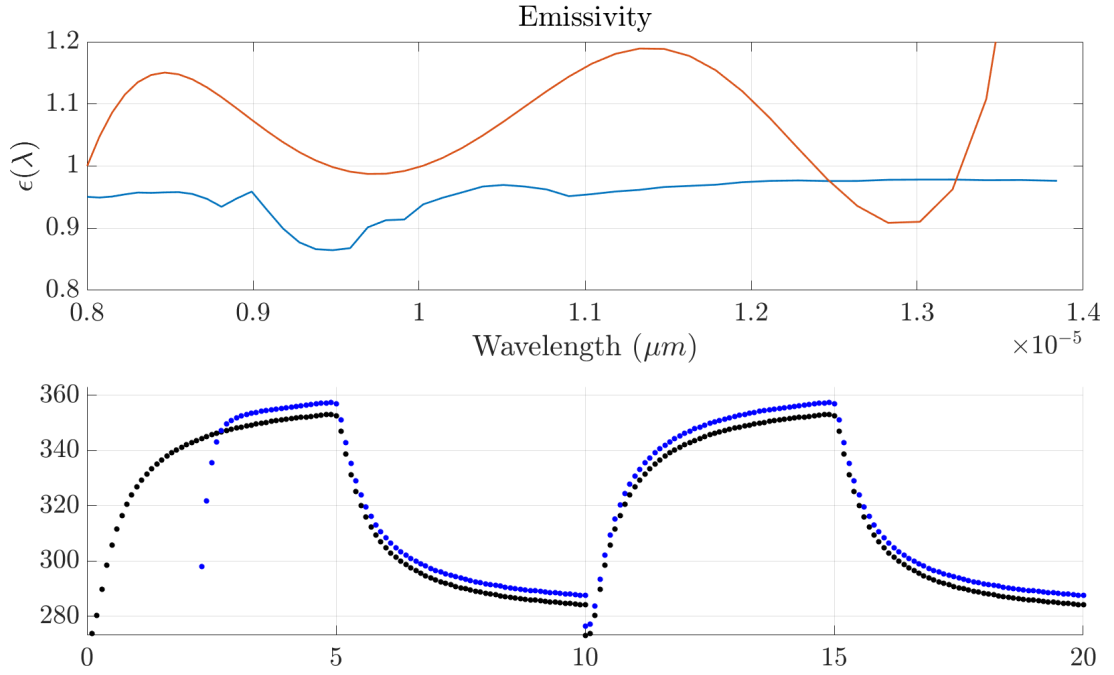
Figure 6.3: 6-order Tchebychev polynomial approximation through combined Kalman filter and gradient descent.

tackled with the previous presented splines system. Even though the convergence occurs to a better estimate, the algorithm is still leading to a local solution which is not satisfying even with the splines equation.

6.3.3 Synthesis

One of the main advantages of the Kalman filter is that updates are computationally efficient. However, the Kalman filter is restricted to unimodal distributions with single hypothesis and it can be sensitive to process noise. In the presented Kalman filter, the emissivity, temperature are estimated as state variables of our system which is not accurate regarding Eq. (6.5). In fact, the Kalman applied on the previous defined system leads to a local solution of the actual system. The Kalman filter is not well suited since the previous model is not linear (an extended Kalman filter would also in this particular case converge to a local solution due to a lack of information on the system parameters). Instead, the A_k scalar and emissivity should be considered as model parameters and therefore be estimated as is. In the next part, a particle filter in addition to the Kalman filter is presented in order to estimate conjointly the model's parameters and the temperature. Therefore, a particle filter is running in supplement to the Kalman one.
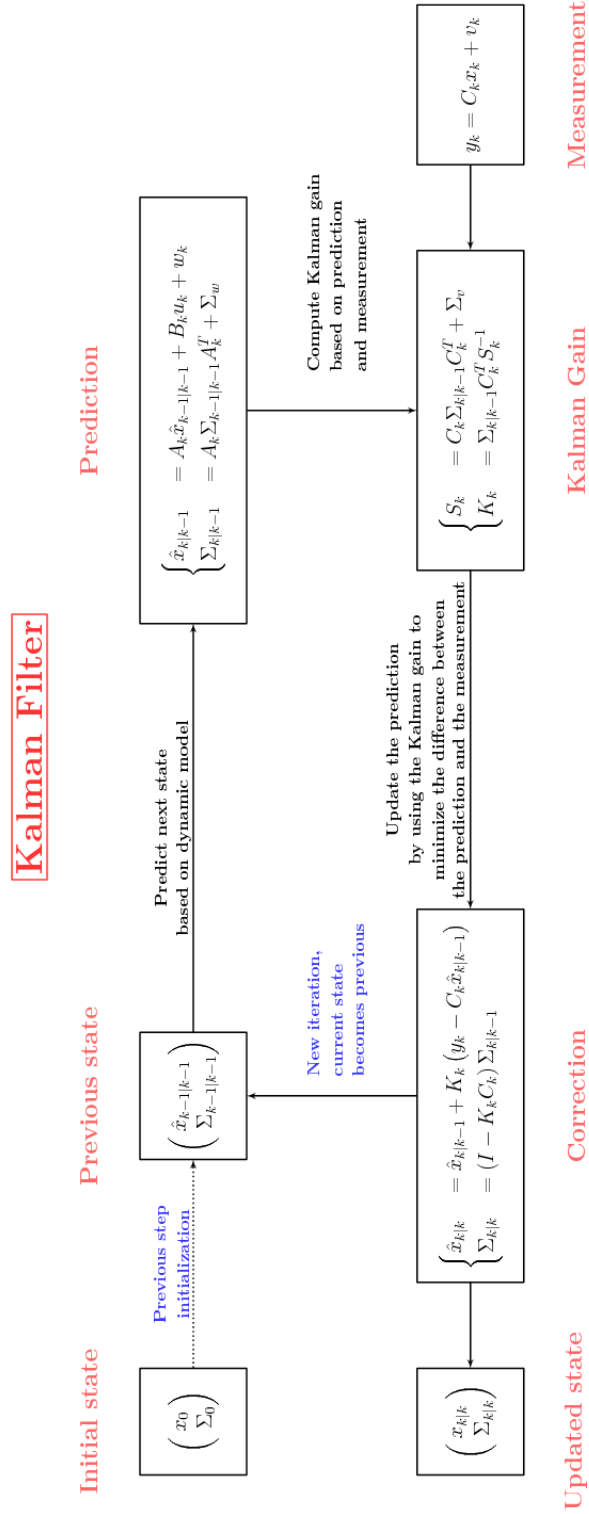
Figure 6.4: Kalman filter flowchart overview

6.4   Particle filter

As for the Kalman filter, the particle filter is a Bayes filter that uses a Monte-Carlo process to esti-
mate a probability density function. The density is represented by a set of particles (one possible state)
and their associated weights. In the Kalman filter, the prediction is made through the dynamics and
uncertainty is added based upon the process noise (dynamic noise). In the particle filter, a prediction
distribution is generated by sampling particles based upon their probability, which enables the usage
of the Bayes filter in the case that the state-space system is nonlinear. In fact, in that particular case
the optimal filter equations cannot be solved explicitly. Then those particles states are propagated
through dynamics and perturbed with noise. By tacking back the state system previously presented
and formulating it as a non-linear state-system:

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &\sim f_k(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) + \boldsymbol{v}_k \\
\boldsymbol{y}_k &\sim g_k(\boldsymbol{y}_k|\boldsymbol{x}_k) + \boldsymbol{w}_k
\end{aligned}
\tag{6.15}
$$

Let $\mathcal{B}(\mathbb{R}^n)$ be the Borel σ-algebra on $\mathbb{R}^n$. The objective is to get $p(\boldsymbol{x}_t \in \mathcal{A}) \approx p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t)$, where $\mathcal{A} \in \mathcal{B}(\mathbb{R}^n)$. If $N_{\text{particles}}$ is the number of particles: $p(\boldsymbol{x}_t \in \mathcal{A}) \underset{N_{\text{particles}} \to \infty}{=} p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t)$. To make those particles
evolve through time, a genetic mutation-selection algorithm is performed.

> **Note**
>
> In the particle filter, only the posterior density $p(\boldsymbol{x}_t|\boldsymbol{\mathcal{Y}}_t)$ is followed by the
> estimates. In contrast, the estimates obtained with the MCMC method pre-
> sented in Chapter 3 follows the full posterior density $p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_0|\boldsymbol{\mathcal{Y}}_t)$.

Let $(\boldsymbol{\zeta}_k^i)_{1 \le i \le N_{\text{particles}}}$ a set of independent random samples particles with their associated weight w_k^i,
following a probability density $p(\boldsymbol{x}_0)$. Two stages are then applied at each iteration: the mutation and
the selection. Those two steps are similar to the prediction and update steps performed in the Bayes
filter. More details of those equations and convergence results can be found in many textbooks such as
[55].

6.4.1   Mutation

A random move is locally applied to every particle at the beginning of the iteration according to
$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. Such move allows the set of particles to browse the field of solutions. The term $\sum_{i=1}^{N_{\text{particles}}} w_k^i \mathbf{1}_{\boldsymbol{\zeta}_k^i}$
is therefore an approximation of the conditional probability density $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$. Once the random move
has been done, an approximation of $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ is given by $\sum_{i=1}^{N_{\text{particles}}} w_k^i \mathbf{1}_{\hat{\boldsymbol{\zeta}}_k^i}$.

6.4.2 Selection

The weights of every particle $\hat{\boldsymbol{\zeta}}_k^i$ are evaluated using the Bayes filter equations :

$$w_k^i = \frac{w_{k-1}^i \, p(\boldsymbol{y}_k|\boldsymbol{x}_k = \hat{\boldsymbol{\zeta}}_k^i)}{\sum_{j=1}^{N_{\text{particles}}} w_{k-1}^j \, p(\boldsymbol{y}_k|\boldsymbol{x}_k = \hat{\boldsymbol{\zeta}}_k^j)} \tag{6.16}$$

6.4.3 Resampling

As the weights are updated, some particles may be irrelevant due to their low weight. A resampling is therefore done to replace this particle by a new one that will bring more significant information. Different sampling strategies can be used such as sequential importance resampling (SIR) [76] or sequential importance sampling (SIS). In order to estimate the moment to perform resampling, the standard effective particles number N_{eff} is used. The resampling is performed when $N_{\text{eff}} < 0.9 N_{\text{particles}}$, according to [119]

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N_{\text{particles}}} \left(w_k^i\right)^2} \tag{6.17}$$

Resampling consists in sampling new particles based on their weights and removing the ones whose weight is less than $1/N_{\text{particles}}$. Then all weights are sets to $1/N_{\text{particles}}$.

6.4.4 Synthesis

The particle filter can be used in the temperature - emissivity estimation by modifying the Eq. (6.5) and including the emissivity in the state vector [31]. Let $\tilde{\boldsymbol{x}}_k$ be this new state vector at time k:

$$\begin{aligned}
\tilde{\boldsymbol{x}}_k &= \begin{pmatrix} x_k & \boldsymbol{u}_k^T \end{pmatrix}^T & \tilde{\boldsymbol{u}}_k &= 0 \\
\tilde{\boldsymbol{C}}_k &= \begin{pmatrix} \boldsymbol{D}_k & \boldsymbol{C}_k \end{pmatrix} & \tilde{\boldsymbol{D}}_k &= 0 \\
\tilde{\boldsymbol{A}}_k &= \begin{pmatrix} A_k & 0 \\ 0 & \boldsymbol{I} \end{pmatrix}
\end{aligned} \tag{6.18}$$

In such method, the particles have components that correspond to both parameters and state approximations. Thus, the computed likelihood will take into account the overall approximation, which may not give a good indication about the approximation of the state or the parameters. This is also link with the expression of $\tilde{\boldsymbol{A}}_k$ for which the parameters are only following a random walk. Instead, another possibility is to use the Kalman filter for the state estimation and combine it with a particle filter for parameters estimation, which is called the interacting Kalman filter (IKF).

6.5 Interacting Kalman filter

In this section, a Bayesian parameter recursive estimation based on an interacting Kalman filter is presented. The parameters to be estimated are the emissivity and the scalar A_k. Interacting Kalman filter is a combination of two widely used Bayesian estimation methods: the particle filter and the Kalman filter [194]. Please note that in this case, the particle filter is used to estimate the parameters of the model and not the state vector, which is estimated by a Kalman filter [97].

6.5.1 Test case

A first approximation of f is to take the previous linear function $f_k : x_k \to A_k x_k$ with A_k a scalar. Such function could be extended to a more complicated one. As for example, [98] shows a regression model of surface temperature depending on wind speed and air temperature, [58] presents other models to predict road's surface and depth temperature based on the air temperature. In Appendix D, a seasonal autoregressive integrated moving average model (SARIMA) is presented in order to model the f_k function.

Depending on the different sensors available on the test site, the g function could also be extended. In a theoretical case, the g function could be taken as the material's emittance (used in the following test case). However, in a more realistic study case and as done later on in Section 6.8, it should be integrated over the sensor's measurement band and derived from the simplified radiative transfer model, presented in Eq. (2.51).

A given particle is defined through its parameter vector $\boldsymbol{\zeta}_k$:

$$\boldsymbol{\zeta}_k = \begin{pmatrix} a_k^\zeta & \alpha_{0,k}^\zeta & \alpha_{1,k}^\zeta & \dots & \alpha_{m,k}^\zeta \end{pmatrix}^T = \begin{pmatrix} a_k^\zeta & \boldsymbol{\alpha}_k^{\zeta T} \end{pmatrix}^T \tag{6.19}$$

Where $a_k^\zeta \in \mathbb{R}$ is a scalar that describes the temperature evolution through time and

$$\boldsymbol{\alpha}_k^\zeta = (\alpha_{i,k}^\zeta = ln(\epsilon_k^\zeta(\Delta\lambda_i)))_{1 \le i \le N}^T \tag{6.20}$$

Please note that the linearized emissivity model could also be used for which emissivity would be given by $\epsilon_k^\zeta(\Delta\lambda) = \sum_{i=0}^m \alpha_{i,k}^\zeta \Phi_i(\Delta\lambda)$ for a given $(\Phi_i)_{1 \le i \le m}$ function basis (see Eq. (3.32)). In this case, the linearized state-space system could still be used. The state-space model for a given particle is then defined as:

$$\begin{cases} x_{k+1}^\zeta &= A_k(\boldsymbol{\zeta}_k)x_k^\zeta + v_k^\zeta \\ \boldsymbol{y}_k^\zeta &= g(x_k, \boldsymbol{\zeta}_k) + \boldsymbol{w}_k^\zeta \end{cases} \tag{6.21}$$

Based on the previous measurement equation, and Eq. (6.20), g can be defined as:

$$g : (x_k, \boldsymbol{\zeta}_k) \mapsto \boldsymbol{\alpha}_k^\zeta - \frac{C_2}{\boldsymbol{\Delta\lambda}} x_k = \boldsymbol{\alpha}_k^\zeta + \boldsymbol{C} x_k \tag{6.22}$$

The likelihood criterion $p(\boldsymbol{y}_k|\zeta_{k|k-1})$ is therefore given by [194]:

$$p(\boldsymbol{y}_k|\boldsymbol{\zeta}_{k|k-1}) = \mathcal{N}\left(\boldsymbol{\alpha}_k^\zeta + \boldsymbol{C}\hat{x}_{k|k-1}^\zeta, \boldsymbol{C}\Sigma_{k|k-1}^\zeta + \Sigma_v\right) \tag{6.23}$$

One can see that the weights are adjusted based on a combination of the Kalman filter and the particle one. In order to improve the result, an additional criterion is added to the particles weights computation, based on the first derivative of the measurement equation in wavelength. By using the linear continuous equation measurement of the system:

$$\frac{\partial y(t,\lambda)}{\partial\lambda} = \frac{1}{\epsilon(\lambda)}\frac{\mathrm{d}\epsilon(\lambda)}{\mathrm{d}\lambda} \tag{6.24}$$

6.5.2 Synthesis

Fig. 6.5 shows the result on the theoretical test case. One can see that the emissivity profile shape is restored leading to a satisfying temperature estimation. However, it is important to note that the current algorithm implementation is computationally intensive. Although the algorithm can be parallelized with a GPGPU approach, the number of particles needed for a high image resolution would make the algorithm complexity explode. Furthermore, due to the additional wavelength first derivative criterion, the algorithm is suitable for hyper-spectral measurement data only, with short distances between the spectral bands.
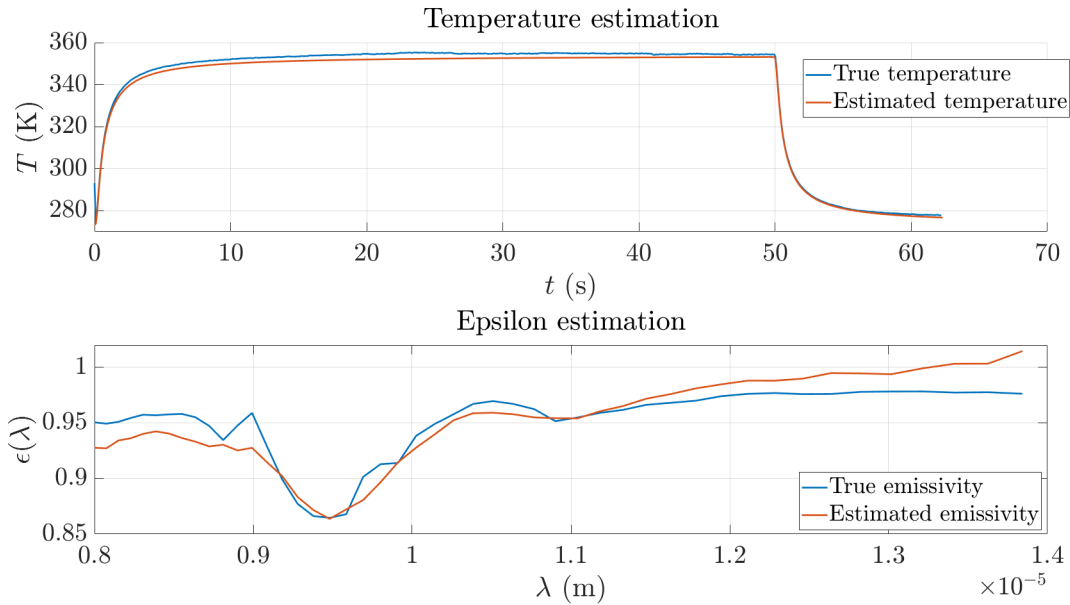

Figure 6.5: Temperature and emissivity estimation through the interacting Kalman filter (8000 particles)

Figure 6.6: Interacting Kalman filter flowchart overview

6.6 Kriging

In the previous section, the temporal and multi-band parts of the data have been used to get an estimate of the temperature and emissivity from infrared image. However, one interesting part in those data has not been taken into acount yet: the spatial information of the image. Spatial issues have been presented in Chapter 4, in terms of discretization, angle of view and calibration. This section focuses on extracting spatial information on the images. Moreover, even if the interactive Kalman filter has shown interesting results on a simple sensor data, the required computation time to process an entire image may not be convenient for *in-situ* practical applications. This section focuses on exploiting the spatial information to serve two purposes:

- Bringing spatial information to the temperature/emissivity retrieval,
- Get rid of the short spectral band distance requirement,
- Reducing the required computation time.

Such goals have been reached by using a Kriging model on top of the interactive Kalman filter. The kriging method was developed in the 60's by Georges Matheron, from Danie G. Krige works in the geo-statistical field. Kriging is a spatial interpolation method that provides estimates at unknown locations based on sampled ones. Under some assumptions, kriging gives the best linear unbiased prediction for the intermediate values. Based on the modeling of the experimental variogram, the spatial interpolation is made by computing the expected value of a random variable. We will introduce first the Kriging equations and in the next section present the final temperature / emissivity retrieval method KIKF (Kriged Interactive Kalman Filter).

Let $\mathscr{S} \subset \mathbb{R}^2$ be the studied spatial domain, $\mathcal{S} \subset \mathscr{S}$ be the measurement domain and $\mathcal{S}^* \subset \mathscr{S}$ be the spatial domain where lies the points on which the estimation has to be made such as $\mathcal{S} \cup \mathcal{S}^* = \mathscr{S}$.

6.6.1 Variogram

Definition

Let $s \in \mathcal{S}$ a spatial point with its associated measurement $y(s)$. Consider now the measurement at point $s + h$, $h \in \mathbb{R}^2$. The variogram is defined as the expectation of the random variable $(y(s) - y(s + h))^2$ [92]. Since in practice only one realization of the couple $(y(s), y(s+h))$ is known, the hypothesis behind the modelization of the variogram is to assume that the variogram only depends on the modulus and direction h and not its location s. Therefore, the variogram is defined as:

$$2\gamma(\boldsymbol{h}) = \frac{1}{N(\boldsymbol{h})} \sum_{i=1}^{N(\boldsymbol{h})} \left(y(\boldsymbol{s_i}) - y(\boldsymbol{s_i} + \boldsymbol{h})\right)^2 \tag{6.25}$$

where $N(\boldsymbol{h})$ represents the number of points separated by a vector $\boldsymbol{h}$ in $\mathcal{S}$. The variogram can be viewed as the estimation of the variance of the resulting estimation prediction error when the mea-

surement in $\boldsymbol{s} + \boldsymbol{h}$ is estimated from the measurement in $\boldsymbol{s}$. Modeling the variogram during kriging is one of the most important part since the final estimation will depend on this spatial model. In the particular case of a spatially stationnary field, this variogram can be expressed as a spatial covariance [48].

