

Universidade Estadual de Campinas
Instituto de Computação



Felipe de Castro Belém

Towards an Agnostic Superpixel Segmentation Framework

Rumo a um Arcabouço Agnóstico de Segmentação em Superpixels

CAMPINAS
2025

Felipe de Castro Belém

Towards an Agnostic Superpixel Segmentation Framework

Rumo a um Arcabouço Agnóstico de Segmentação em Superpixels

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação no âmbito do acordo de Cotutela firmado entre a Unicamp e a Université Gustave-Eiffel.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science under the double-diploma agreement between Unicamp and Université Gustave-Eiffel.

Supervisor/Orientador: Prof. Dr. Alexandre Xavier Falcão and Prof. Jean Cousty
Co-supervisor/Coorientador: Prof. Dr. Silvio Jamil Ferzoli Guimarães and Prof. Benjamin Perret

Este exemplar corresponde à versão final da Tese defendida por Felipe de Castro Belém e orientada pelos Profs. Drs