Variogram models

The variogram can be deduced from the experimental data based on Eq. (6.25) (experimental variogram). However, such methodology will not give a predictive model neither respect kriging constraint's hypothesis (continuous variance function, the variance function is defined on $\mathcal{S}^2$ and is stationnary of order 2). As a consequence, once the experimental variogram is computed, a variogram model is fitted to it. The choice of the variogram model is usually problem-dependant and the choice for a model should be done carefully, by considering the physics behind the measurements and looking at the experimental's variogram shape. Many variograms model exists, the following non-exhaustive list of models is an illustration of the most common ones:

Pure nugget This model corresponds to a constant one. It corresponds to a white noise.

$$\gamma(\boldsymbol{h}) = \begin{cases} C, & \text{if } h > 0 \\ 0, & \text{if } h = 0 \end{cases} \tag{6.26}$$

Gaussian

$$\gamma(\boldsymbol{h}) = C(1 - e^{-\left(\frac{h}{a}\right)^2}) \tag{6.27}$$

Matérn The Matérn covariance function is commonly used for its mathematicals properties, in particular due to its finite differentiability (for finite ν) [173] and the fact that it includes the modified Bessel function of second kind makes it respect physical equations, depending on the choice of ν.

$$C_\nu(\boldsymbol{h}) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\boldsymbol{h}}{\nu}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{\boldsymbol{h}}{\nu}\right) \tag{6.28}$$

where Γ is the gamma function, and K_ν the modified Bessel function of second kind, $\rho \in \mathbb{R}^{+*}, \nu \in \mathbb{R}^{+*}$

6.6.2 Simple Kriging

Depending on the knowledge that one can get on its data, or the knowledge that one wants to infere into its model, the kriging method and formula has to be adapted. In Simple Kriging (SK), the measurements y are assumed to be the sum of a known trend function $\mu : \boldsymbol{s} \in \mathcal{S} \rightarrow \mu(\boldsymbol{s}) \in \mathbb{R}$ and a

centered square-integrable process Z. The covariance of Z is known (learned from previous section) :

$$y(\boldsymbol{s}) = \mu(\boldsymbol{s}) + Z(\boldsymbol{s}) \tag{6.29}$$

Let $\boldsymbol{s}^* \in \mathscr{S}^*$ and $\boldsymbol{S} = (\boldsymbol{s_i})_{1 \leq i \leq n}$, the Simple Kriging equations are [148]:

$$
\begin{aligned}
\mu(\boldsymbol{s}^*|\boldsymbol{S}) &= \mu(\boldsymbol{s}^*) + \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})^T \boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{S})) \\
\sigma^2(\boldsymbol{s}^*|\boldsymbol{S}) &= C(\boldsymbol{s}^*, \boldsymbol{s}^*) - \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})^T \boldsymbol{C}^{-1} \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})
\end{aligned}
\tag{6.30}
$$

where $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{s_i}, \boldsymbol{s_j})_{1 \leq i,j \leq n}$ is the covariance matrix of $y(\boldsymbol{S})$

Let consider now that Z is a Gaussian random vector $\mathcal{N}(0, C)$ where the covariance function C is known:

$$
\begin{bmatrix} \boldsymbol{Y}(\boldsymbol{S}) = \boldsymbol{\mu}(\boldsymbol{S}) + \boldsymbol{Z}(\boldsymbol{S}) \\ Y(\boldsymbol{s}^*) = \mu(\boldsymbol{s}^*) + Z(\boldsymbol{s}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(\boldsymbol{S}) \\ \mu(\boldsymbol{s}^*) \end{bmatrix}, \begin{bmatrix} \boldsymbol{C} & \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{s})^T \\ \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{s}) & C(\boldsymbol{s}^*, \boldsymbol{s}^*) \end{bmatrix} \right) \tag{6.31}
$$

The conditioning on a Gaussian vector leads to

$$
\begin{aligned}
p(Y(\boldsymbol{s}^*)|\boldsymbol{Y}(\boldsymbol{S}) = \boldsymbol{y}) \sim \mathcal{N}(&\boldsymbol{\mu}(\boldsymbol{s}^*) + \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})^T \boldsymbol{C}^{-1}\left(\boldsymbol{y} - \boldsymbol{\mu}(\boldsymbol{S})\right), \\
&C(\boldsymbol{s}^*, \boldsymbol{s}^*) - \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})^T \boldsymbol{C}^{-1} \boldsymbol{C}(\boldsymbol{s}^*, \boldsymbol{S})
\end{aligned}
\tag{6.32}
$$

In the previous equation, we recognize the SK equations. In fact, in the case that Z is assumed Gaussian, the SK equations coincide with the classical conditionning equations. From the SK equations, one can note the following properties of the Kriging: the SK interpolates exactly the experimental data. The SK variance is independant of the measurements y (homoscedasticity), whatever the chosen kernel C. For a spatially stationary field, the kriging equations are a simple linear minimum mean square error (LMMSE) interpolator.

6.6.3 Universal Kriging

In SK, the trend function μ is assumed to be known. However, in most practical applications, this is not the case. Universal Kriging's (UK) objective is to estimate on the fly the best linear predictions of Y on $\mathscr{S}$ and the trend μ. In this case, the trend is of the form $\mu(\boldsymbol{s}) = \sum_{i=1}^{q} \beta_i f_i(\boldsymbol{s})$ where the vector $\boldsymbol{\beta} = (\beta_i)_{1 \leq i \leq q}$ is unknown and $f_i : \mathbb{R}^2 \to \mathbb{R}$ are given basis functions. Therefore, we have:

$$Y(\boldsymbol{s}) = \sum_{i=1}^{q} \beta_i f_i(\boldsymbol{s}) + Z(\boldsymbol{s}) \tag{6.33}$$

Let note $\boldsymbol{F}(\boldsymbol{s}^*) = (f_i(\boldsymbol{s}^*))_{1 \le i \le q}$ and abusively, $\boldsymbol{F}(\boldsymbol{S}) = (f_i(\boldsymbol{s_j}))_{1 \le i \le q, 1 \le j \le n}$. In the case that $Z \sim \mathcal{N}(0, C)$ and $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B})$:

$$
\begin{bmatrix} Y(\boldsymbol{S}) \\ Y(\boldsymbol{s}^*) \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}(\boldsymbol{S})^T \boldsymbol{\beta} + Z(\boldsymbol{S}) \\ \boldsymbol{F}(\boldsymbol{s}^*)^T \boldsymbol{\beta} + Z(\boldsymbol{s}^*) \end{bmatrix}
$$
$$
\sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{F}(\boldsymbol{S})^T \boldsymbol{b} \\ \boldsymbol{F}(\boldsymbol{s}^*)^T \boldsymbol{b} \end{bmatrix}, \begin{bmatrix} \boldsymbol{C} + \boldsymbol{F}(\boldsymbol{S})\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T & C(\boldsymbol{S}, \boldsymbol{s}^*) + \boldsymbol{F}(\boldsymbol{S})\boldsymbol{B}\boldsymbol{F}(\boldsymbol{s}^*)^T \\ C(\boldsymbol{S}, \boldsymbol{s}^*)^T + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T & C(\boldsymbol{s}^*, \boldsymbol{s}^*) + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{s}^*)^T \end{bmatrix} \right)
$$
(6.34)

By using the conditionning on Eq. (6.34), we get the kriging estimator's equations:

$$
\mu(\boldsymbol{s}^*|\boldsymbol{S}) = F(\boldsymbol{s}^*)^T \boldsymbol{b} + \left(\boldsymbol{C}(\boldsymbol{S}, \boldsymbol{s}^*)^T + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T \right)
$$
$$
\left(\boldsymbol{C} + \boldsymbol{F}(\boldsymbol{S})\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T \right)^{-1} \left(\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{S})^T \boldsymbol{b} \right)
$$
$$
\sigma^2(\boldsymbol{s}^*|\boldsymbol{S}) = C(\boldsymbol{s}^*, \boldsymbol{s}^*) + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{s}^*)^T - \left(\boldsymbol{C}(\boldsymbol{S}, \boldsymbol{s}^*)^T + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T \right)
$$
$$
\left(\boldsymbol{C} + \boldsymbol{F}(\boldsymbol{S})\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T \right)^{-1} \left(\boldsymbol{C}(\boldsymbol{S}, \boldsymbol{s}^*)^T + \boldsymbol{F}(\boldsymbol{s}^*)\boldsymbol{B}\boldsymbol{F}(\boldsymbol{S})^T \right)^T
$$
(6.35)

In order to get the UK formula, we take the limits of $\mu(\boldsymbol{s}^*|\boldsymbol{S})$ and $\sigma^2(\boldsymbol{s}^*|\boldsymbol{S})$ when the prior on the trend weights $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B} = \lambda\tilde{\boldsymbol{B}}), \lambda \in \mathbb{R}$ becomes non informative. That is to say when $\lambda \to \infty$ and $b = 0$. Therefore, we define the estimates as:

$$
\hat{\mu}(\boldsymbol{s}^*) = \lim_{\substack{\lambda \to \infty \\ \boldsymbol{b}=0}} \mu(\boldsymbol{s}^*|\boldsymbol{S})
$$
$$
\hat{\sigma}^2(\boldsymbol{s}^*) = \lim_{\substack{\lambda \to \infty \\ \boldsymbol{b}=0}} \sigma^2(\boldsymbol{s}^*|\boldsymbol{S})
$$
(6.36)

In order to simplify the expression of the previous equation, the two following matrix inversion lemma are used [87]:

Lemme 6.6.1 $((A + UBV)^{-1} = A^{-1} - A^{-1}U \left(I + BVA^{-1}U \right)^{-1} BVA^{-1})$.

Lemme 6.6.2 $((A^{-1} + B^{-1})^{-1} = A - A(A + B)^{-1}A)$. Let K be a commutative field. Let A and B be two inversible matrices of $\mathbb{K}^N$ and suppose that their sum and the sum of their inverse are also inversible. Let $x \in \mathbb{K}^N$, I be the identity matrix of $\mathbb{K}^N$ and look for a solution such as $(A^{-1} + B^{-1})^{-1} = A + x$.

Therefore,

$$(A + X)(A^{-1} + B^{-1}) = I$$
$$\Longleftrightarrow AA^{-1} + AB^{-1} + X(A^{-1} + B^{-1}) = I$$
$$\Longleftrightarrow X = -AB^{-1}(A^{-1} + B^{-1})^{-1}$$
$$\Longleftrightarrow X = -AB^{-1}\left[(A^{-1}B + I)B^{-1}\right]^{-1} \qquad (6.37)$$
$$\Longleftrightarrow X = -AB^{-1}B(A^{-1}B + I)^{-1}$$
$$\Longleftrightarrow X = -A(A^{-1}B + I)^{-1}$$
$$\Longleftrightarrow X = -A(A + B)^{-1}A$$

By using successively the first and second lemma, we can derive the expression of $(C + F(s)BF(s)^T)^{-1}$:

$$\left(C + F(S)BF(S)^T\right)^{-1} = C^{-1} - C^{-1}F(S)\left(B^{-1} + F(S)^TC^{-1}F(S)\right)^{-1}F(S)^TC^{-1}$$
$$= C^{-1} - C^{-1}F(S)\left[\left(F(S)^TC^{-1}F(S)\right)^{-1} - \left(F(S)^TC^{-1}F(S)\right)^{-1}\left(\left(F(S)^TC^{-1}F(S)\right)^{-1} + B\right)^{-1}\right.$$
$$\left.\left(F(S)^TC^{-1}F(S)\right)^{-1}\right]F(S)^TC^{-1}$$

$$(6.38)$$

By multiplying the previous equation on the left by $F(s^*)BF(S)^T$ and on the right by y:

$$F(s^*)BF(S)^T(C + F(S)BF(S)^T)^{-1}y$$
$$= F(s^*)B((F(S)^TC^{-1}F(S))^{-1} + B)^{-1}(F(S)^TC^{-1}F(S))^{-1}F(S)^TC^{-1}y$$
$$= F(s^*)((F(S)^TC^{-1}F(S))^{-1}B^{-1} + I)^{-1}(F(S)^TC^{-1}F(S))^{-1}F(S)^TC^{-1}y \qquad (6.39)$$
$$\underset{B\to\infty}{\to} F(s^*)(F(S)^TC^{-1}F(S))^{-1}F(S)^TC^{-1}y$$

Let $\tilde{\beta} = (F(S)^TC^{-1}F(S))^{-1}F(S)^TC^{-1}y$:

$$F(s^*)BF(S)^T(C + F(S)BF(S)^T)^{-1}y \underset{B\to\infty}{\to} F(s^*)\tilde{\beta} \qquad (6.40)$$

This time, by multiplying the Eq. (6.38) on the left by $C(S, s^*)^T$ and on the right by y:

$$C(S, s^*)^T(C + F(S)BF(S)^T)^{-1}y$$
$$= C(S, s^*)^T\left[C^{-1} - C^{-1}F(S)\left(B^{-1} + F(S)^TC^{-1}F(S)\right)^{-1}F(S)^TC^{-1}\right]y$$
$$\underset{B\to\infty}{\to} C(S, s^*)^TC^{-1}\left[y - F(S)\left(F(S)^TC^{-1}F(S)\right)^{-1}F(S)^TC^{-1}y\right] \qquad (6.41)$$
$$= C(S, s^*)^TC^{-1}\left[y - F(S)\tilde{\beta}\right]$$

Eq. (6.40) and Eq. (6.41) are the two terms in Eq. (6.36):

$$\hat{\mu}(\boldsymbol{s}^*) = \boldsymbol{F}\left(\boldsymbol{s}^*\right)\tilde{\boldsymbol{\beta}} + \boldsymbol{C}\left(\boldsymbol{S}, \boldsymbol{s}^*\right)^T \boldsymbol{C}^{-1}\left[\boldsymbol{y} - \boldsymbol{F}(\boldsymbol{S})\tilde{\boldsymbol{\beta}}\right] \tag{6.42}$$

Similarly,

$$\hat{\sigma}^2(\boldsymbol{s}^*) = C(\boldsymbol{s}^*, \boldsymbol{s}^*) - \boldsymbol{C}(\boldsymbol{S}, \boldsymbol{s}^*)^T \boldsymbol{C}^{-1} C(\boldsymbol{S}, \boldsymbol{s}^*) + \boldsymbol{U}^T\left(\boldsymbol{F}(\boldsymbol{S})^T \boldsymbol{C}^{-1} \boldsymbol{F}(\boldsymbol{S})\right)^{-1}\boldsymbol{U} \tag{6.43}$$

where $\boldsymbol{U} = F(S)^T\boldsymbol{C}^{-1}\boldsymbol{C}(\boldsymbol{S}, \boldsymbol{s}^*) - \boldsymbol{F}(\boldsymbol{s}^*)^T$.

6.6.4 Kriging examples

In order to get an overview of Kriging, here are simple examples made with the DiceKriging [149] library in R. Those examples are inspired from the DiceKriging help package.

1D interpolation case: covariance kernel type influence

Let 5 measurement points taken every 0.5s starting at 1s. We want to make the interpolation on $t \in [0, 4]$ with a second order polynomial trend.
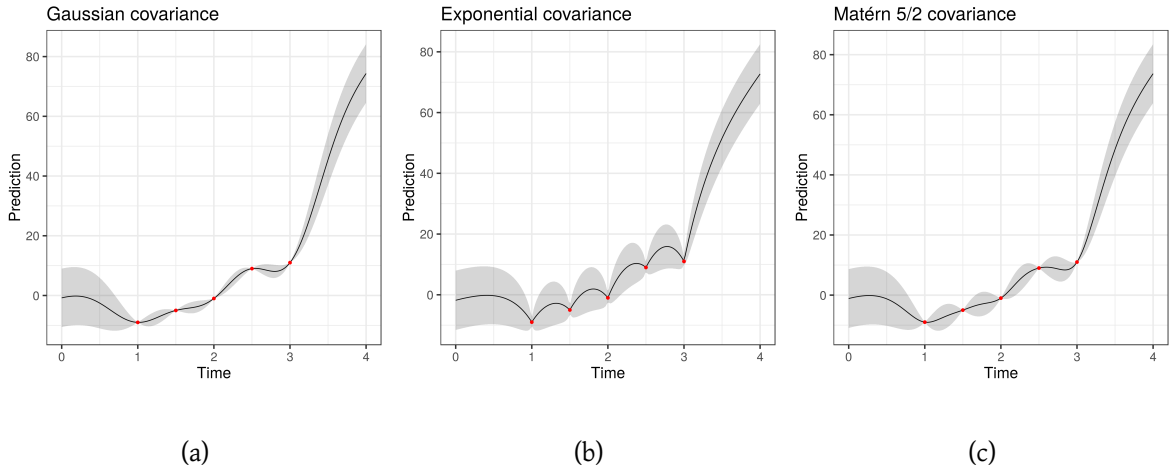

(a)   (b)   (c)

Figure 6.7: Influence of the covariance kernel on the prediction (in black). The 95% confidence interval in light gray and the input measurements points in red.

It is clear on this simple case that the interpolation will perform differently depending on the chosen covariance. The choice of the covariance kernel will mainly depend on the shape and nature of the input data. As described earlier, the Matérn 5/2 is often used by default, and it is the standard option in the DiceKriging package. Moreover, the confidence interval is larger at the boundaries, due to the lack of information. At the opposite, the measurements points are exactly interpolated.

1D interpolation case: trend influence

Similarly to the choice of the covariance kernel, the choice of the trend function will also have an impact on the prediction. By using the same example but this time with a Matérn 5/2 covariance kernel and different trend models, Fig. 6.8 shows the influence of the trend model, for three different cases: constant, effing and sinusoidal.
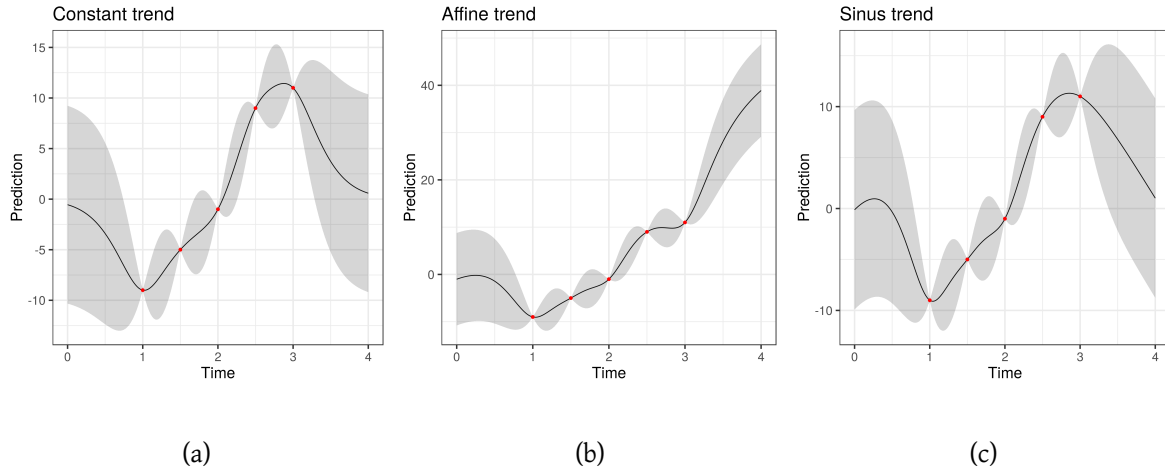

Figure 6.8: Influence of the trend on the prediction (in black). The 95% confidence interval in light gray and the input measurements points in red.

2D interpolation case: experimental variogram

In order to illustrate the importance of choosing the right model for the variogram fitting, a simple example of 2D data have been taken from the Meuse R package [139]. Zinc concentration over a region has to be estimated based on some measurements. Fig. 6.9 shows the differences between a Spherical and a Matérn model for this specific case. The final concentration estimation will therefore be different depending on the chosen covariance model.

6.6.5 Synthesis

The equations of the Kriging have been derived in this section. The simple examples at the end show that the variogram model choice is a fundamental parameter in Kriging estimation. In particular, the fit of the experimental variogram has to be done cautiously in order to be consistent results. In the following, a Kriging step will be combined with the previously presented IKF method.
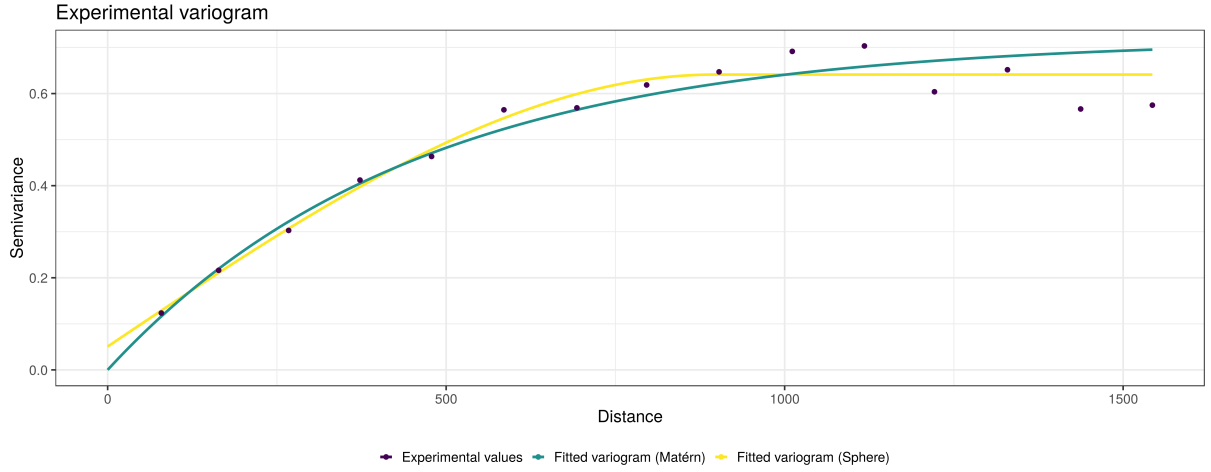
Figure 6.9: Difference between a spherical and Matérn variogram models for a 2D problem

6.7 New Bayesian approaches

6.7.1 A Kriged Interacting Kalman Filter based temperature and emissivity estimation

In order to exploit most information from the recorded images, a combination of both Kriging and Kalman methods has been developed. The idea behind this approach is to use the Kriging to conserve the spatial correlation between the objects in the scene and the Kalman filter to use the long-term characteristic of the measurements. Kriged-Kalman Filters (KKF) have been particularly used in the climatic domain, to establish temporal and spatial evolving models with few *a-priori* [48, 143]. Such filters can also be used for optimal sensors placement as in [171] for example.

Main problem statement

Let $S \subset \mathbb{R}^2$ be the image space domain and $T \subset \mathbb{R}$ the time domain. Let suppose that an object has a slow varying emissivity value through time. After removing the geometric coefficients, the measurement equation can be written as:

$$y(s, t, \lambda) = \epsilon(s, \lambda) M(s, t, \lambda), \quad s \in S, t \in T, \lambda \in \mathbb{R} \tag{6.44}$$

with y the measured value at the sensor and wavelength λ after considering the geometry of the scene, ϵ the emissivity at location s and M the emittance at location s and time t.

A separation of the dynamic and stationary parts of the measure is performed:

$$log(y(s, t, \lambda)) = \underbrace{log(\epsilon(s, \lambda))}_{\text{stationnary}} + \underbrace{log(M(s, t, \lambda))}_{\text{dynamic}} \tag{6.45}$$

155

As done previously by using Wien's approximation and putting a linearized model on $\epsilon(s, \lambda)$, let v_s and v_d be the stationary component and the dynamic one respectively.

$$\begin{aligned}
\tilde{y}(s,t,\lambda) &= log(\epsilon(s,\lambda)) - \frac{C_2}{\lambda T(s,t)} \\
&= C_t(\lambda)\left(v_s(s) + v_d(s,t)\right)
\end{aligned} \tag{6.46}$$

In the following, one important assumption will be that the quantities follow a Gaussian law. This assumption can be discussed regarding the involved log functions in the previous equations. However, since the sampling of the IRT is generally more important than the involved thermal time constants of the observed surfaces, this oversampling is used so that the measurements are averaged over N time values. We finally get the spatio-temporal evolution with the following state model, with a process noise vector $\nu_s(t) \sim \mathcal{N}(0, \sigma_s(t))$:

$$\tilde{y}(s,t,\lambda) = C_t(\lambda)\left(v_s(s) + v_d(s,t)\right) + \nu_s(t) \tag{6.47}$$

Tracking of the dynamic component - Kalman

Let the spatially stationary field $v_s(s)$ follows a random process with μ_ϵ its spatial expectation and Σ_ϵ its spatial covariance matrix:

$$v_s(s) \sim \mathcal{N}(\mu_\epsilon, \Sigma_\epsilon) \tag{6.48}$$

$v_s(s)$ is firstly interpreted as a noise term to get the measurement equation that will be used in the Kalman filter [192]:

$$\tilde{y}(s,t,\lambda) - C_t(\lambda)\mu_\epsilon = C_t(\lambda)v_d(s,t) + C_t(\lambda)\left(v_s(s) - \mu_\epsilon\right) + \nu_s(t) \tag{6.49}$$

The measurement model equation is therefore derived from Eq. (6.49):

$$\hat{y}(s,t,\lambda) = C_t(\lambda)v_d(s,t) + e_s(t), \quad \text{with} \begin{cases} \hat{y}(s,t,\lambda) &= \tilde{y}(s,t,\lambda) - C_t(\lambda)\mu_\epsilon \\ e_s(t) &= C_t(\lambda)\left(v_s(s) - \mu_\epsilon\right) + \nu_s(t) \\ e_s(t) &\sim \mathcal{N}(0, C_t(\lambda)\Sigma_\epsilon C_t(\lambda)^T + \sigma_\nu^2) \end{cases} \tag{6.50}$$

With the dynamic equation defined in the previous section and process noise $\eta_s(t) \sim \mathcal{N}(0, \sigma_\eta^2)$, the resulting state-space model is:

$$\begin{cases} \hat{y}(s,t,\lambda) &= C_t(\lambda)v_d(s,t) + e_s(t) \\ v_d(s,t) &= A_t v_d(s,t) + \eta_s(t) \end{cases} \tag{6.51}$$

This state-space model is updated each time new incoming data are gathered through a Kalman Filter, as explained in Section 6.3.

One can see from Eq. (6.50) that the presented method needs the priors μ_ϵ and Σ_ϵ. The prior on Σ_ϵ can be roughly estimated by the input data assuming that the space covariance emissivity matrix will remain the same through time (as long as the scene is static). However, the prior on μ_ϵ is a strong one that will have an impact on the final temperature and emissivity estimation. If reference emissivity points are available and spread in the image, then those references should be taken as locations for which the Kalman filter must be executed due to a fair estimation of μ_ϵ.

More generally, to overcome the lack of knowledge on μ_ϵ, a possibility is to replace the Kalman filter by an interacting Kalman filter. In that case, the weights of the particles are updated by assuming that for each s's neighbors measurements locations, the temperature is the same as in s.

Furthermore, the A_t dynamic evolution of the temperature is also unknown. The A_t value is therefore added to the unknowns of the model and added to the components of the particles.

> **Note**
>
> A_t represents the temperature's evolution from one time to the next one. As a consequence, A_t can be greater than 1. This is actually an issue for the Kalman recursive filter. If A_t is greater than 1, the initial temperature value may not be forgotten as the algorithm is computed. To overcome this issue, and for outdoor data process, the algorithm is started during the morning to guarantee that $A_t < 1$ most of the time (the dynamic evolution is in $\frac{1}{T}$).

Spatial data information - Kriging

The Kriging step is used in order to exploit the spatial information by using the assumption that the spatial covariance matrix model is constant through time. The Kriging interpolation is used to retrieve the emissivity values on the entire image. As a consequence, only a few points on the image are tracked through time. This methodology helps to reduce the computation time of the particle filter algorithm on the entire image. By writing the measurement equation with the stationary component Eq. (6.52), the emissivity at any new location $\tilde{s} \in S$ within the image can be derived from the Kriging Eq. (6.53) (see Section 6.6):

$$\tilde{y}(s,t,\lambda) - C_t(\lambda)v_d(s,t) = C_t(\lambda)v_s(s) + \nu_s(t) \tag{6.52}$$

$$v_s(\tilde{s},t) = \mu_\epsilon + \Sigma_\epsilon C_t^T (C_t^T \Sigma_\epsilon C_t^T + \sigma_\nu^2 I)^{-1}(\hat{y}(s,t,\lambda) - C_t v_d(s,t) - C_t \mu_\epsilon) \tag{6.53}$$

The Kriging needs therefore an experimental variogram to fit a model to the emissivity component. In order to get an initial estimate of the emissivity, it is necessary to give a first estimate of the temperature at locations s. This is achieved by choosing a measurement for which the temperature is uniform or at least some knowledge about the temperature gradient is known for the considered image.

Particles update

The weights of the particles are then updated by assuming that locally (on a given neighborhood) the temperature is the same. The likelihood is then derived by reprojecting the estimated emissivity and the given local temperature to a measurement value and computing the mean square error. Contrary to the particle filter presented in section Section 6.4, the particles are thrown for only a given number of pixel over the entire image and not for a single pixel, which reduces the algorithm complexity.
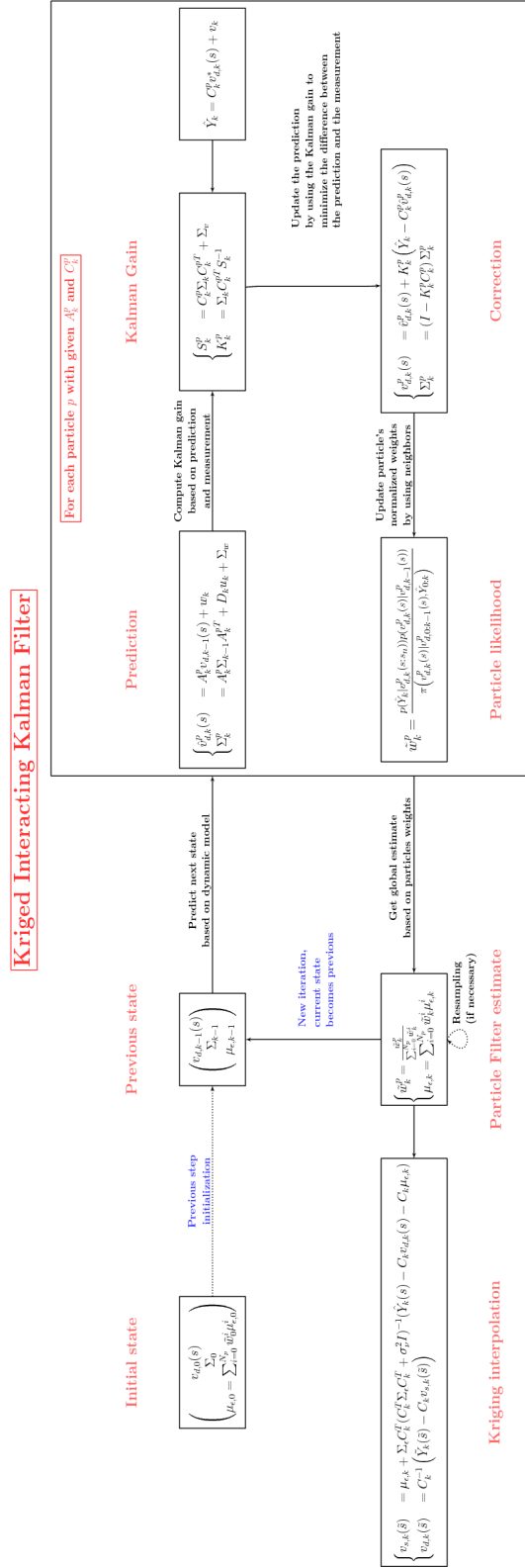
Figure 6.10: Kriged Interacting Kalman filter flowchart overview

6.7.2 CMA-ES applied to the temperature / emissivity estimation

CMA-ES (covariance matrix adaptation evolution strategy) belongs to stochastic and derivative-free methods, based on an evolution strategy [84]. It has been used to address the temperature and emissivity separation problem and compare it to the KIKF method. The idea behind an evolution strategy is to choose λ children from μ parents population. In order to produce one child, ρ parents are combined. Then children are mutated, usually based on a multivariate normal distribution. Finally, the μ best children based on the evaluation of a minimization criteria are selected and become the new parents population.

The CMA-ES algorithm relies on this principle. However, instead of mutating the children with a multivariate normal distribution *a priori*, the covariance matrix of this distribution is updated.

Theory

As for the particle filter presented previously, the CMA-ES algorithm will have a sampling step, a mutation and then a selection. However, instead of updating only the mean of the search points, the covariance matrix (*i.e.* the shape of the distribution ellipsoid) is also updated. Let λ be the population size and $i \in [1, \lambda]$. A sample x_i is therefore normally distributed with its mean m and covariance matrix C

$$x_i \sim m + \sigma \underbrace{\mathcal{N}_i(0, C)}_{y_i} \tag{6.54}$$

Where $\sigma \in \mathbb{R}_+$ is a step-size that controls the particles evolution.

From Eq. (6.54) different evolution strategies (ES) can be applied to update μ_i, Σ_i and σ. The $(\mu/\mu, \lambda)$-ES chooses the best μ points from new solutions which is a non-elitistic selection. Then, recombination is applied. Let f be the objective function to be minimized. The solutions points are ranked according to their objective value $f(x_{1:\lambda}) \leq \ldots \leq f(x_{\lambda:\lambda})$. The mean is then updated according to their weights:

$$\tilde{m} = m + \sigma \sum_{i=1}^{\mu} w_i y_{i:\lambda} \tag{6.55}$$

with $w_1 \geq \ldots \geq w_\mu > 0$, $\sum_{i=1}^{\mu} w_i = 1$ and $N_{\text{eff}} \approx \lambda/4$ where N_{eff} has been defined in the particle filter section.

In order to control the algorithm evolution, a path length control quantity p is introduced to update both the covariance matrix and the step-size factor. This is achieved by using a cumulative step-size adaptation (CSA) as described in [32]:

Spatial optimization problem

As for the KIKF method, the approximation that the temperature is the same on a given pixel's window has been used.

Let w_p a window of size $W \times H$ (where $W, H \in \mathbb{N}$ are usually odd) that is centered on pixel $p = (i, j)$, where $i \in [2, N-1]$ and $j \in [2, N-1]$. The assumption that the emissivity and temperature are constant over this window is made:

$$\forall k \in w, \begin{cases} \epsilon_k &= \epsilon_p \\ T_k &= T_p \end{cases} \tag{6.56}$$

Therefore, we can formulate the following optimization problem, where $\boldsymbol{Y}$ is the image data and $g : (\boldsymbol{x}) \to \mathbb{R}$ the observation model :

$$\forall p \in (N-2) \times (M-2), \quad \boldsymbol{x_p} = (\epsilon_p, T_p) = \arg\min_{\vec{x}} \left(\sum_{k \in w_p} \|\boldsymbol{Y_k} - g(\boldsymbol{x})\|_2^2 \right) \tag{6.57}$$

Such optimization problem can be solved by using the standard algorithms such as `lsqnonlin` function from MATLAB. However, the `lsqnonlin` does not guarantee a global minimum, but a local one. As a consequence, we compare here the `lsqnonlin` function with the Trust-Region-Reflective Least Squares algorithm to the CMA-ES metaheuristic.

Trust-Region-Reflective Least Squares algorithm

In order to browse the solution space, the idea is to update the current point to $x + h$ if $f(x + h) < f(x)$. If $f(x+h) \geq f(x)$ then the region is shrunk and a new h is computed. This algorithm is based on an approximation of the function f to be minimized to a quadratic approximation. Such approximation can be defined with the first two terms of the Taylor approximation of f at x. Then, the problem is formulated as follow:

$$\min \left(\frac{1}{2} h^T H h + h^T g \quad \text{such that} \|Dh\|_2 < \Delta \right) \tag{6.58}$$

where H is the Hessian matrix, g the gradient at x, D is a diagonal scaling matrix and Δ a positive scalar.

Multi-level approach

In order to solve this system through an optimization problem, a multi-level approach has been used. The algorithm is first applied to a high level (low-resolution) image. Then, the result is used as input to the algorithm on the next lowest level. Therefore, this multi-resolution approach makes the window size defined previously varying automatically. Also, it provides an estimate to the next level. Of course, the result obtained at low resolution looks blurry, but this result is refined at each iteration, as shown in Fig. 6.11.
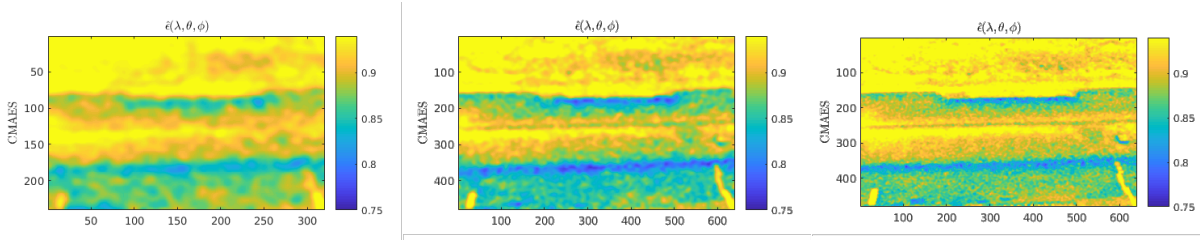
Figure 6.11: Multi-resolution approach for solving the optimization problem. From left to right: coarse level to highest final resolution details.

Results

Whereas the `lsqnonlin` can give a local optimum, the CMA-ES has the advantage of providing a global optimum, making it more robust. One other advantage of the CMA-ES is its quasi parameter-free aspect: only the initial solution point, the initial step-size and population size have to be provided. However such algorithm has a cost and the computation of the result is quite long and expensive, see Fig. 6.14.

Fig. 6.12 shows the comparison of the two algorithms. The first row shows the result for a constant emissivity over the entire image. The second one shows the result of the trust-region-reflective algorithm and the last one the result obtained with the CMA-ES algorithm. One can see that the result of the CMA-ES differs from the two others. Whereas in the `lsqnonlin`, a local solution is computed where the emissivity is almost constant, the CMA-ES tends to a more homogeneous temperature field. In this algorithm, the emissivity map makes more sense in terms of physics. In fact, the emissivity has slight changes in this image with the angle of view.

In particular, if we focus on the cuboid at the middle-right part of the scene, we can see that the CMA-ES and `lsqnonlin` emissivity take into account the angle of view, leading to a more homogeneous surface's temperature, as shown in the Fig. 6.13. An homogeneous temperature surface in this case makes more sense. However, no ground truth information on those data are available, so the final interpretation of this result is difficult. One can still note that the CMA-ES emissivity map is quite blurry, this detail could be improved in a post-processing step.

In order to perform a comparison as fair as possible, it is necessary to compare not only the result but also the processing time. Since the CMA-ES is a stochastic global optimization algorithm, the number of iteration is quite important compared to the trust reflective algorithm. Fig. 6.14 shows the processing time for each algorithm, depending on the number of pixels in the image, for a 3×3 window size. One can see that depending on image's size, the processing time is quite large for the CMA-ES algorithm. on a 41×36 image, the CMA-ES takes over 2 minutes and 40 seconds (just for this image, not for the entire multi-scale pyramid's process). For the full image (480×640), it represents more than 10 hours of process, compared to 1 hour and a half for the `lsqnonlin` function.
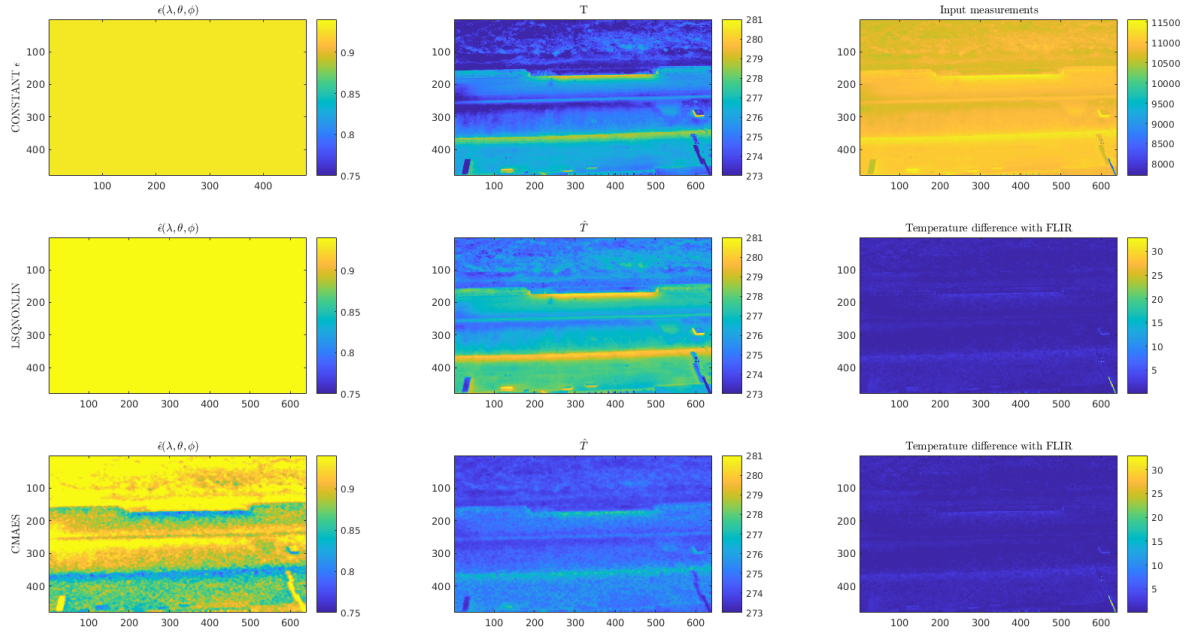
Figure 6.12: Comparison of the trust-region-reflective algorithm and the CMA-ES
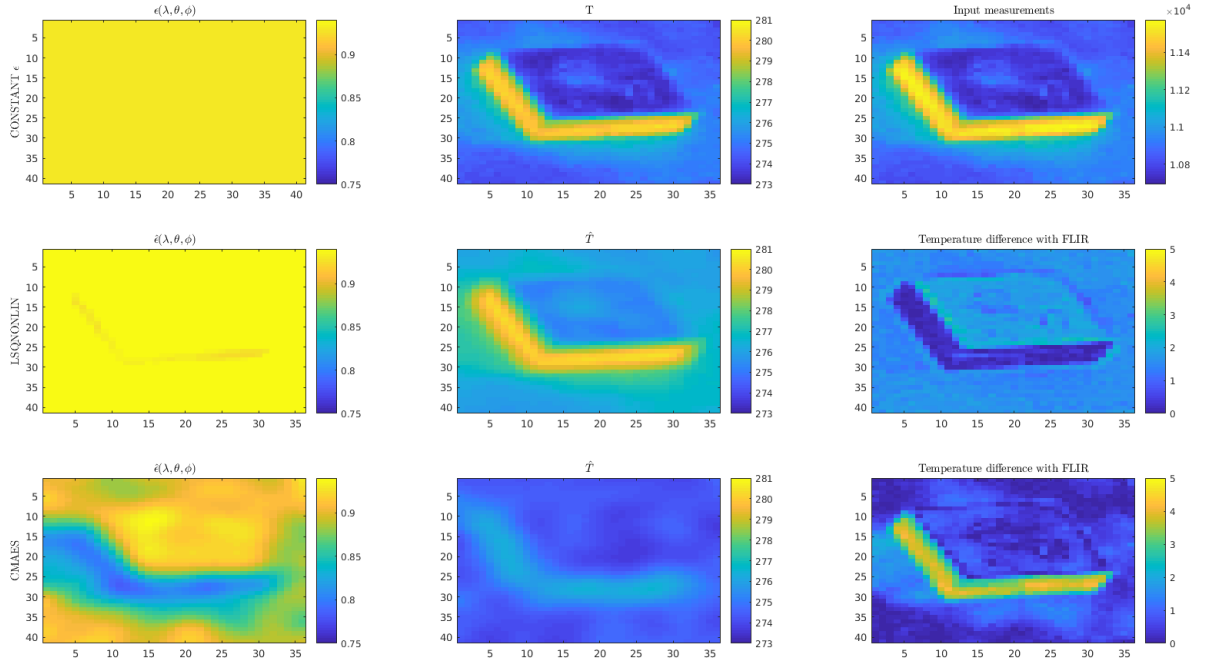

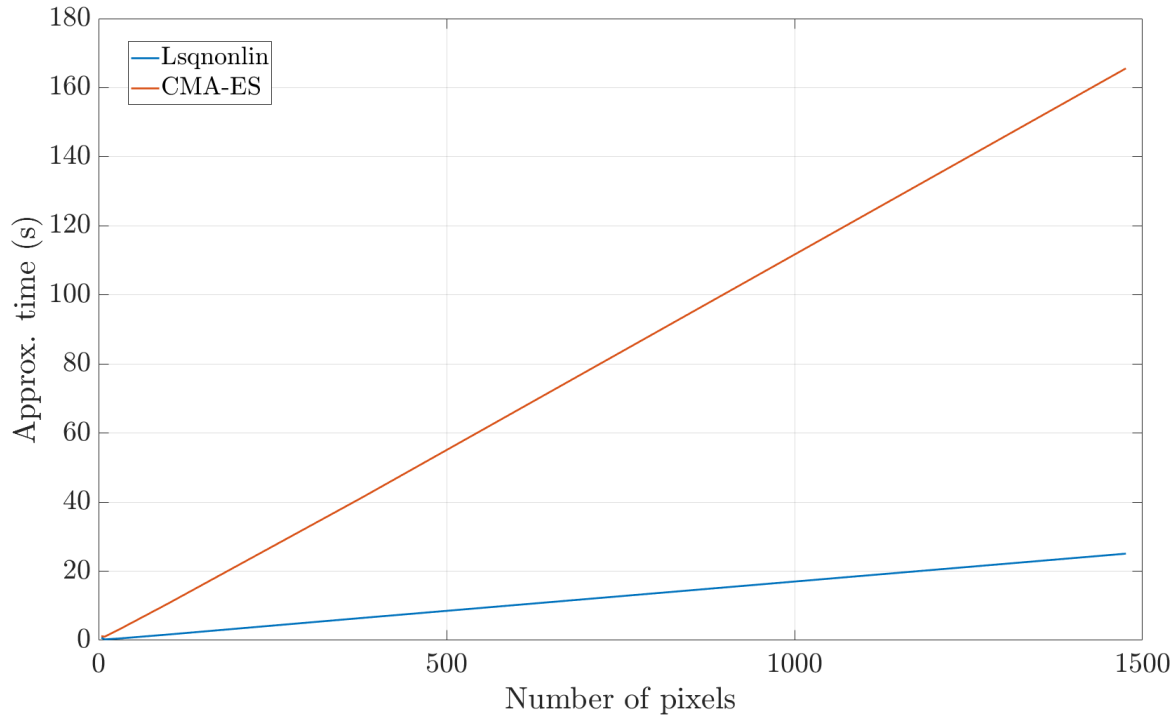Figure 6.13: Cuboid's result comparison

Figure 6.14: Comparison of processing time, for one given scale with a given number of pixels

6.7.3   Synthesis

An overview of different methods for the simultaneous emissivity and temperature estimation have been proposed. In the following section, those methods will be compared to a test case obtained by numerical simulation.

6.8   Results analysis

In order to test the developed methods, a study case presented in the next section has been set up. A comparison of the different introduced methods is performed: MCMC, CMA-ES and KIKF. Due to the fact that the IKF needs small spectral bandwidth, this method has not been tested but will be part of the final results summary proposed at the end of this section.

6.8.1   Study case

The comparison of the methods is done through the creation of a material target (see Fig. 6.15), similarly to colorimetric calibration. This material target is made of four materials with known radiative properties. By using the developed software presented in Chapter 5, infrared images acquisition has been simulated. In this case, a simple configuration has been chosen. The emissivity is assumed to

be constant over a given spectral band and the ambient environment is at $293.15K$. The materials are thin layers, directly controlled in temperature without any diffusion or convection phenomena.
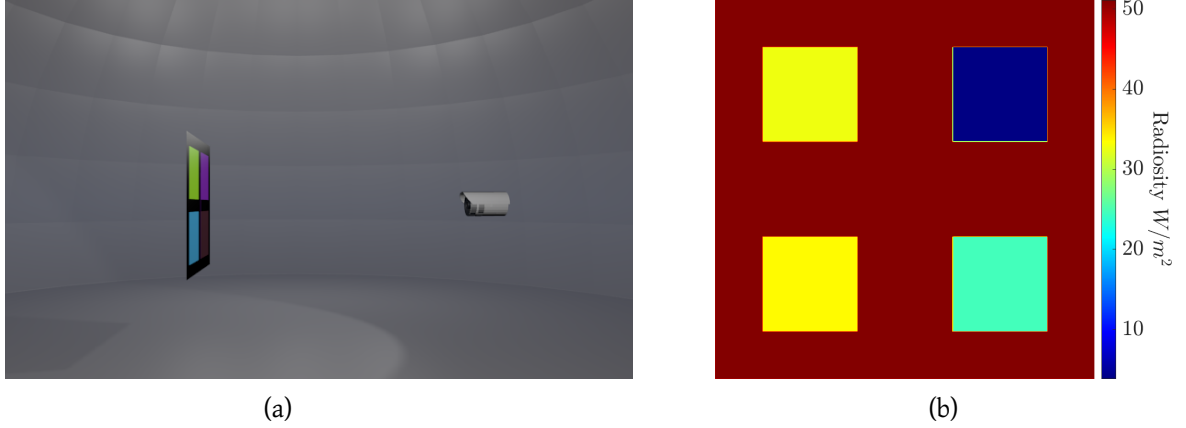

| (a) | (b) |

Figure 6.15: Target used for the study case. (a) Rendering of the scene in the visible spectrum. The camera is represented, the target and the hemisphere for the environment. (b) Result example for a frame at $20°C$ and other materials at $5.2°C$

The material properties are derived from the ECOSTRESS spectral library [14, 125] except from the alumina spectrum, derived from previous studies [130]. The four materials have been chosen for their particular emissivity spectral profile as shown in Fig. 6.16. The gold offers a reflective material, while black paint from which the frame is made of, an emissive one. The alumina is particularly interesting for its particular emissive window in the thermal IR band. Wood and concrete materials are commonly used in the civil engineering field. Five different spectral bands are used corresponding to possible infrared detectors spectral bands:

- Band 1: $1.5\mu m - 3\mu m$
- Band 2: $3\mu m - 5\mu m$
- Band 3: $8\mu m - 10\mu m$
- Band 4: $10\mu m - 12\mu m$
- Band 5: $8\mu m - 12\mu m$

In the proposed scenario, the frame temperature remains constant at $293.15K$. The temperature of the different materials is assumed to be controlled by a Peltier device and varies according to real temperature measurements at ground taken from the instrumented road section, presented in Chapter 4 and shown in Fig. 6.17. The temperature profile has been decimated to have a sampling rate every 16min.
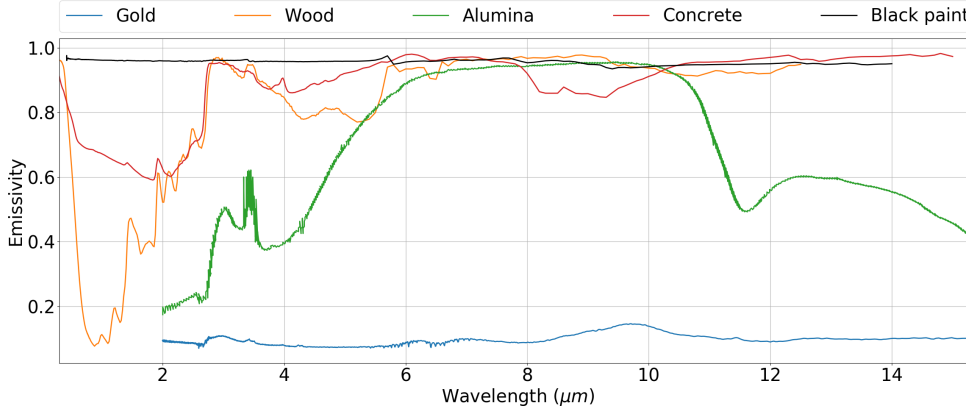
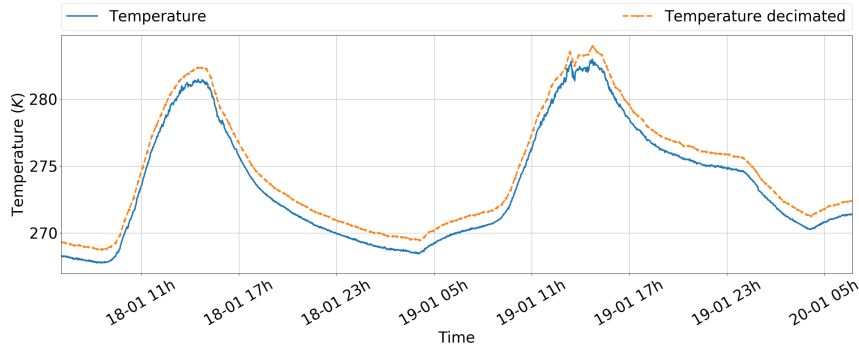Figure 6.16: Emissivity spectrum of the materials of the target.

Figure 6.17: Target material's temperature evolution through time. The measurements correpond to the temperature at ground during two days of january 2017. The decimated temperature profile is shifted by 1K for drawing purposes.

6.8.2 Monte-Carlo Markov Chain (MCMC)

The Monte-Carlo Markov Chain algorithm, as presented in Chapter 3 has been used on the simulated data. The prior used in this test for the emissivity is weak: $\mathcal{N}(0.5, 0.5)$. For the temperature, the initial prior has been chosen with $\mathcal{N}(273.15, 5)$ which is reasonable for practical applications where a first guess could be taken by using the air temperature for example. Fig. 6.18 and Fig. 6.19 show the result obtained with the algorithm after 10k iterations, for a single pixel value of each material for each spectral band. The histograms Fig. 6.19b, Fig. 6.19d, Fig. 6.19f and Fig. 6.19h show a shift of the mean in the posterior distribution of the temperature. The temperature and emissivity simultaneous estimation is satisfying on this test case. In fact, one can see that the emissivity spectral shape is retrieved and that the temperature mean tends to the actual temperature. This meta-heuristic provides interesting results on single-time spectral data, with weak priors. However, one drawback of the MCMC method is that it is difficult to know when the convergence has been reached. Moreover, the method is computationally intensive and may need many iterations to have a good estimate. The result has

been computed in approximately $4 \times 5 = 20\text{minute}$ on a `i7-6820HQ` CPU @ 2.70GHz on $\text{MATLAB}^{®}$), for a single pixel at a single timestamp. As a consequence, only the temperature and emissivity of one pixel at one time for each of the material have been estimated. The implementation could be parallelized through a GPGPU approach to reduce the computation time. Nevertheless, even with a parallel implementation the time needed for a full resolution image over multiple days would take a long time. Instead, the method could be improved. For instance, a change of the proposal distribution and / or going onto an Ensemble MCMC method or other more advanced methods could be investigated to get better results and faster convergence [59].

(a) Gold material

(b) Alumina material.
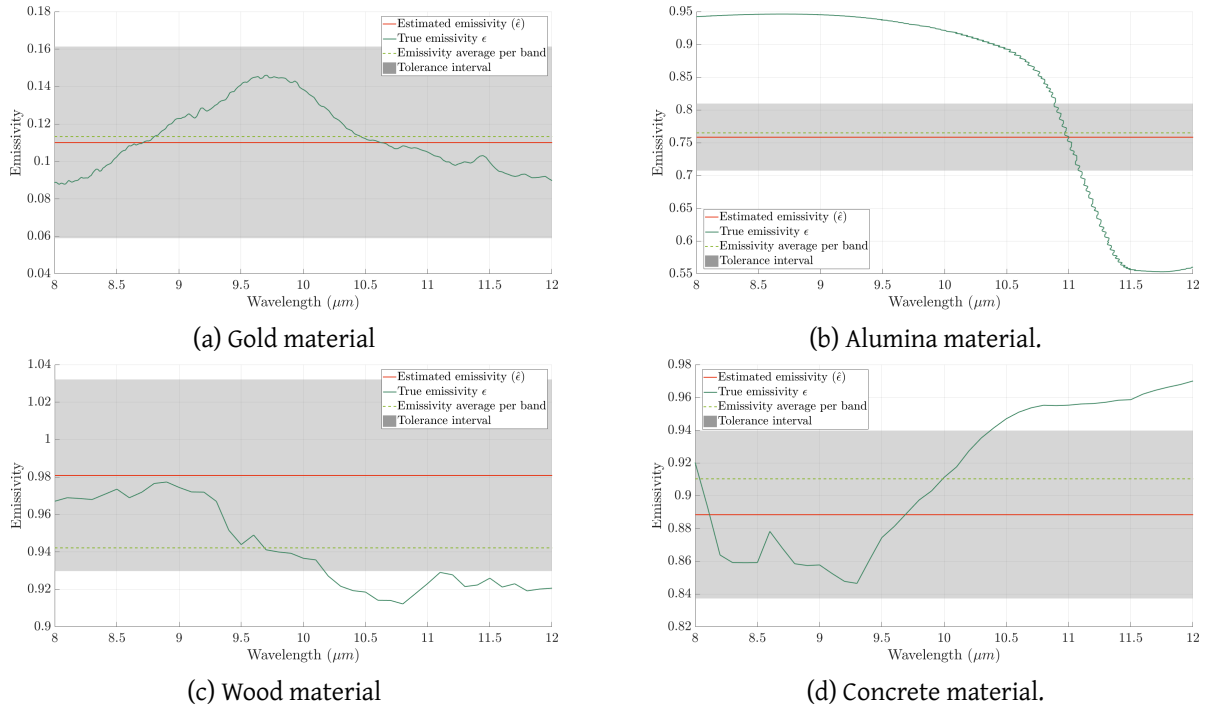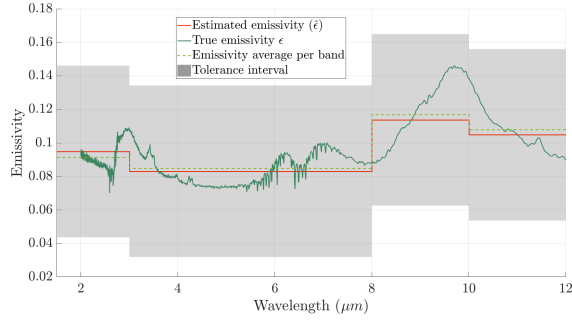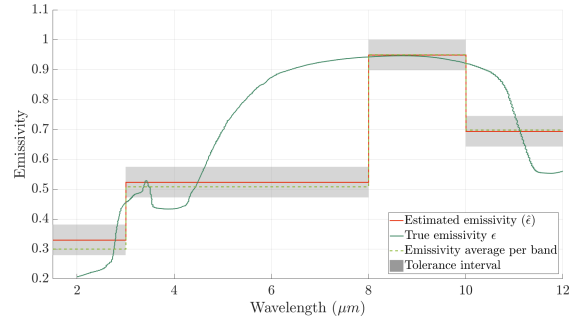
(c) Wood material

(d) Concrete material.

Figure 6.18: Focus on the emissivity retrieval on one pixel of each of the four materials for the $8\mu\text{m} - 12\mu\text{m}$ spectral band.
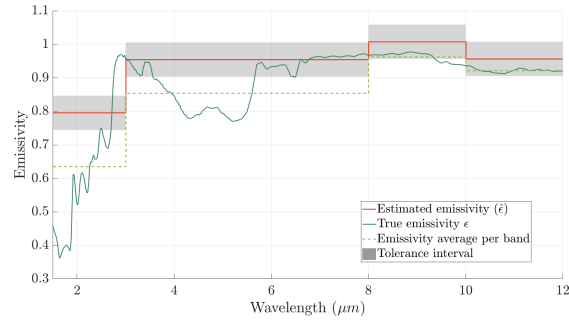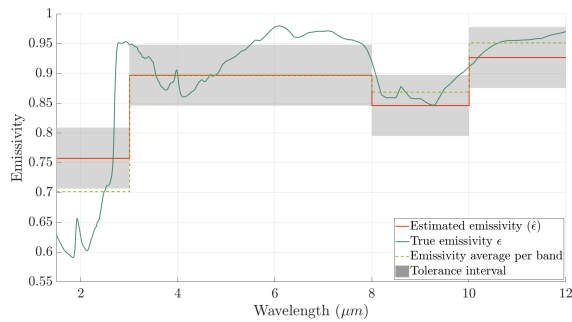
(a) Gold material
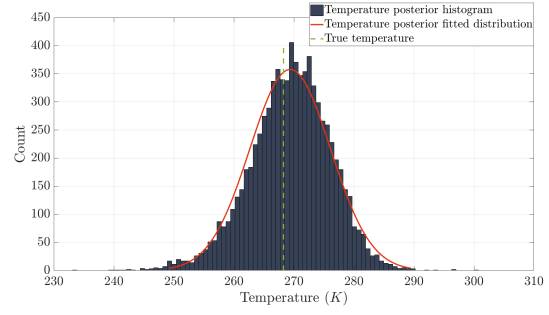
(b) Gold temperature posterior histogram.
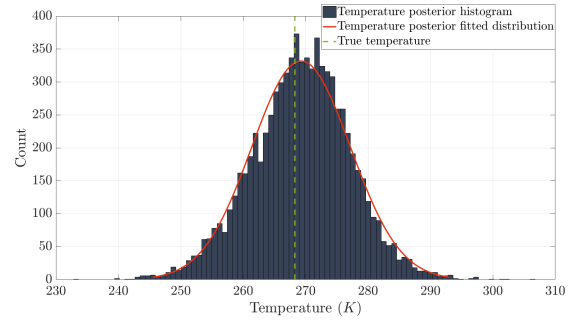
(c) Alumina material

(d) Alumine temperature posterior histogram.
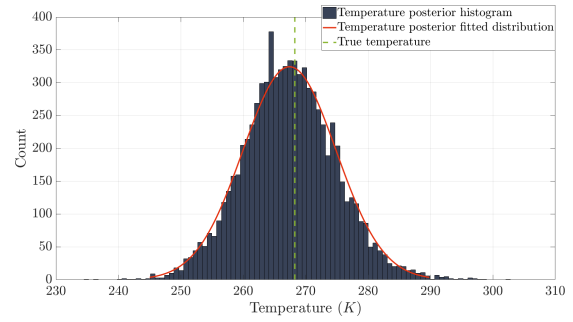
(e) Wood material

(f) Wood temperature posterior histogram.

(g) Concrete material
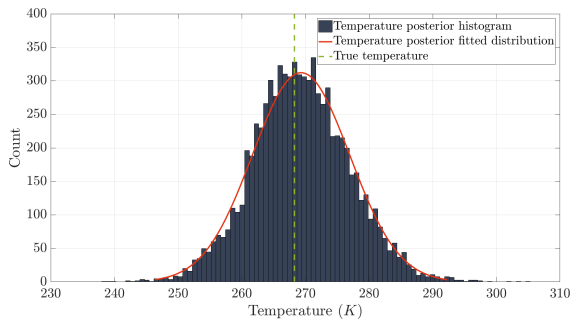
(h) Concrete temperature posterior histogram.

Figure 6.19: Results of the temperature / emissivity retrieval on one pixel of each of the four materials. (a) (c) (e) (g): Emissivity retrieval. The ground truth profile (continous) is compared to the estimated one (per spectral band).(b) (d) (f) (h): Temperature estimation. The ground truth temperature is represented by the dashed line.

6.8.3 CMA-ES

Then, the CMA-ES described in this chapter has been tested on the simulated data. As explained previously, the temperature is assumed to be constant over a given pixel's neighborhood. Such assumption can be discussed. In fact, at thermal equilibrium, for a short distance and small angle of view this assumption remains valid. However, when important distances are involved, a given pixel's represents a larger area. In that case, two pixels may have very different temperature. The method has been applied over the entire image at one timestamp and the results are presented for the band $(10\mu m - 12\mu m)$. Boundaries are applied to the optimization problem: $0 < \epsilon < 1$ and $260.15K < T < 300.15K$.

An initial emissivity constant value is given as a starting point of the optimization algorithm. This first guess shown in Fig. 6.20 leads to important difference between initial temperature and ground truth temperature.
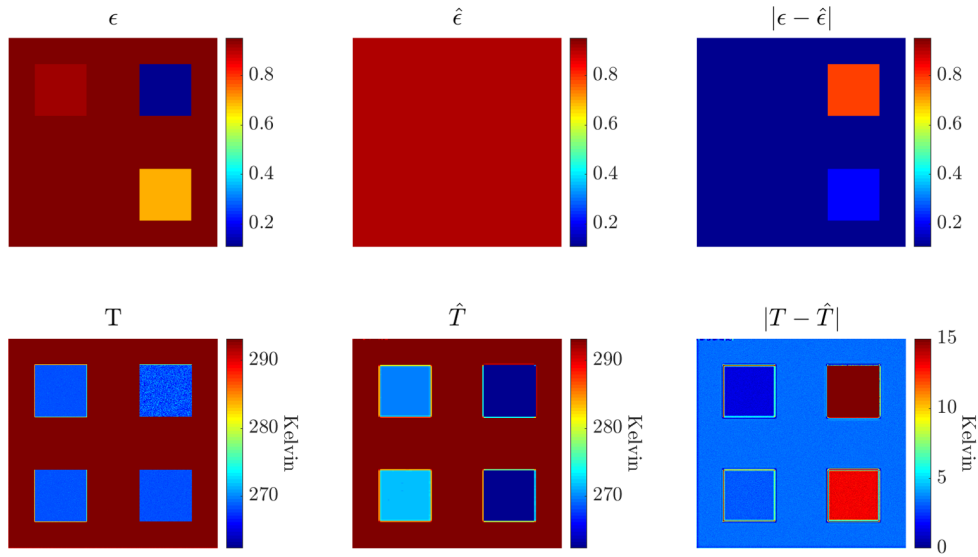
Figure 6.20: Initial prior for the CMA-ES optimization algorithm. A constant emissivity is chosen leading to important differences with the ground truth values. Band: $10\mu m - 12\mu m$.

The result at level 3 is shown in Fig. 6.21. The emissivity map starts to appear in this coarse preliminary result. However, the temperature estimation suffers from the coarse approximation, due to the assumption that it is constant over a given neighborhood.
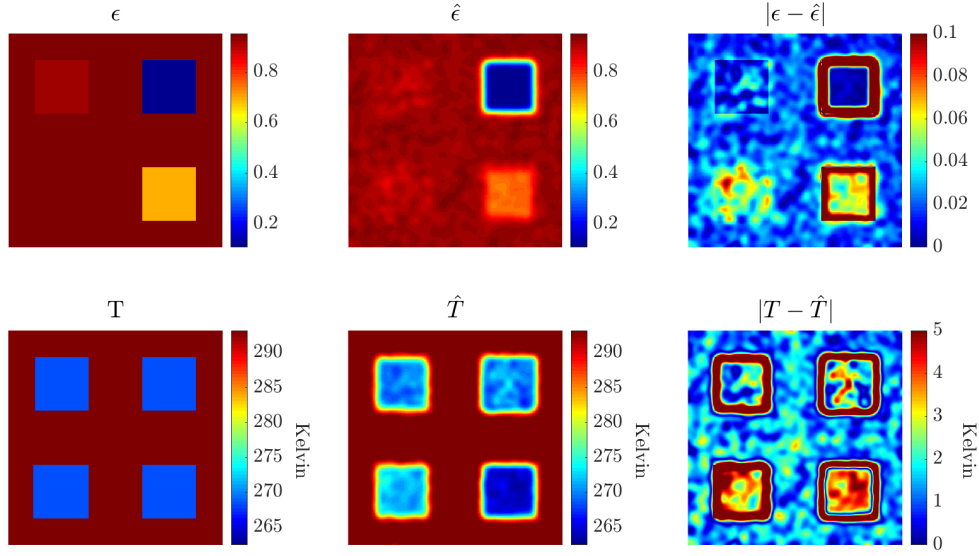
Figure 6.21: Coarse level preliminary CMAES result. The image is blurry due to the level used. Band: $10\mu m - 12\mu m$.

Fig. 6.22 shows the final result at fine resolution. Due to the window size of 5×5 pixels, the result looks noisy. This noise could be removed either by increment the window size to an extent that would not compromise the results or, as done here, by a post-process filtering (median filter).
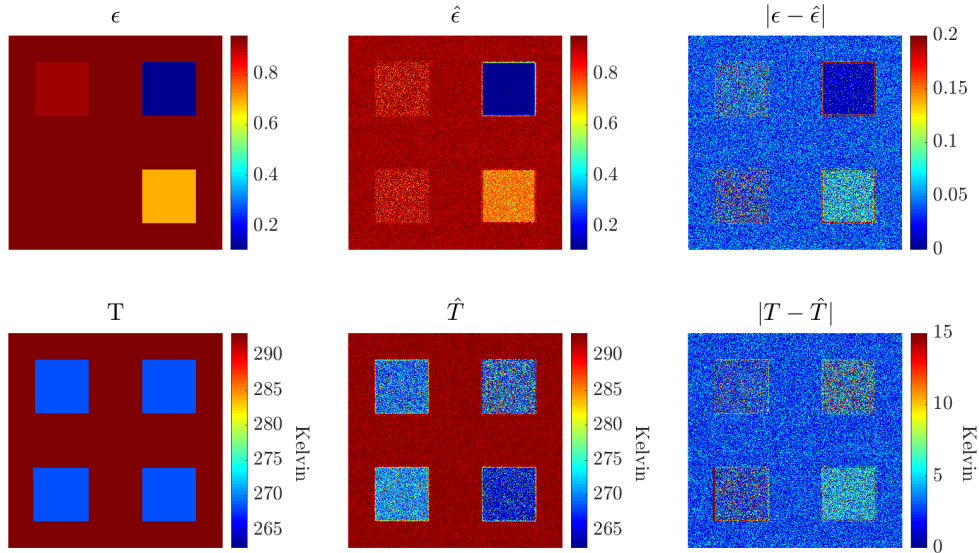
Figure 6.22: Full result of the CMAES algorithm at a given time and given band. At the opposite of Fig. 6.21, the image is noisy due to the small neighborhood involved. Band: $10\mu m - 12\mu m$.

The post process of Fig. 6.22 is shown in Fig. 6.23.
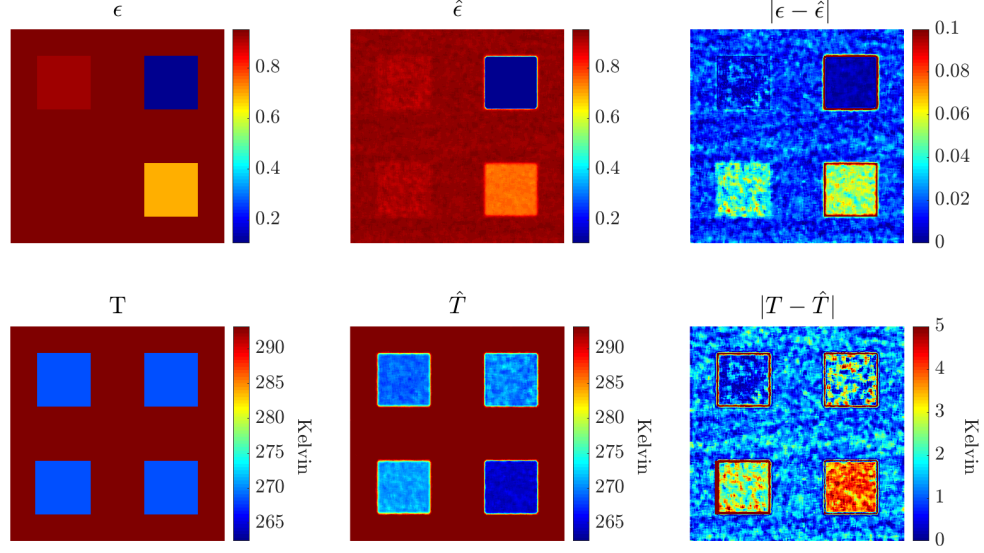
Figure 6.23: Filtered CMAES result after a 2D median filter. The materials have distinct emissivities values on the emissivity map. As stated in Chapter 4 and observed on the temperature absolute difference image, a small error on low emissivity values can lead to important error in the temperature estimate. Band: $10\mu m - 12\mu m$.

Two metrics have been used to compare the result given by the CMAES method with the ground truth images. The first one is the root-mean-square error (RMSE) that evaluates the distance between the estimated values and the expected ones. The RMSE is performed locally on a given window of 5×5 pixels. The second one is called the structural similarity (SSIM) which aims at measuring the similarity between two images, developed for image reconstruction and compression. The SSIM index is defined between two window u and v as:

$$\text{SSIM}_{\text{index}}(u, v) = \frac{\left(2\mu_u\mu_v + c_1\right)\left(2\sigma_u\sigma_v + c_2\right)\left(2\text{cov}_{uv} + c_3\right)}{\left(\mu_u^2 + \mu_v^2 + c_1\right)\left(\sigma_u^2 + \sigma_v^2 + c_2\right)\left(\sigma_u\sigma_v + c_3\right)} \tag{6.59}$$

where

- μ_u is the mean of u,
- μ_v is the mean of v,
- σ_u is the variance of u,
- σ_v is the variance of v,
- cov_{uv} is the covariance of u and v,
- c_1, c_2 and c_3 are constants that aim to characterize the saturation effects and ensure numerical stability for denominators close to zero.

In this study, a SSIM SSIM$_m$ metrics is used instead of the SSIM index, for its mathematical properties [22]:

$$\text{SSIM} = \sqrt{1 - \text{SSIM}_{\text{index}}} \tag{6.60}$$

Finally, Fig. 6.24 shows the comparison of ground truth values to estimated ones. As shown in Fig. 6.23, the different emissivity materials levels are found. However, some differences exist between the expected value and the estimated one. By looking at the RMSE of the emissivity, an important gap exists between the bottom right material (alumina) emissivity expected and estimated values. However, the SSIM shows that this estimated solution is found over the entire material and that the spatial structure of the emissivity map corresponds to the expected values.
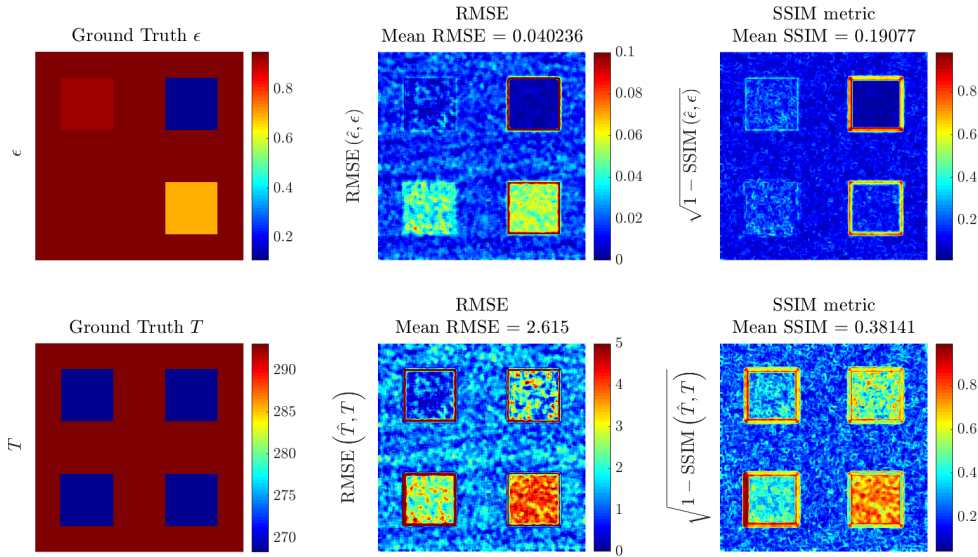
Figure 6.24: CMAES: SSIM and RMSE local values between estimated emissivity and temperature and ground truth values. A small error on materials with low emissivity leads to a more important error in temperature than materials with higher emissivity. The material average RMSE is up to 4K against 0.05 for the emissivity, which is still lower in average than for a constant emissivity case. Band: 10μm − 12μm.

As stated in Chapter 4, a small error on a small emissivity value will lead to an important one on the temperature. Such behaviour is shown with the gold material (top right material). Even if the RMSE and SSIM have low values, the final RMSE and SSIM on the temperature are more important. As a comparison to the MCMC method, the CMA-ES provides interesting results due to its constant local temperature assumption only. No other priors are given. In the current implementation (MATLAB®), the full image is processed in approximately 10h. As a consequence, the CMA-ES seems to be a good way to get an emissivity map of the scene without any knowledge. As shown in Chapter 4, an emissivity map can reduce the temperature estimation error drastically. Another possibility for this algorithm is the

possibility to identify the different materials within the scene, for classification purposes for example. Moreover, the algorithm could be improved by using the multi-band measurements, which has not been done yet. Such improvement could lead to disambiguate the solutions and therefore improve the emissivity estimation. In any case, the estimated temperature error is still lower in average than in a constant emissivity case as in Fig. 6.20.

6.8.4 KIKF

Finally, the developed KIKF method has been tested on the study case. However, some adjustments need to be performed in order to have a fair comparison with the other algorithms. First, in the simulation, the materials are assumed to be heated by Peltier modules whereas the frame is at constant temperature of $20°C$. One of the assumption of the KIKF is to suppose that the scene temperature is heated by the sun which does not apply here. As a consequence, the references locations (denoted previously by s) are taken within the materials areas, outside the target's frame. The initial temperature is taken at $T_0 = 270$K. Fig. 6.25 shows the result of the first kriging step, with this first T_0. The first image in this figure shows the estimated emissivity at T_0 and the second and third one the kriging interpolation from this emissivity map and the references points.
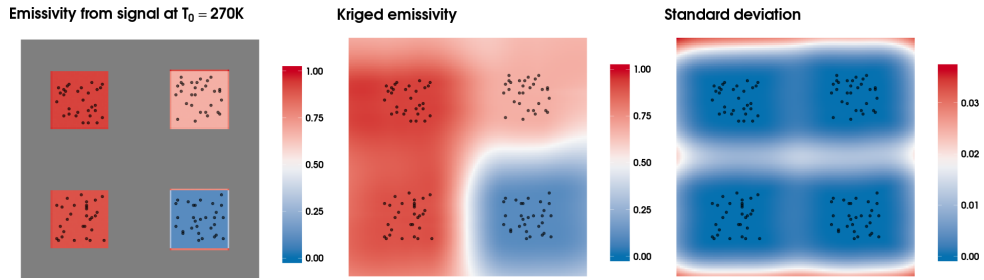
Figure 6.25: Initial estimate by using a constant temperature value $T = T_0 = 270$K. The emissivity at T_0 is shown in the first image. The Kriged model result from this emissivity map is shown in the two other images with the mean estimate and standard deviation. Band: $10\mu m - 12\mu m$.

Then, from this first estimate, the KIKF is run as new data are incoming. During the morning (corresponding to $A_k < 1$), the KIKF filter is applied while a simple Kalman filter is applied during the rest of the day, with the last emissivity approximation. The final estimates are shown in Fig. 6.26. For one given time, the estimation of the emissivity can be more or less accurate. Such behavior could be improved by modifying the particle filter to mutate accordingly to other particles solutions, similarly to particle swarm optimization algorithms. Moreover, it appears that the estimation is more difficult for reflective materials which is an expected result in reality, particularly when dealing with direct solar illumination (not considered here). Fig. 6.27 is an example of one pixel's time evolution. Again, depending on the selected pixel, the estimation will be more or less accurate. In any case, the filter provides information about how the estimates can be trusted: if the estimated temperature is not exact,

the error made is at least quantifiable.
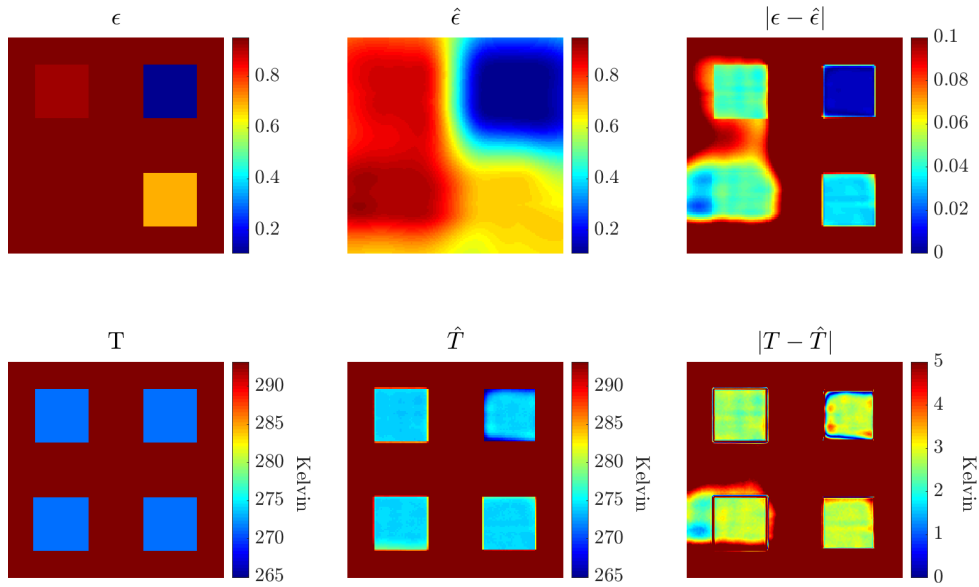
Figure 6.26: Final estimate over the entire period. The ground truth values are compared to the estimates. As for the CMA-ES the materials emissivities are distincts. Band: $10\mu m - 12\mu m$.

Figure 6.27: Estimated temperature time evolution at one pixel

174

As for the CMA-ES method, the SSIM and RMSE are shown in Fig. 6.28. As a comparison, the results errors are more homogeneous in the KIKF method, with a RMSE around 2.5K in this case. However, this result is conditioned by the estimate found with the IKF. As a consequence, it may be better or worst depending on the time evolution of the temperature (see Fig. 6.27).
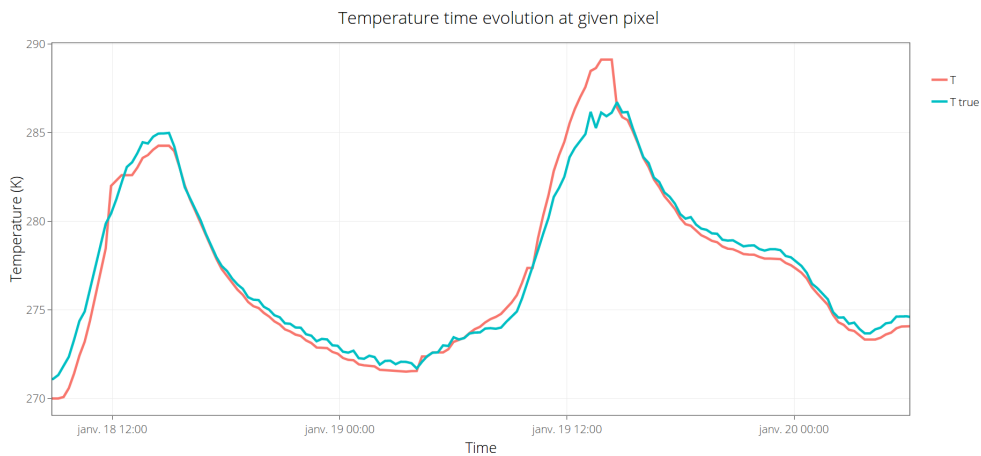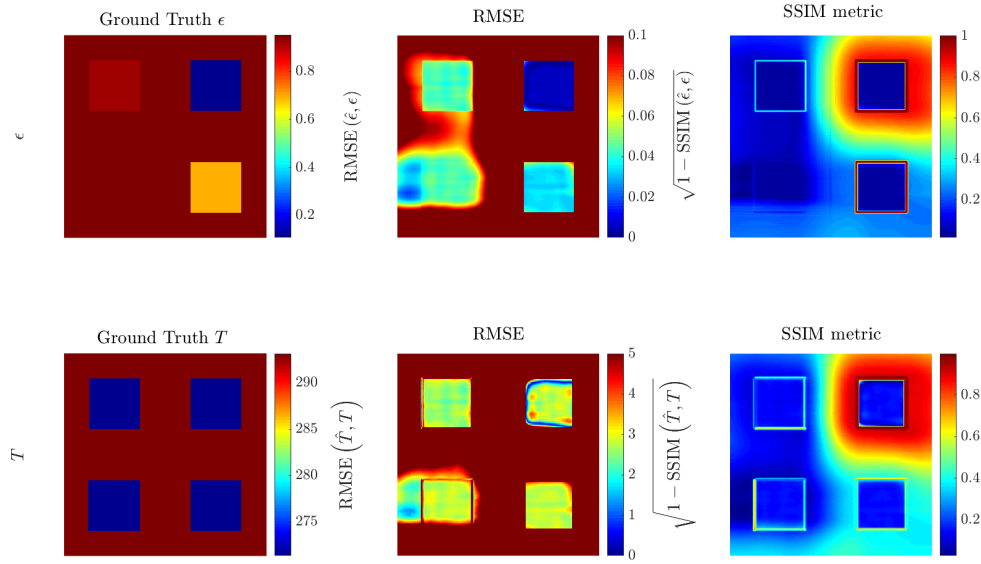

Figure 6.28: KIKF: SSIM and RMSE local values between estimated emissivity and temperature and ground truth values. A small error on materials with low emissivity leads to a more important error in temperature than materials with higher emissivity. The material average RMSE is less than 3K against 0.04 for the emissivity, which is still lower in average than for a constant emissivity case. Band: $10\mu m - 12\mu m$.

This method provides an interesting option to the previous meta-heuristics. In fact, the model is able to track in time the evolution of the system's parameters and could, at term, indicates of a change of properties in the observed structure. Such change can be noticed by the Kriging prediction that will be compromised. A reset factor could be introduced to catch-up any change. Thanks to the Kriging, only part of the pixels are tracked through time, reducing the data storage and computation time. The current implementation in R has estimated the results in 10min for two days of single-band data over the entire image. Moreover, the method is designed for a single-band camera. Introducing the other bands could, as for the CMA-ES disambiguate the solutions. Furthermore, the Kriging model has been put on the pixel spatial covariance. In Chapter 4, the spatial calibration process has been introduced. This calibration process could be used to get a spatial covariance model based on the world scale and not the image scale anymore which would lead to a more physical meaning. The KIKF method could be extended to take into account the previous remarks but more particularly to improve its robustness. In fact, the current method needs to set-up the parameters of the Kalman filter, the particle filter and the Kriging which is not always trivial.

6.8.5   Synthesis

In this thesis, methods have been studied and developed to estimate conjointly the temperature and emissivity for *in-situ* infrared multi-spectral measurements. Bayesian methods offer a manor of including prior information that could be used when additional knowledge is available such as emissivity material specific prior [10]. Furthermore, Bayesian methods offer a way to measure the uncertainty made on the estimation which is valuable for practical applications. Table 6.1 shows a summary of the physical assumptions and the application domain of the algorithms.

Method	Physical hypothesis	Application
MCMC	ϵ and T independent	Per pixel Single-band Single sample
CMA-ES	ϵ and T independent T homogeneous on a given neighborhood	Per pixel Single-band Single sample
IKF	ϵ and T independent Small bandwidth	Per pixel Multi-band only Needs temporal data
KIKF	ϵ and T independent T_0 at thermal equilibrium T homogeneous on a given neighborhood ϵ time evolution slow compared to T	Full image Single-band Needs temporal data

Table 6.1: Summary of the compared simultaneous emissivity and temperature methods: physical assumptions

If the MCMC and CMA-ES methods provide interesting results, the computation energy and time needed are an obstacle to a practical long-term monitoring solution. At the opposite, the KIKF method proposes a time tracking solution suitable for long-term thermal monitoring of infrastructures. However, this method needs to be deepened to be more robust and easier to setup. Table 6.2 summarizes the method's pros and cons.

Method	Pros	Cons
MCMC	Weak assumption Can be parallelized	Computationally intensive No temporal tracking
CMA-ES	Faster than MCMC Coarse approximation possible	Computationally intensive No temporal tracking
IKF	Faster than the former T tracked through time	Bandwidth assumption Multi-spectral only Temporal data needed
KIKF	Faster than the former T tracked through time ϵ covariance tracking through time	Assumption on T_0 Temporal data needed

Table 6.2: Summary of the compared simultaneous emissivity and temperature methods: pros and cons

Finally, in order to improve the estimation process, one idea is to use hybrid methods by combining for example the CMA-ES and KIKF. The CMA-ES could provide a first estimate to the KIKF method that would track dynamically the evolution of the system's parameters. When the KIKF tracking is made difficult, the CMA-ES could be called again.

Conclusion and future work

The work done in this thesis allowed to review the infrared thermography for *in-situ* infrastructures long-term thermal monitoring. The necessity and advantages of infrared thermography on a structural health monitoring context has been presented. In order to understand fully the measurement process of infrared thermography, the different equations and quantities used in the radiative transfer theory have been introduced as well. Then, needs for emissivity and temperature separation methods have been emphasized based on those equations.

A bibliographical study has been driven. The temperature and emissivity separation methods from the literature and particularly in the Remote Sensing field have been listed and analysed first. Secondly, with an aim of developing a 3D *in-situ* infrared scene simulator, the main algorithms of the image synthesis field have been described. Based on the study of literature's algorithms, it has been shown that the progressive radiosity method allows an iterative resolution of the radiative equation system that respect the physical equations.

As a consequence, an infrared simulation tool that uses the progressive radiosity method has been developed. This tool uses the latest computer graphics technologies and has a graphical interface to visualize the 3D scene and the rendering. Based on image synthesis and derived from the thermal field, the progressive radiosity method solves the radiosity equation in a computationally efficient way. Due to the fact that all the surfaces emit in the infrared spectrum, the algorithm has been parallelized and more particularly the shooting step through a GPGPU approach. The software makes use of the graphical libraries and hardware algorithms to render the scene. In addition, a simple model of atmospheric transfer has been transcribed in the software to simulate the spectral solar irradiance at ground. Furthermore, a camera and detector models are used. Please note that the camera and detector implementations are parallelized and rendered on the GPU in real-time by benefiting from hardware acceleration. However, the detector model is based on quantum detectors and should be modified to other type of detector so that it would actually fit the infrared thermography measurement process. Moreover, the camera model optic does not consider the optical aberrations yet. A transfer function of the camera could therefore be added.

A sensitivity analysis of the parameters that influence the infrared thermography measurement has also been conducted. In particular, the use of coupled measurements refines the radiative model and helps at reducing the uncertainty on the temperature estimate. If for some reason sensors and local data are not available, it is possible to retrieve those informations online thanks to recent open

databases. Especially, the use of the European Copernicus climate data store and their API coupled with empirical models from the literature provides a way to estimate most of the parameters of the equation. Moreover, a tool has been developed to visualize and process the large amount of data gathered by long-term measurements. This tool and its graphical interface helps users to get an overview of the spatial, spectral and time data. To improve the display of the images, the radiative conversion process from digital levels to temperatures have been implemented onto the graphical processing unit (GPU). Some extensions could be added to this software. At the present time, the data from online sources are firstly saved onto files and then read and parsed *via* the software. However, it would be interesting to incorporate the gathering of online data from different sources directly onto the software's interface. Furthermore, a batch processing tool that allows the process of many data at once in a simplified configuration file was developed. This tool is helpful for repeating tasks on different data and could be generalized to any other application.

Finally, new methods for estimating simultaneously the temperature and emissivity have been presented. The Monte-Carlo Markov chains method from the literature has been implemented. By using statistical priors on the temperature and emissivity distributions, the simultaneous estimation of temperature and emissivity is performed, pixel by pixel, at every moment. If this method provides satisfying results, its current implementation suffers from a long processing time. However, improvements could be made by using Ensemble MCMC for example and some parallelization process. A second method based on an interacting Kalman filter has been proposed. It enables the tracking of the temperature through time and the estimation of the emissivity. However, this estimation relies on a hypothesis of small bandwidth which is suitable for hyper-spectral applications only. Then, the CMA-ES optimization method has been used, relying on a local homogeneous hypothesis. As for the MCMC method, if CMA-ES provides interesting results, its computation is intensive. To overcome those issues, a Kriged Interacting Kalman filter (KIKF) has been developed. By using a prior on the spatial emissivity distribution, an interacting kalman filter tracks the temperature and estimates the system parameters. Preliminary results are interesting: on simulation data, the temperature profile is retrieved and the emissivity map values restoration satisfying even if differences with actual values exist. The algorithm needs to be improved to be more reliable, particularly in setting the different filters parameters. Moreover, tests on laboratory with real measurements data and then on-field measurements need to be performed to fully validate the method.

Bibliography

[1] *A Catalog of Radiation Heat Transfer Configuration Factors*, http://www.thermalradiation.net/indexCat.html.

[2] *A Generalized Split-Window Algorithm for Retrieving Land-Surface Temperature from Space - IEEE Journals & Magazine*, https://ieeexplore.ieee.org/abstract/document/508406/.

[3] *A Temperature and Emissivity Separation Algorithm for Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) Images - IEEE Journals & Magazine*, http://ieeexplore.ieee.org/document/700995/?reload=true.

[4] L. Adelard et al., "Sky Temperature Modelisation and Applications in Building Simulation", en, *in*: *Renewable Energy* 15 (Sept. 1998), pp. 418–430, DOI: 10.1016/s0960-1481(98)00198-0.

[5] F. Aires, "Remote Sensing from the Infrared Atmospheric Sounding Interferometer Instrument 2. Simultaneous Retrieval of Temperature, Water Vapor, and Ozone Atmospheric Profiles", en, *in*: *Journal of Geophysical Research* 107.*D22* (2002), ISSN: 0148-0227, DOI: 10.1029/2001JD001591.

[6] F. Aires et al., "A New Neural Network Approach Including First Guess for Retrieval of Atmospheric Water Vapor, Cloud Liquid Water Path, Surface Temperature, and Emissivities over Land from Satellite Microwave Observations", en, *in*: *Journal of Geophysical Research: Atmospheres* 106.*D14* (2001), pp. 14887–14907, ISSN: 2156-2202, DOI: 10.1029/2001JD900085.

[7] "AN EMPIRICAL EVALUATION OF A GPU RADIOSITY SOLVER:" en, *in*: *Proceedings of the International Conference on Computer Graphics Theory and Applications*, Angers, France: SciTePress - Science and and Technology Publications, 2010, pp. 225–232, ISBN: 978-989-674-026-9, DOI: 10.5220/0002822202250232.

[8] Arthur Appel, "Some Techniques for Shading Machine Renderings of Solids.", *in*: *AFIPS Conference Proc.* 1968.

[9] D. Arquès and S. Michelin, "Proximity Radiosity : Exploiting Coherence to Accelerate Form Factor Computations", en, *in*: *Rendering Techniques '96*, ed. by Xavier Pueyo and Peter Schröder, Eurographics, Springer Vienna, 1996, pp. 143–152, ISBN: 978-3-7091-7484-5.

[10] Joshua N. Ash and Joseph Meola, "Temperature-Emissivity Separation for LWIR Sensing Using MCMC", *in*: *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXII*, vol. 9840, International Society for Optics and Photonics, May 2016, 98401O, DOI: 10.1117/12.2223263.

[11] *ASSIMP - The Open-Asset-Importer-Lib*, http://www.assimp.org/.

[12]   "Atmospheric Radiative Transfer Codes", en, *in*: *Wikipedia* (Jan. 2019), Page Version ID: 881103099.

[13]   Daniel Balageas, "Introduction to Structural Health Monitoring", en, *in*: *Structural Health Monitoring*, John Wiley & Sons, Ltd, 2010, pp. 13–43, ISBN: 978-0-470-61207-1, DOI: 10.1002/9780470612071.ch1.

[14]   A.M. Baldridge et al., "The ASTER Spectral Library Version 2.0", en, *in*: *Remote Sensing of Environment* 113.*4* (Apr. 2009), pp. 711–715, ISSN: 00344257, DOI: 10.1016/j.rse.2008.11.007.

[15]   Alessandro Barducci and Ivan Pippi, "Temperature and Emissivity Retrieval from Remotely Sensed Images Using the" Grey Body Emissivity" Method", *in*: *IEEE Transactions on Geoscience and Remote Sensing* 34.*3* (1996), pp. 681–695.

[16]   François Becker and Zhao-Liang Li, "Surface Temperature and Emissivity at Various Scales: Definition, Measurement and Related Problems", en, *in*: *Remote Sensing Reviews* 12.*3-4* (Jan. 1995), pp. 225–253, ISSN: 0275-7257, DOI: 10.1080/02757259509532286.

[17]   François Becker and Zhao-Liang Li, "Temperature-Independent Spectral Indices in Thermal Infrared Bands", *in*: *Remote Sensing of Environment* 32.*1* (Apr. 1990), pp. 17–33, ISSN: 0034-4257, DOI: 10.1016/0034-4257(90)90095-4.

[18]   François Becker and Zhao-Liang Li, "Towards a Local Split Window Method over Land Surfaces", en, *in*: *International Journal of Remote Sensing* 11.*3* (Mar. 1990), pp. 369–393, ISSN: 0143-1161, 1366-5901, DOI: 10.1080/01431169008955028.

[19]   Alexander Berk, Lawrence S. Bernstein, and David C. Robertson, *MODTRAN: A Moderate Resolution Model for LOWTRAN*, en, tech. rep. SSI-TR-124, SPECTRAL SCIENCES INC BURLINGTON MA, July 1987.

[20]   Candace Berrett et al., "A Bayesian Nonparametric Model for Temperature-Emissivity Separation of Long-Wave Hyperspectral Images", *in*: *Technometrics* 56.*2* (Apr. 2014), pp. 200–211, ISSN: 0040-1706, DOI: 10.1080/00401706.2013.869262.

[21]   Craig F. Bohren and Donald R. Huffman, *Absorption and Scattering of Light by Small Particles*, Wiley-VCH, 1998.

[22]   D. Brunet, E. R. Vrscay, and Zhou Wang, "On the Mathematical Properties of the Structural Similarity Index", en, *in*: *IEEE Transactions on Image Processing* 21.*4* (Apr. 2012), pp. 1488–1499, ISSN: 1057-7149, 1941-0042, DOI: 10.1109/TIP.2011.2173206.

[23]   James J. Buglia, *Introduction to the Theory of Atmsopheric Radiative Transfer*, Studies in Geophysical Optics and Remote Sensing, Hampton, Va., USA: A. Deepak, 1985, ISBN: 978-0-937194-05-8.

[24]   *.:: C++ Mathematical Expression Library (ExprTk) - By Arash Partow ::.* http://www.partow.net/programming/exprtk/.

[25] Eva Cerezo et al., "A Survey on Participating Media Rendering Techniques", en, *in*: *The Visual Computer* 21.*5* (June 2005), pp. 303–328, ISSN: 0178-2789, 1432-2315, DOI: `10.1007/s00371-005-0287-1`.

[26] S. Chandrasekhar, *Radiative Transfer*, Dover Publications Inc., 1960, ISBN: 978-0-486-60590-6.

[27] Y. May Chang, "An Investigation Into The Factors Affecting Infrared Temperature Measurements For Building Applications", *in*: *SPIE* 780.*Thermosense IX* (1987), pp. 11–18.

[28] Changchang Wu et al., "3D Model Matching with Viewpoint-Invariant Patches (VIP)", *in*: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8, DOI: `10.1109/CVPR.2008.4587501`.

[29] Changchang Wu et al., "3D Model Search and Pose Estimation from Single Images Using VIP Features", *in*: *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2008, pp. 1–8, DOI: `10.1109/CVPRW.2008.4563037`.

[30] Xiang Chen et al., "The Mixed Pixel Effect in Land Surface Phenology: A Simulation Study", *in*: *Remote Sensing of Environment* 211 (June 2018), pp. 338–344, ISSN: 0034-4257, DOI: `10.1016/j.rse.2018.04.030`.

[31] Jianye Ching, James L. Beck, and Keith A. Porter, "Bayesian State and Parameter Estimation of Uncertain Dynamical Systems", *in*: *Probabilistic Engineering Mechanics* 21.*1* (Jan. 2006), pp. 81–96, ISSN: 0266-8920, DOI: `10.1016/j.probengmech.2005.08.003`.

[32] Alexandre Adrien Chotard, Anne Auger, and Nikolaus Hansen, "Cumulative Step-Size Adaptation on Linear Functions: Technical Report", *in*: *arXiv:1206.1208 [cs]* (June 2012), arXiv: `1206.1208 [cs]`.

[33] Per H Christensen, "Point-Based Approximate Color Bleeding", en, *in*: *Annecy Animation Film Festival*, Annecy, France, June 2019, p. 9.

[34] K. Chrzanowski, "Comparison of Shortwave and Longwave Measuring Thermal-Imaging Systems", *in*: *Applied optics* 34.*16* (1995), pp. 2888–2897.

[35] K. Chrzanowski, "Influence of Measurement Conditions and System Parameters on Accuracy of Remote Temperature Measurement with Dualspectral IR Systems", *in*: *Infrared Physics & Technology 37* (1996), pp. 295–306.

[36] M. R Clark, D. M McCann, and M. C Forde, "Application of Infrared Thermography to the Non-Destructive Testing of Concrete and Masonry Bridges", *in*: *NDT & E International*, Structural Faults and Repair 36.*4* (June 2003), pp. 265–275, ISSN: 0963-8695, DOI: `10.1016/S0963-8695(02)00060-9`.

[37] P. B. Coates, "Multi-Wavelength Pyrometry", *in*: *Metrologia* 17.*3* (1981), p. 103.

[38] M. F. Cohen et al., "An Efficient Radiosity Approach for Realistic Image Synthesis", *in: IEEE Computer Graphics and Applications* 6.3 (Mar. 1986), pp. 26–35, ISSN: 0272-1716, DOI: 10.1109/MCG.1986.276629.

[39] Michael F. Cohen and Donald P. Greenberg, "The Hemi-Cube: A Radiosity Solution for Complex Environments", *in: ACM SIGGRAPH Computer Graphics*, vol. 19, ACM, 1985, pp. 31–40.

[40] Michael F. Cohen and John R. Wallace, *Radiosity and Realistic Image Synthesis*, en, AP Professional, 1995, ISBN: 978-0-12-059756-7.

[41] Michael F. Cohen et al., "A Progressive Refinement Approach to Fast Radiosity Image Generation", *in: ACM SIGGRAPH computer graphics* 22.4 (1988), pp. 75–84.

[42] Greg Coombe, Mark J. Harris, and Anselmo Lastra, "Radiosity on Graphics Hardware", en, *in: ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*, Los Angeles, California: ACM Press, 2005, p. 179, DOI: 10.1145/1198555.1198782.

[43] P. I. Cooper, "The Absorption of Radiation in Solar Stills", *in: Solar Energy* 12.3 (Jan. 1969), pp. 333–346, ISSN: 0038-092X, DOI: 10.1016/0038-092X(69)90047-4.

[44] *Copernicus Data Store*, https://cds.climate.copernicus.eu/cdsapp#!/home.

[45] *Cornell Box Data*, http://www.graphics.cornell.edu/online/box/data.html.

[46] T. M. Cover and Joy A. Thomas, *Elements of Information Theory*, en, Wiley Series in Telecommunications, New York: Wiley, 1991, ISBN: 978-0-471-06259-2.

[47] Cyril Crassin et al., "Interactive Indirect Illumination Using Voxel Cone Tracing", en, *in*: (2011), p. 10.

[48] Noel A. C. Cressie and Christopher K. Wikle, *Statistics for Spatio-Temporal Data*, Wiley Series in Probability and Statistics, Hoboken, N.J: Wiley, 2011, ISBN: 978-0-471-69274-4.

[49] Antoine Crinière, "Contribution au développement d'outils d'analyse de séquences d'images infrarouges: Application au contrôle non destructif de structures de Génie Civil", fr, PhD thesis, IFSTTAR - Nantes: Centrale nantes, Oct. 2014.

[50] Antoine Crinière, Jean Dumoulin, and Laurent Mevel, "Management of Local Multi-Sensors Applied to SHM and Long-Term Infrared Monitoring: Cloud2IR Implementation", *in: Quantitative InfraRed Thermography Journal* 0.0 (Oct. 2018), pp. 1–19, ISSN: 1768-6733, DOI: 10.1080/17686733.2018.1519752.

[51] Antoine Crinière et al., "Cloud2IR: Infrared Thermography and Environmental Sensors Integrated in an Autonomoussystem for Long Term Monitoring of Structures", en, *in: European Geoscience Union EGU2016*, Apr. 2016, DOI: 10.1088/1742-2132/10/6/064003.

[52] Biswa Nath Datta, *Numerical Methods for Linear Control Systems: Design and Analysis*, en, Amsterdam ; Boston: Elsevier Academic Press, 2004, ISBN: 978-0-12-203590-6.

[53] A. de Vriendt, *La transmission de la chaleur : introduction au rayonnement thermique*, Français, 3rd ed., vol. 2, Gaëtan Morin, ISBN: 978-2-89105-461-4.

[54] Shakeb Deane et al., "Application of NDT Thermographic Imaging of Aerospace Structures", *in: Infrared Physics & Technology* 97 (2019), pp. 456–466, ISSN: 1350-4495, DOI: `https://doi.org/10.1016/j.infrared.2019.02.002`.

[55] Pierre Del Moral, *Mean Field Simulation for Monte Carlo Integration*, Monographs on Statistics & Applied Probability, Chapman&Hall, June 2013.

[56] F Derkx et al., "Concept de Plate-forme Mobile Instrumentée (PMI) pour l'inspection des ouvrages d'art", fr, *in*: (), p. 6.

[57] *Determination of Toxicity of Spoil Substrates After Brown Coal Mining Using a Laboratory Reproduction Test with Enchytraeus Crypticus (Oligochaeta) | SpringerLink*, `https://link.springer.com/article/10.1007/s11270-005-5991-y`.

[58] *Development of Statistical Temperature Prediction Models for a Test Road in Edmonton, Alberta, Canada*, en, `https://reader.elsevier.com/reader/sd/pii/S1996681416303078?token=`, DOI: `10.1016/j.ijprt.2017.05.004`.

[59] Christopher Drovandi et al., "Ensemble MCMC: Accelerating Pseudo-Marginal MCMC for State Space Models Using the Ensemble Kalman Filter", en, *in: arXiv:1906.02014 [stat]* (June 2019), arXiv: `1906.02014 [stat]`.

[60] Lorène Dubreuil, Yann Quinsat, and Claire Lartigue, *Calibration Sur Modèle CAO Pour La Mesure Par Vision In-Situ*, `https://hal.archives-ouvertes.fr/hal-01139851`, Apr. 2015.

[61] John A. Duffie and William A. Beckman, *Solar Engineering of Thermal Processes*, Anglais, 4th Edition, Hoboken: John Wiley & Sons, May 2013, ISBN: 978-0-470-87366-3.

[62] Jean Dumoulin and Vincent Boucher, "Infrared Thermography System for Transport Infrastructures Survey with Inline Local Atmospheric Parameter Measurements and Offline Model for Radiation Attenuation Evaluations", *in: Journal of Applied Remote Sensing* 8.1 (2014), pp. 084978–084978.

[63] Jean Dumoulin, Antoine Crinière, and Rodolphe Averty, "The Detection and Thermal Characterization of the Inner Structure of the 'Musmeci' Bridge Deck by Infrared Thermography Monitoring", *in: Journal of Geophysics and Engineering* 10.6 (Dec. 2013), p. 064003, ISSN: 1742-2132, 1742-2140, DOI: `10.1088/1742-2132/10/6/064003`.

[64] Philip Dutré, Kavita Bala, and Philippe Bekaert, *Advanced Global Illumination*, en, 2nd ed, OCLC: ocm68965768, Wellesley, Mass: AK Peters, 2006, ISBN: 978-1-56881-307-3.

[65] *Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps*, `https://dl.acm.org/citation.cfm?id=280925`.

[66] *Explaining the Gibbs Sampler: The American Statistician: Vol 46, No 3*, https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475878.

[67] Sandrine Fauqueux, "MATISSE-v2.0 Infrared Sea Images: Sensitivity Analysis and Experimental Validation Using MIRAMER Campaign Measurements in Solar Glint Configurations", eng, *in*: *Applied Optics* 52.*24* (Aug. 2013), pp. 6063–6073, ISSN: 1539-4522, DOI: 10.1364/AO.52.006063.

[68] Martin A. Fischler and Robert C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *in*: *Readings in Computer Vision*, San Francisco (CA): Morgan Kaufmann, 1987, pp. 726–740, ISBN: 978-0-08-051581-6, DOI: 10.1016/B978-0-08-051581-6.50070-2.

[69] Jean-Dominique Gascuel et al., "Fast Non-Linear Projections Using Graphics Hardware", en, *in*: *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games - SI3D '08*, Redwood City, California: ACM Press, 2008, p. 107, ISBN: 978-1-59593-983-8, DOI: 10.1145/1342250.1342267.

[70] Noel Gastinel, *Linear Numerical Analysis*, Anglais, Paris, New York: HERMANN, Dec. 1970, ISBN: 978-0-12-277150-7.

[71] G. Gaussorgues and S. Chomet, *Infrared Thermography*, en, Springer Science & Business Media, Dec. 1993, ISBN: 978-0-412-47900-7.

[72] W. R. Gilks and P. Wild, "Adaptive Rejection Sampling for Gibbs Sampling", *in*: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 41.*2* (1992), pp. 337–348, ISSN: 0035-9254, DOI: 10.2307/2347565.

[73] A R Gillespie et al., *Temperature/Emissivity Separation Algorithm Theoretical Basis Document, Version 2.4*, en, 1999.

[74] Alan R. Gillespie et al., "Residual Errors in ASTER Temperature and Emissivity Standard Products AST08 and AST05", *in*: *Remote Sensing of Environment* 115.*12* (Dec. 2011), pp. 3681–3694, ISSN: 0034-4257, DOI: 10.1016/j.rse.2011.09.007.

[75] Cindy M. Goral et al., "Modeling the Interaction of Light Between Diffuse Surfaces", *in*: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, New York, NY, USA: ACM, 1984, pp. 213–222, ISBN: 978-0-89791-138-2, DOI: 10.1145/800031.808601.

[76] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", *in*: *IEE Proceedings F - Radar and Signal Processing* 140.*2* (Apr. 1993), pp. 107–113, ISSN: 0956-375X, DOI: 10.1049/ip-f-2.1993.0015.

[77] S. Gortler, M.F. Cohen, and P. Slusallek, "Radiosity and Relaxation Methods", en, *in*: *IEEE Computer Graphics and Applications* 14.*6* (Nov. 1994), pp. 48–58, ISSN: 0272-1716, DOI: 10.1109/38.329094.

[78] *GPU CUDA and MEX Programming - MATLAB & Simulink - MathWorks France*, https://fr.mathworks.com/help/parallel-computing/gpu-cuda-and-mex-programming.html?s_tid=CRUX_lftnav.

[79] Greg Coombe and Mark Harris, "Chapter 39. Global Illumination Using Progressive Refinement Radiosity", *in*: *GPU Gems: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, vol. 2, ISBN: 0-321-33559-7.

[80] E. Grinzato, V. Vavilov, and T. Kauppinen, "Quantitative Infrared Thermography in Buildings", *in*: *Energy and Buildings* 29.*1* (Dec. 1998), pp. 1–9, ISSN: 0378-7788, DOI: 10.1016/S0378-7788(97)00039-X.

[81] Dominique Groleau, Fernand Fragnaud, and Jean-Michel Rosant, "Simulation of the Radiative Behavior of an Urban Quarter of Marseille with the SOLENE Model", en, *in*: *Proceedings ICUC-5*, University of Lodz, Poland, Sept. 2003, p. 4.

[82] Christian Gueymard, "Simple Model for the Atmospheric Radiative Transfer of Sunshine (SMARTS2) Algorithms and Performance Assessment", en, *in*: (1995).

[83] D. C. Hamilton and W. R. Morgan, "Radiant-Interchange Configuration Factors", en, *in*: (Dec. 1952).

[84] Nikolaus Hansen, *The CMA Evolution Strategy: A Comparing Review*, 2006.

[85] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second, Cambridge University Press, ISBN: 0521540518, 2004.

[86] Paul S. Heckbert, "Adaptive Radiosity Textures for Bidirectional Ray Tracing", *in*: *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, New York, NY, USA: ACM, 1990, pp. 145–154, ISBN: 978-0-89791-344-7, DOI: 10.1145/97879.97895.

[87] H. V. Henderson and S. R. Searle, "On Deriving the Inverse of a Sum of Matrices", en, *in*: *SIAM Review* 23.*1* (Jan. 1981), pp. 53–60, ISSN: 0036-1445, 1095-7200, DOI: 10.1137/1023004.

[88] Aurélien Hénon et al., "High Resolution Thermo-Radiative Modeling of an Urban Fragment in Marseilles City Center during the UBL-ESCOMPTE Campaign", en, *in*: *Building and Environment* 46.*9* (Sept. 2011), pp. 1747–1764, ISSN: 03601323, DOI: 10.1016/j.buildenv.2011.02.001.

[89] *Hierarchical Radiosity On Curved Surfaces | SpringerLink*, https://link.springer.com/chapter/10.1007%2F978-3-7091-6858-5_17.

[90] Simon J Hook et al., "A Comparison of Techniques for Extracting Emissivity Information from Thermal Infrared Data for Geologic Studies", *in*: *Remote Sensing of Environment* 42.*2* (Nov. 1992), pp. 123–135, ISSN: 0034-4257, DOI: 10.1016/0034-4257(92)90096-3.

[91] H.C. Hottel, "Radiant Heat Transmission between Surfaces Separated by Non-Absorbing Media", *in*: *Journal of Heat Transfer* (1931), pp. 265–273.

[92]   R. J. Howarth, "(A. G.) Journel and (Ch. J.) Huijbregts. Mining Geostatistics.", en, *in*: *Mineralogical Magazine* 43.*328* (Dec. 1979), pp. 563–564, ISSN: 0026-461X, 1471-8022, DOI: 10.1180/minmag.1979.043.328.34.

[93]   J.R. Howell, R. Siegel, and M.P. Pinar, *Thermal Radiation Heat Transfer.* 5th, CRC Press, 2010.

[94]   *IASLonline NetArt: History of Computer Art IV.2 Computer Animation*, http://iasl.uni-muenchen.de/links/GCA-IV.2e.html#Sechziger.

[95]   *Integrated System of Structural Health Monitoring and Intelligent Management for a Cable-Stayed Bridge*, https://www.hindawi.com/journals/tswj/2014/689471/.

[96]   Hamlyn Jones and Xavier Sirault, "Scaling of Thermal Images at Different Spatial Resolution: The Mixed Pixel Problem", en, *in*: *Agronomy* 4.*3* (July 2014), pp. 380–396, ISSN: 2073-4395, DOI: 10.3390/agronomy4030380.

[97]   Nikolas Kantas et al., "On Particle Methods for Parameter Estimation in State-Space Models", en, *in*: *Statistical Science* 30.*3* (Aug. 2015), pp. 328–351, ISSN: 0883-4237, DOI: 10.1214/14-STS511.

[98]   Shigeto Kawashima et al., "Relations between Surface Temperature and Air Temperature on a Local Scale during Winter Nights", en, *in*: *Journal of Applied Meteorology* 39.*9* (Sept. 2000), pp. 1570–1579, ISSN: 0894-8763, 1520-0450, DOI: 10.1175/1520-0450(2000)039<1570:RBSTAA>2.0.CO;2.

[99]   Alexander Keller, "Instant Radiosity", en, *in*: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '97*, Not Known: ACM Press, 1997, pp. 49–56, ISBN: 978-0-89791-896-1, DOI: 10.1145/258734.258769.

[100]   Mansoor A. Khan, Charly Allemand, and Thomas W. Eagar, "Noncontact Temperature Measurement. I. Interpolation Based Techniques", *in*: *Review of Scientific Instruments* 62.*2* (Feb. 1991), pp. 392–402, ISSN: 0034-6748, DOI: 10.1063/1.1142133.

[101]   Mikhail Konnik and James Welsh, "High-Level Numerical Simulations of Noise in CCD and CMOS Photosensors: Review and Tutorial", *in*: *arXiv:1412.4031 [astro-ph]* (Dec. 2014), arXiv: 1412.4031 [astro-ph].

[102]   Jean-Claude Krapez, "Radiative Measurements of Temperature", *in*: *Thermal Measurements and Inverse Techniques*, Heat Transfer, OCLC: ocn587104377, Boca Raton, FL: CRC Press, 2011, ISBN: 978-1-4398-4555-4.

[103]   Nicolas Lalanne et al., "Development and Validation of a Numerical Tool for Simulating the Surface Temperature Field and the Infrared Radiance Rendering in an Urban Scene", en, *in*: *Quantitative InfraRed Thermography Journal* 12.*2* (July 2015), pp. 196–218, ISSN: 1768-6733, 2116-7176, DOI: 10.1080/17686733.2015.1066134.

[104] B Lane et al., "Uncertainty of Temperature Measurements by Infrared Thermography for Metal Cutting Applications", *in*: *Metrologia* 50.*6* (Dec. 2013), pp. 637–653, ISSN: 0026-1394, 1681-7575, DOI: 10.1088/0026-1394/50/6/637.

[105] Bertrand Lascoup et al., "On the Feasibility of Defect Detection in Composite Material Based on Thermal Periodic Excitation", *in*: *Composites Part B: Engineering* 45 (Feb. 2013), pp. 1023–1030, DOI: 10.1016/j.compositesb.2012.07.020.

[106] Gilles Laurent et al., "Forward Light Cuts: A Scalable Approach to Real-Time Global Illumination", en, *in*: *Computer Graphics Forum* 35.*4* (July 2016), pp. 79–88, ISSN: 01677055, DOI: 10.1111/cgf.12951.

[107] Mickaël Le Bohec, "Contribution Du Rayonnement Au Confort Thermique et Aux Économies d'énergie Dans l'habitat", Thesis, Chasseneuil-du-Poitou, Ecole nationale supérieure de mécanique et d'aérotechnique, Dec. 2017.

[108] Vincent Le Cam et al., "Applications Des Réseaux de Capteurs Intelligents et de La Communication sans Fil à l'instrumentation Des Structures de Génie Civil", *in*: *Bulletin des Laboratoires des Ponts et Chaussées 273* (Jan. 2008), pp. 9–37.

[109] Nicolas Le Touz, Thibaud Toullier, and Jean Dumoulin, "Infrared Thermography Applied to the Study of Heated and Solar Pavement: From Numerical Modeling to Small Scale Laboratory Experiments", *in*: *Thermosense: Thermal Infrared Applications XXXIX*, vol. 10214, International Society for Optics and Photonics, 2017, p. 1021413.

[110] Sylvain Lefebvre, Samuel Hornus, and Fabrice Neyret, "Chapter 37. Octree Textures on the GPU", *in*: *GPU Gems: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, vol. 2, ISBN: 0-321-33559-7.

[111] Jaakko Lehtinen et al., "A Meshless Hierarchical Representation for Light Transport", en, *in*: *ACM Transactions on Graphics* 27.*3* (Aug. 2008), p. 1, ISSN: 07300301, DOI: 10.1145/1360612.1360636.

[112] Jaakko Lehtinen et al., *Meshless Finite Elements for Hierarchical Global Illumination*, en, Technical Report Technical Report TML-B7, Publications in Telecommunications Software and Multimedia, Helsinki University of Technology, p. 12.

[113] *L'entretien du réseau routier en France*, fr, http://www.ecologique-solidaire.gouv.fr/lentretien-du-reseau-routier-en-france.

[114] Zhao-Liang Li and François Becker, "Feasibility of Land Surface Temperature and Emissivity Determination from AVHRR Data", en, *in*: *Remote Sensing of Environment* 43.*1* (Jan. 1993), pp. 67–85, ISSN: 00344257, DOI: 10.1016/0034-4257(93)90065-6.

[115] Zhaoliang Li, F. Petitcolin, and Renhua Zhang, "A Physically Based Algorithm for Land Surface Emissivity Retrieval from Combined Mid-Infrared and Thermal Infrared Data", en, *in*: *Science in China Series E: Technological Sciences* 43.*1* (Dec. 2000), pp. 23–33, ISSN: 1862-281X, DOI: 10.1007/BF02916575.

[116] Zhao-Liang Li et al., "Evaluation of Six Methods for Extracting Relative Emissivity Spectra from Thermal Infrared Images", *in*: *Remote sensing of Environment* 69.*3* (1999), pp. 197–214.

[117] Zhao-Liang Li et al., "Satellite-Derived Land Surface Temperature: Current Status and Perspectives", *in*: *Remote Sensing of Environment* 131 (Apr. 2013), pp. 14–37, ISSN: 0034-4257, DOI: 10.1016/j.rse.2012.12.008.

[118] Zhen Li, Lifang Zhao, and Zhuo Fu, "Estimating Net Radiation Flux in the Tibetan Plateau by Assimilating MODIS LST Products with an Ensemble Kalman Filter and Particle Filter", *in*: *International Journal of Applied Earth Observation and Geoinformation* 19 (Oct. 2012), pp. 1–11, ISSN: 0303-2434, DOI: 10.1016/j.jag.2012.04.003.

[119] Jun S. Liu and Rong Chen, "Sequential Monte Carlo Methods for Dynamic Systems", *in*: *Journal of the American Statistical Association* 93.*443* (1998), pp. 1032–1044, ISSN: 0162-1459, DOI: 10.2307/2669847.

[120] G Machin and B Chu, "High-Quality Blackbody Sources for Infrared Thermometry and Thermography between -40 and 1000řC", en, *in*: *The Imaging Science Journal* 48.*1* (Jan. 2000), pp. 15–22, ISSN: 1368-2199, 1743-131X, DOI: 10.1080/13682199.2000.11784341.

[121] G. Masiello et al., "Kalman Filter Physical Retrieval of Surface Emissivity and Temperature from Geostationary Infrared Radiances", en, *in*: *Atmospheric Measurement Techniques* 6.*12* (Dec. 2013), pp. 3613–3634, ISSN: 1867-8548, DOI: 10.5194/amt-6-3613-2013.

[122] Aristide Mbiock and Roman Weber, *Radiation in Enclosures: Elliptic Boundary Value Problem*, en, Scientific Computation, Berlin ; New York: Springer, 2000, ISBN: 978-3-540-66095-8.

[123] Larry M. McMillin, "Estimation of Sea Surface Temperatures from Two Infrared Window Measurements with Different Absorption", en, *in*: *Journal of Geophysical Research (1896-1977)* 80.*36* (1975), pp. 5113–5117, ISSN: 2156-2202, DOI: 10.1029/JC080i036p05113.

[124] Rihab Mechri et al., "Genetic Particle Filter Application to Land Surface Temperature Downscaling", en, *in*: *Journal of Geophysical Research: Atmospheres* 119.*5* (Mar. 2014), pp. 2131–2146, ISSN: 2169897X, DOI: 10.1002/2013JD020354.

[125] Susan K. Meerdink et al., "The ECOSTRESS Spectral Library Version 1.0", *in*: *Remote Sensing of Environment* 230 (Sept. 2019), p. 111196, ISSN: 0034-4257, DOI: 10.1016/j.rse.2019.05.015.

[126] Micha Mettke, *Nuklear: A Single-Header ANSI C Gui Library*, Aug. 2018.

[127] W. Minkina and S. Dudzik, "Simulation Analysis of Uncertainty of Infrared Camera Measurement and Processing Path", *in*: *Measurement* 39.*8* (Oct. 2006), pp. 758–763, ISSN: 0263-2241, DOI: 10 . 1016/j.measurement.2006.03.001.

[128] Waldemar Minkina and Sebastian Dudzik, *Infrared Thermography: Errors and Uncertainties*, en, John Wiley & Sons, Sept. 2009, ISBN: 978-0-470-68224-1.

[129] M. F. Modest, *Radiative Heat Transfer*, en, Third Edition, OCLC: ocn813855549, New York: Academic Press, 2013, ISBN: 978-0-12-386944-9.

[130] Jean-Pierre Monchau, "Mesure d'émissivité pour la thermographie infrarouge appliquée au diagnostic quantitatif des structures", fr, PhD thesis, Université Paris-Est, Nov. 2013.

[131] Parry Hiram Moon, *The Scientific Basis of Illuminating Engineering*, en, McGraw-Hill book Company, Incorporated, 1936.

[132] Wilhelm Nusselt, "Graphische bestimmung des winkelverhaltnisses bei der wärmestrahlung", de, *in*: *Zeitschrift des Vereines Deutscher Ingenieure 20* (1928), p. 673.

[133] Tarek Omar and Moncef L. Nehdi, "Remote Sensing of Concrete Bridge Decks Using Unmanned Aerial Vehicle Infrared Thermography", *in*: *Automation in Construction* 83 (2017), pp. 360–371, ISSN: 0926-5805, DOI: https://doi.org/10.1016/j.autcon.2017.06.024.

[134] *OpenGL - The Industry Standard for High Performance Graphics*, https://www.opengl.org/.

[135] Helcio R. B. Orlande, ed., *Thermal Measurements and Inverse Techniques*, Heat Transfer, OCLC: ocn587104377, Boca Raton, FL: CRC Press, 2011, ISBN: 978-1-4398-4555-4.

[136] Mathias Paulin and Jean-Pierre Jessel, "Adaptive Mesh Generation for Progressive Radiosity: A Ray-Tracing Based Algorithm.", en, *in*: *Computer Graphics Forum* 13.*3* (1994), pp. 421–432, ISSN: 1467-8659, DOI: 10.1111/1467-8659.1330421.

[137] *(PDF) Land Surface Temperatures Derived from the Advanced Very High Resolution Radiometer and the Along-Track Scanning Radiometer 1. Theory*, https://www.researchgate.net/publication/ 238027345_Land_surface_temperatures_derived_from_the_Advanced_Very_High_ Resolution_Radiometer_and_the_Along-Track_Scanning_Radiometer_1_Theory.

[138] *(PDF) The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium*, https: //www.researchgate.net/publication/220720239_The_zonal_method_for_ calculating_light_intensities_in_the_presence_of_a_participating_medium.

[139] Edzer Pebesma, *The Meuse Data Set: A Brief Tutorial for the Gstat R Package*, en, May 2019.

[140] M. Pellegrini, "Monte Carlo Approximation of Form Factors with Error Bounded a Priori", en, *in*: *Discrete & Computational Geometry* 17.*3* (Apr. 1997), pp. 319–337, ISSN: 1432-0444, DOI: 10.1007/ PL00009297.

[141] Markus Pfänder, Eckhard Lüpfert, and Peter Heller, "Pyrometric Temperature Measurements on Solar Thermal High Temperature Receivers", en, *in: Journal of Solar Energy Engineering* 128.*3* (2006), p. 285, ISSN: 01996231, DOI: 10.1115/1.2210499.

[142] *Physically Based Rendering: From Theory to Implementation*, http://www.pbr-book.org/3ed-2018/contents.html.

[143] Joseph H. Pierluissi, Christos E. Maragoudakis, and Rouzbeh Tehrani-Movahed, "New Lowtran Band Model for Water Vapor", en, *in: Applied Optics* 28.*18* (Sept. 1989), p. 3792, ISSN: 0003-6935, 1539-4522, DOI: 10.1364/AO.28.003792.

[144] Max Planck, "The Theory of Heat Radiation", *in: Entropie* 144.*190* (1900), p. 164.

[145] Wang Qi, Fu Li, and Liu Zhenzhong, "Review on Camera Calibration", *in: 2010 Chinese Control and Decision Conference*, May 2010, pp. 3354–3358, DOI: 10.1109/CCDC.2010.5498574.

[146] *Retrieval of Atmospheric and Land Surface Parameters from Satellite-Based Thermal Infrared Hyperspectral Data Using a Neural Network Technique: International Journal of Remote Sensing: Vol 34, No 9-10*, https://www.tandfonline.com/doi/abs/10.1080/01431161.2012.716536.

[147] A. Rogalski, "History of Infrared Detectors", en, *in: Opto-Electronics Review* 20.*3* (Jan. 2012), ISSN: 1896-3757, DOI: 10.2478/s11772-012-0037-7.

[148] Olivier Roustant and Yves Deville, "(Re)Dice Packages for Computer Experiments", fr, *in: Troisièmes rencontres R*, Montpellier, France, June 2014.

[149] Olivier Roustant, David Ginsbourger, and Yves Deville, ", : Two Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization", en, *in: Journal of Statistical Software* 51 (2012), DOI: 10.18637/jss.v051.i01.

[150] Donald E. Sabol et al., "Field Validation of the ASTER TemperatureEmissivity Separation Algorithm", *in: Remote Sensing of Environment* 113.*11* (Nov. 2009), pp. 2328–2344, ISSN: 0034-4257, DOI: 10.1016/j.rse.2009.06.008.

[151] Fumihiro Sakuma and Susumu Hattori, "Establishing a practical temperature standard by using a narrow-band radiation thermometer with a silicon detector", jpn, *in: Keiry kenkyjo hkoku* 2 (Apr. 1983), p91–97.

[152] Paul D. Sampson, "Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm", *in: Computer graphics and image processing* 18.*1* (1982), pp. 97–108.

[153] P. Saunders, "General Interpolation Equations for the Calibration of Radiation Thermometers", en, *in: Metrologia* 34.*3* (1997), p. 201, ISSN: 0026-1394, DOI: 10.1088/0026-1394/34/3/1.

[154] B. Segovia, J.C. Iehl, and B. Péroche, "Metropolis Instant Radiosity", en, *in: Computer Graphics Forum* 26.*3* (Sept. 2007), pp. 425–434, ISSN: 0167-7055, 1467-8659, DOI: 10.1111/j.1467-8659.2007.01065.x.

[155] A. B. Shapiro, "Computer Implementation, Accuracy, and Timing of Radiation View Factor Algorithms", *in*: *Journal of Heat Transfer* 107.*3* (Aug. 1985), pp. 730–732, ISSN: 0022-1481, DOI: 10. 1115/1.3247490.

[156] Mark S. Shephard, "Approaches to the Automatic Generation and Control of Finite Element Meshes", *in*: *Applied Mechanics Reviews* 41.*4* (Apr. 1988), pp. 169–185, ISSN: 0003-6900, DOI: 10. 1115/1.3151889.

[157] Peter Shirley, "Hybrid Radiosity/Monte Carlo Methods", en, *in*: *Monte Carlo Methods* (), p. 24.

[158] Peter Shirley, "Time Complexity of Monte Carlo Radiosity", *in*: *Computers & Graphics* 16.*1* (Jan. 1992), pp. 117–120, ISSN: 0097-8493, DOI: 10.1016/0097-8493(92)90078-A.

[159] Dave Shreiner et al., *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*, Anglais, 8th ed., Addison-Wesley Professional, Mar. 2013.

[160] Robert Shumway and David Stoffer, *Time Series: A Data Analysis Approach Using R*, English, 1 edition, Boca Raton: Chapman and Hall/CRC, May 2019, ISBN: 978-0-367-22109-6.

[161] François X. Sillion, "A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters", en, *in*: *IEEE Transactions on Visualization and Computer Graphics* 1.*3* (1995).

[162] François X. Sillion and Claude Puech, "A General Two-Pass Method Integrating Specular and Diffuse Reflection", en, *in*: *Proceedings of SIGGRAPH '89*, ACM Press, 1989, pp. 335–344.

[163] François X. Sillion and Claude Puech, *Radiosity and Global Illumination*, vol. 11, Morgan Kaufmann Publishers, 1994, DOI: 10.1007/BF01909880.

[164] François X. Sillion et al., "A Global Illumination Solution for General Reflectance Distributions", *in*: *SIGGRAPH'91 Conference Proceedings*, Las Vegas, United States: ACM Press, 1991, pp. 187–196.

[165] *SMARTS: Simple Model of the Atmospheric Radiative Transfer of Sunshine | Grid Modernization | NREL*, https://www.nrel.gov/grid/solar-resource/smarts.html.

[166] W. C. Snyder et al., "Classification-Based Emissivity for Land Surface Temperature Measurement from Space", en, *in*: *International Journal of Remote Sensing* 19.*14* (Jan. 1998), pp. 2753–2774, ISSN: 0143-1161, 1366-5901, DOI: 10.1080/014311698214497.

[167] J.A. Sobrino et al., "Improvements in the Split-Window Technique for Land Surface Temperature Determination", *in*: *IEEE Transactions on Geoscience and Remote Sensing* 32.*2* (Mar. 1994), pp. 243–253, ISSN: 01962892, DOI: 10.1109/36.295038.

[168] Jose Sobrino et al., "Multi-Channel and Multi-Angle Algorithms for Estimating Sea and Land Surface Temperature with ATSR Data", *in*: *International Journal of Remote Sensing* 17 (July 1996), pp. 2089–2114, DOI: 10.1080/01431169608948760.

[169] Guillem Sòria and José A. Sobrino, "ENVISAT/AATSR Derived Land Surface Temperature over a Heterogeneous Region", *in*: *Remote Sensing of Environment* 111.4 (Dec. 2007), pp. 409–422, ISSN: 0034-4257, DOI: 10.1016/j.rse.2007.03.017.

[170] Ephraim M. Sparrow and R. D. Cess, *Radiation Heat Transfer*, en, Hemisphere Pub. Corp., 1978, ISBN: 978-0-07-059910-9.

[171] *Spatio-Temporal Field Estimation Using Kriged Kalman Filter (KKF) with Sparsity-Enforcing Sensor Placement*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6021959/.

[172] J. W. Spencer, "Fourier Series Representation of the Position of the Sun", *in*: *Search* 2.5 (1971), p. 172.

[173] Michael L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, en, Springer Series in Statistics, New York: Springer-Verlag, 1999, ISBN: 978-0-387-98629-6.

[174] I. E. Sutherland, "Three-Dimensional Data Input by Tablet", *in*: *Proceedings of the IEEE* 62.4 (Apr. 1974), pp. 453–461, ISSN: 0018-9219, DOI: 10.1109/PROC.1974.9449.

[175] Flir Systems, *The Ultimate Infrared Handbook for R&D Professionals*.

[176] Rod Tennyson et al., "Structural Health Monitoring of Innovative Bridges in Canada with Fiber Optic Sensors", *in*: *Smart Materials and Structures* 10 (June 2001), p. 560, DOI: 10.1088/0964-1726/10/3/320.

[177] *The Application of Artificial Neural Networks to the Analysis of Remotely Sensed Data: International Journal of Remote Sensing: Vol 29, No 3*, https://www.tandfonline.com/doi/full/10.1080/01431160701352154.

[178] *The Official YAML Web Site*, https://yaml.org/.

[179] Thibaud Toullier, J. Dumoulin, and V Bourgeois, "Comparative Study of Moving Train Hot Boxes Pre-Detection and Axles Counting by in-Situ Implementation of Two Infrared Cameras", *in*: *QIRT Asia 2019 - Quantitative InfraRed Thermography Conference*, Tokyo, Japan, July 2019.

[180] Nicolas Le Touz, "Conception et étude d'infrastructures de transports à énergie positive: de la modélisation thermomécanique à l'optimisation de tels systèmes énergétiques", fr, PhD thesis, IFSTTAR - Nantes: Centrale Nantes, 2018.

[181] M. P. Utrillas et al., "A Comparative Study of SPCTRAL2 and SMARTS2 Parameterised Models Based on Spectral Irradiance Measurements at Valencia, Spain", *in*: *Solar Energy* 63.3 (Sept. 1998), pp. 161–171, ISSN: 0038-092X, DOI: 10.1016/S0038-092X(98)00058-9.

[182] H.C. Van de Hulst, *Light Scattering by Small Particles*, John Wiley & Sons, 1957.

[183] Sven Van De Vijver et al., "The Influence of Environmental Parameters on the Thermographic Analysis of the Building Envelope", *in*: *12th International Conference on Quantitative InfraRed Thermography (QIRT 2014)*, 2014.

[184] Christophe Vedel and Claude Puech, "A Testbed for Adaptive Subdivision in Progressive Radiosity", *in*: *Second Eurographics Workshop on Rendering*, Barcelona, Spain, 1991.

[185] Sara Venafra et al., "RETRIEVAL OF SURFACE TEMPERATURE AND EMISSIVITY FROM SEVIRI DATA: IMPLEMENTATION OF KALMAN FILTER METHODOLOGY TO FULL DISK AND VALIDATION WITH IN-SITU OBSERVATIONS", en, *in*: (2015), DOI: 10.13140/rg.2.1.2339.5924.

[186] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg, "A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods", *in*: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, New York, NY, USA: ACM, 1987, pp. 311–320, ISBN: 978-0-89791-227-3, DOI: 10.1145/37401.37438.

[187] Günter Wallner, "An Extended GPU Radiosity Solver: Including Diffuse and Specular Reflectance and Transmission", en, *in*: *The Visual Computer* 25.*5-7* (May 2009), pp. 529–537, ISSN: 0178-2789, 1432-2315, DOI: 10.1007/s00371-009-0347-z.

[188] Bruce Walter et al., "Lightcuts: A Scalable Approach to Illumination", en, *in*: *ACM SIGGRAPH 2005*, Los Angeles, California, July 2005, p. 10.

[189] Hongchen Wang et al., "Fabricating Microbolometer Array on Unplanar Readout Integrated Circuit", en, *in*: *International Journal of Infrared and Millimeter Waves* 26.5 (May 2005), pp. 751–762, ISSN: 0195-9271, 1572-9559, DOI: 10.1007/s10762-005-4983-8.

[190] Turner Whitted, "An Improved Illumination Model for Shaded Display", *in*: *Commun. ACM* 23.6 (June 1980), pp. 343–349, ISSN: 0001-0782, DOI: 10.1145/358876.358882.

[191] OpenGL Wiki, *Rendering Pipeline Overview — OpenGL Wiki,* [Online; last accessed June-2019], 2019, URL: %5Curl%7Bhttp://www.khronos.org/opengl/wiki_opengl/index.php?title=Rendering_Pipeline_Overview&oldid=14511%7D.

[192] Christopher K. Wikle and Noel Cressie, *A Dimension-Reduction Approach to Space-Time Kalman Filtering*, 1999.

[193] J.-J. Yon et al., "320 x 240 Uncooled IRFPA with Pixel Wise Thin Film Vacuum Packaging", *in*: *Electro-Optical and Infrared Systems: Technology and Applications IX*, vol. 8541, International Society for Optics and Photonics, Oct. 2012, p. 85410D, DOI: 10.1117/12.974846.

[194] Meriem Zghal, Laurent Mevel, and Pierre Del Moral, "Modal parameter estimation using interacting Kalman filter", *in*: *Mechanical Systems and Signal Processing*, Special Issue on identification of time varying systems 47.*1-2* (Feb. 2014), pp. 139–150, DOI: 10.1016/j.ymssp.2012.11.005, URL: https://hal.inria.fr/hal-00824108.

Appendices

YAML batch processing

This appendix shows an example of a YAML batch processing file used in the multi-sensors data visualization and processing tool.

```
1   #================================================================
2   # This is a sample YAML file for running batch processes
3   # [o] stands for optionnal
4   properties:          # [o] -> properties of the simulation
5     recursive:         #      -> Either to look onto subfolder or not.
6     showprogress:      #      -> show progress of the process
7     logfile:           # [o] -> save the output to a log file
8
9   input:               #      -> tag for defining the IR data
10    folders:           #      -> can be a path to a file or a folder
11    date:              # [o] -> defines the range of time to consider
12      beginDate:       #      ->
13      endDate:         #      ->
14      fileFormat:      #      -> the format of date in the filename
15      fileDateExtract: #      -> the way to extract the date in the filename (regexp)
16    output:            #      -> Initialize the fields of outputData
17
18  actionList:          #      -> which actions should be executed in which order
19    myaction:          #      -> the action name
20    ...
21
22  myaction:            #      -> The actual definition of the action "myaction"
23    func:              #      -> The function that will be executed
24    type:              #      -> batch: will be executed for every single file
25                       #      -> before/after: will be executed in the main program
26    frameRate:         # [o] -> FrameRate if batch
27    beginDate:         # [o] -> Begin date if batch
28    endDate:           #      -> End date if batch
29    args:              # [o] -> arguments that will be given to the function
30    extra:             # [o] -> extra arguments. For clarity, you can pass
31      name1: value1    #         arguments with the tag 'extra'. Its child tags
32      name2: value2    #         will be parsed as 'name', 'value' arguments.
33  #================================================================
```

YAML example configuration file

An example of a YAML configuration file used in the infrared multi-spectral images simulator is given.

```yaml
1  #================================================================
2  # This is a sample YAML file for creating a simulation.
3  # The configuration can be defined using the following tags
4  # [o] stands for optionnal
5  parameters:             # [o] -> parameters used in the following, always FIRST
6    width  : &width 800   #          You can also create anchors with the '&' operator
7    height : &height 600  #
8    wheelSize : &wheelSize 5
9
10 scene:                                         #     -> Scene properties
11   file              : "./defaults/cornellBox.obj" #    -> Path to the 3D model
12   working-directory : "/tmp"                   # [o] -> Where to save variables
13   exr-data:                                    # [o] -> to save / import data
14     type : "export"                            #     -> export / import mode
15     file : "/tmp/cornellBox.exr"               #     -> which file
16
17 cameras:                       # -> Cameras defined here, at least one
18   - name        : "Camera 1"   # -> name of the camera
19     width       : *width       # -> width (resolution in pixels)
20     height      : *height      # -> height (resolution in pixels)
21     position    :              # -> position in the scene
22       - !!formula "floor(-6.5 + 5 * 2)" #ã-> use of formulae
23       - !!formula "floor(-6.5 + 5 * 2)"
24       - !!formula "1.0 + wheelSize"
25   ...
26 objects:              # [o] -> Scene objects list properties
27   - name       : "left"  # -> name of the object
28     material   : "brick" # -> which material to use (see below)
29     temperature : 293.15  # -> Object's temperature
30 materials:                # [o] -> Scene materials
31   - name       : "brick"    # -> name of the material, for reference in the objects list
32     emissivity : [0.9, 0.6] # -> emissivity (put brackets even with one band: [0.9])
```

Simulator Python script example

This appendix shows an example of a Python script, used to simulate a 3D scene over 3 days.

```python
from matplotlib.image import imsave
from matplotlib import pyplot as plt

print("Simulation over 3 days from 19th march")
days = range(79, 82, 1)
hours = range(0, 24, 1)                                   # 24H
minutes = range(0, 59, 60)                                # 24H
radValues = []                                            # For saving data
for idxD, day in enumerate(days):
    infrashell.set_day(day)                               # Set the current day
    for idxH, hour in enumerate(hours):                   # For every hour
        infrashell.log(str(idxH+1) + '/' + str(len(hours))) # Print progress
        infrashell.set_hour(hour)                         # Set the current hour
        for idxMn, mn in enumerate(minutes):              # For every minute
            infrashell.set_minutes(mn)                    # Set the current minute
            infrashell.reset_radiosity()                  # Reset the computation
            infrashell.render()                           # Render the radiosity
            img = infrashell.get_image()                  # Get the result
            fname = 'day_' + str(day) + \
                    '_hour_' + str(hour) + \
                    '_mn_' + str(mn) + '.png';            # Save it
            imsave('./'+fname, img, origin='lower')
            radValues.append(infrashell.get_value("roof", 91)) # Value from 91st face

plt.plot(radValues)                                       # The values can be plotted
```

Temperature auto-regressive model

In this appendix, the dynamic model that will describe the temperature's evolution through time is discussed. Since the temperature evolution through time has to be predicted in the Kalman filter, a model for this prediction is needed. The evolution of daily temperature follows some patterns that can be exploited.

D.1   Seasonal ARIMA model

In order to get a dynamic model from the data, an autoregressive integrated moving average (ARIMA) model with seasoning can be used. Let $\{X_t, t \in \mathbb{Z}\}$ be a time serie process. An autocorrelation function (ACF) gives correlations between the different measurements $X_{t-1}, X_{t-2}, X_{t-3},...$ respectively called lags $1, 2, 3,...$ (time span between observations).

Definition D.1.1 (Weakly stationary)**.** The process $\{X_t, t \in \mathbb{Z}\}$ is weakly stationary if
- The mean value μ is the same at all time points
- The variance γ is the same at all time points
- The covariance between any two time points $t, t - k$ depends only on k and not on the value of t

The definition of a weakly stationary process is important for the ACF to make sense and therefore for an ARIMA model. An ARIMA model is defined by the combination of an autoregressive model (AR), a moving average one (MA) and a differencing part (I).

An autoregressive model of order p, denoted $AR(p)$ is given by:

Definition D.1.2 (AR(p))**.**

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + w_t \tag{D.1}$$

with $(\Phi_i)_{1 \leq i \leq p}$ the model parameters, c a constant and $w_t \sim \mathcal{N}(0, \sigma_w^2)$

A moving average model of order q, denoted $MA(q)$ is a weighted past error, given by:

Definition D.1.3 (MA(q))**.**

$$X_t = \mu + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + + \theta_q w_{t-q} \tag{D.2}$$

Let defines the lag operator, L which will be used in the ARIMA model definition.

Definition D.1.4 (Lag operator). Given the time serie $X_t, t \in \mathbb{Z}$, the lag operator L produces the previous element of a time serie:

$$LX_t = X_{t-1}, \forall t \geq 1 \tag{D.3}$$

An $ARMA(p, q)$ model can be defined as:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + ... + +\theta_q w_{t-q} \tag{D.4}$$

By using the lag operator L:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) X_t = c + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) w_t \tag{D.5}$$

In the case that the polynomial $\left(1 - \sum_{i=1}^{p} \phi_i L^i\right)$ has a unit root of multiplicity d, this defines an ARIMA(p', d, q) model:

$$\left(1 - \sum_{i=1}^{p'} \phi_i L^i\right)(1 - L)^d X_t = c + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) w_t \tag{D.6}$$

Please note that an ARIMA(0,1,0) leads to a random walk. Moreover, seasonality can be added to the ARIMA model when high values or low values tend always occur in particular pattern. In that case, a seasonal ARIMA model (SARIMA) can be added for which an ARIMA model is multiplied by the seasonality lag:

Definition D.1.5 (Seasonal ARIMA model). In a SARIMA model, seasonal and non-seasonal components are mixed up. Let S be the seasonality ($S = 12$ for monthly seasonality for example). We define the SARIMA model such as: $(p, d, q) \times (P, D, Q)_S$ where (p, d, q) and (P, D, Q) are the non-seasonal and seasonal components respectively.

$$\begin{cases} \left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1 - L)^d X_t & = c + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) w_t \\ \left(1 - \sum_{i=1}^{P} \phi_i L^{Si}\right)(1 - L)^D X_t & = c + \left(1 + \sum_{i=1}^{Q} \theta_i L^{Si}\right) w_t \end{cases} \tag{D.7}$$

By removing the seasonal component of the signal, the SARIMA model enables the use of lower polynomial orders than an ARIMA equivalent would have used in order to improve the computation performances.

D.2 SARIMA model for daily surface temperature

The SARIMA model have been fitted to actual measurements by using the `astsa` (Applied Statistical Time Series Analysis) R package [160]. A daily seasonal model has been chosen to take into account the

day / night variation of the temperature.

The figure Fig. D.1 shows the result of such model applied to one pixel's measurement evolution through time. The SARIMA model $((2, 0, 0) \times (0, 1, 1)_1)$ is trained on the last 3 days and 1 day forecast is performed and compared to the next day measurement.
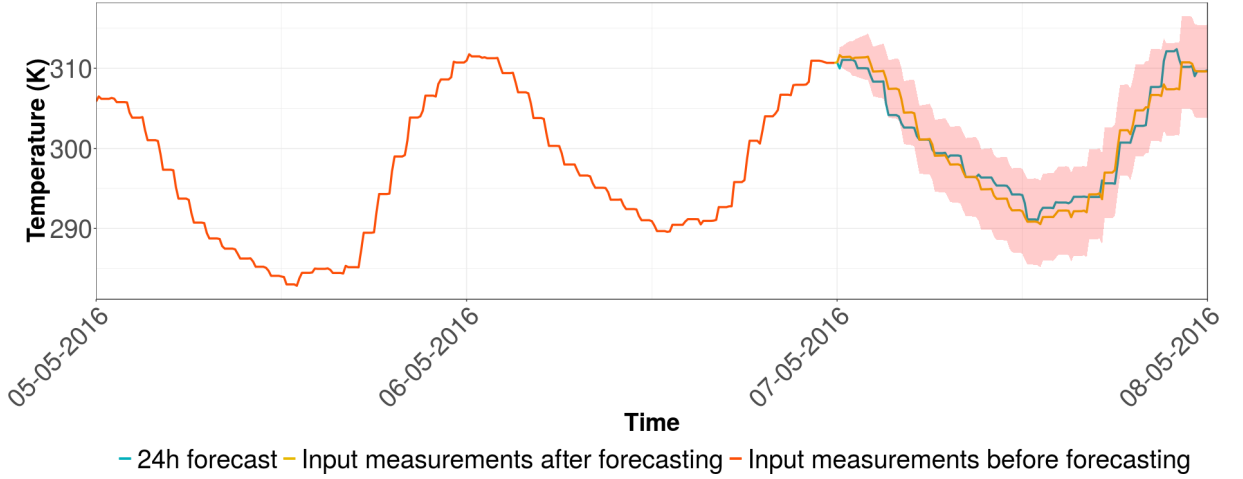

Figure D.1: A day forecasting with the SARIMA model used $(2, 0, 0) \times (0, 1, 1)_1$ and its comparison to actual measurements with the 95% confidence interval

Even though the forecast result is satisfying in most cases, it is more difficult when the meteorological conditions vary from one day to another, as shown in figure Fig. D.2. Nevertheless, the SARIMA model can be trained periodically to take into account such variations.
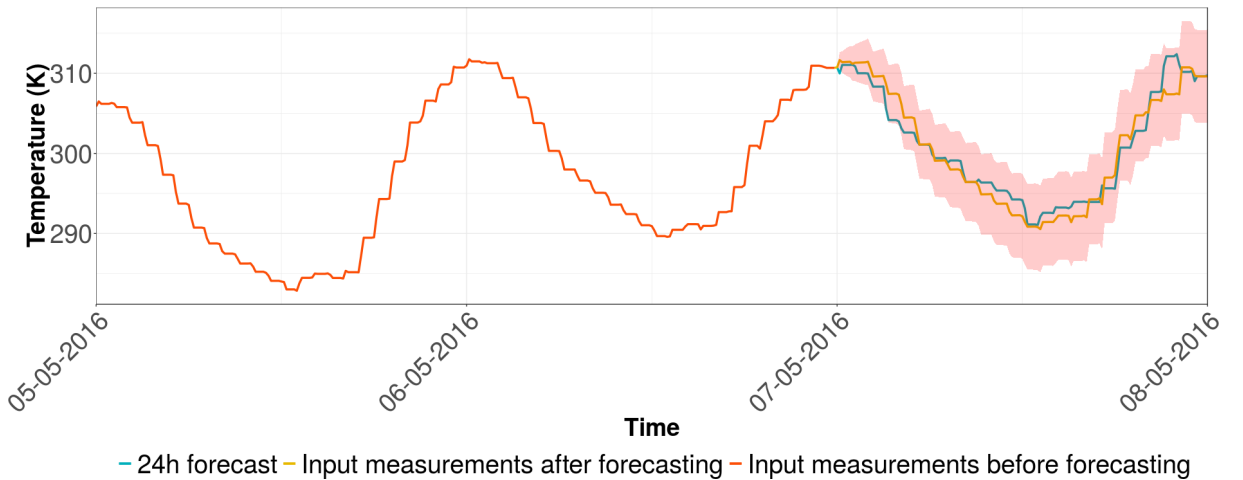

Figure D.2: A day forecasting with the SARIMA model used $(2, 0, 0) \times (0, 1, 1)_1$ and its comparison to actual measurements with the 95% confidence interval

Liste des publications

Publications dans des revues internationales avec comité de lecture

- T. Toullier, J. Dumoulin, L. Mevel, " Quantitative analysis of measurements bias due to environmental and spatial discretization in long term thermal monitoring of structures using outdoor infrared thermography ", Quantitative Infrared Thermography (QRT) Journal, **soumission en cours**, 2019.

Publication dans des conférences internationales avec acte et comité de lecture

- T. Toullier, J. Dumoulin, " Comparative study of moving train hot boxes pre-detection and axles counting by in-situ implementation of two infrared cameras", 3rd Quantitative InfraRed Thermography Conference, QIRT Asia 2019, Tokyo, 1-5 july 2019. Présentation par Thibaud Toullier
- T. Toullier, J. Dumoulin, L. Mevel, "Study of complementary multi-sensors data influence on infrared thermography measurements for in-situ long-term monitoring", SPIE 2019, Optical Metrology, Munich, Germany, 24-27 june 2019.
- T. Toullier, J. Dumoulin, L. Mevel, " Study of measurements bias due to environmental and spatial discretization in long term thermal monitoring of structures by infrared thermography ", 14 th Quantitative InfraRed Thermography Conference, QIRT 2018, Berlin, 25 June-29 June 2018. Présentation par Thibaud Toullier.
- T. Toullier, J. Dumoulin, L. Mevel, "Comparison of thermocouples surface measurements and infrared thermography for transport infrastructure thermal monitoring", 14th International Workshop on Advanced Infrared Technology and Applications, AITA 2017, September 27-29, 2017, Quebec City, Canada
- N. Le Touz, T. Toullier, J. Dumoulin, "Infrared thermography applied to the study of heated and solar pavement: from numerical modeling to small scale laboratory experiments", SPIE DCS 2017, Conférence Thermosense XXXIX, Anaheim, USA, 9-13 Avril 2017. Présentation par Thibaud Toullier.

Communications dans des conférences internationales avec acte

- T. Toullier, J. Dumoulin, L. Mevel, " Sensitivity of different methods for simultaneous evaluation of emissivity and temperature through multispectral infrared thermography simulation", Innovative instrumentations, techniques, geophysical methods and models for near surface geophysics, cities and transport infrastructures, EGU 2019, April 7-12, 2019, Vienna, Austria. Présentation poster.
- T. Toullier, J. Dumoulin, L. Mevel, "Design and study of a post-processing tool for long-term thermal infrared image sequences and multi-sensor data exploitation ", Innovative instrumentations, techniques, geophysical methods and models for near surface geophysics, cities and transport infrastructures, EGU 2018, April 8-13, 2018, Vienna, Austria. Présentation orale par Thibaud Toullier.

Publications dans des conférences nationales avec acte et comité de lecture

- T. Toullier, J. Dumoulin, L. Mevel, Étude et développement dun simulateur déchanges radiatifs dans des scènes 3D statiques et dynamiques surveillées par thermographie infrarouge ", 27ème Conférence de la Société Française de Thermique, SFT 2019, Nantes, 3-6 juin 2019
- T. Toullier, J. Dumoulin, L. Mevel, " Étude de sensibilité de différentes méthodes de séparation pour lévaluation simultanée de lémissivité et de la température par thermographie infrarouge multispectrale ", 26ème Conférence de la Société Française de Thermique, SFT 2018, Pau, 29 Mai-1 Juin 2018
- T. Toullier, J. Dumoulin, L. Mevel, " Étude comparative de deux approches, thermocouples intégrés et thermographie infrarouge, pour la surveillance thermique dune infrastructure de transport ", Conférence de la Société Française de Thermique, SFT 2017, Marseille, 30 Mai-2 Juin 2017.

Présentations en workshop ou séminaires sans acte

- Doctoriales du Département Cosys de l'IFSTTAR, présentation des travaux réalisés à ce jour, 3-4 Juillet 2018, IFSTTAR, Marne La Vallée.
- Doctoriales du Département Cosys de l'IFSTTAR, présentation des travaux réalisés à ce jour, 22-23 Juin 2017, IFSTTAR, Marne La Vallée.
- 27ème édition de la "Fête de la science" à Nantes

Titre: Caractérisation conjointe de la température et des propriétés radiatives des objets par thermographie infrarouge multispectrale

Mot clés : thermographie infrarouge, monitoring thermique, propriétés thermo-optiques, approche Bayésienne, filtres de Kalman, radiosités progressives

Résumé : L'utilisation de caméras infrarouges bas coûts pour la surveillance long-terme d'infrastructures est prometteuse grâce aux dernières avancées technologiques du domaine. Une mesure précise de la température des surfaces observées *in-situ* se heurte au manque de connaissance des propriétés radiatives de la scène. L'utilisation d'une instrumentation multicapteurs permet d'affiner le modèle de mesure afin d'obtenir une estimation plus précise de la température. *A contrario*, il est montré qu'il est toujours possible d'exploiter des données climatiques en ligne pour pallier un manque de capteur. Des méthodes Bayesiennes d'estimation conjointe d'émissivité et de température sont ensuite développées et comparées aux méthodes de la littérature. Un simulateur d'échanges radiatifs diffus de scènes 3D a été implémenté afin de tester ces différentes méthodes. Ce logiciel utilise l'accélération matérielle de la machine pour réduire les temps de calcul. Les résultats numériques obtenus mettent en perspective une utilisation avancée de la thermographie infrarouge multi-spectrale pour la surveillance de structures. Cette estimation conjointe permet alors d'obtenir un estimé de la température par thermographie infrarouge avec une incertitude connue.

Title: Simultaneous characterization of objects temperature and radiative properties through multispectral infrared thermography

Keywords : infrared thermography, thermal monitoring, thermo-optical properties, Bayesian methods, Kalman filters, progressive radiosities

Abstract : The latest technological improvements in low-cost infrared cameras have brought new opportunities for long-term infrastructures monitoring. The accurate measurement of surfaces' temperatures is facing the lack of knowledge of radiatives properties of the scene. By using multi-sensors instrumentation, the measurement model can be refined to get a better estimate of the temperature. To overcome a lack of sensors instrumentation, it is shown that online and free available climatic data can be used. Then, Bayesian methods to estimate simultaneously the emissivity and temperature have been developed and compared to literature's methods. A radiative exchange simulator of 3D scenes have been developed to compare those different methods on numerical data. This software uses the hardware acceleration as well as a GPGPU approach to reduce the computation time. As a consequence, obtained numerical results emphasized an advanced use of multi-spectral infrared thermography for the monitoring of structures. This simultaneous estimation enables to have an estimate of the temperature by infrared thermography with a known uncertainty.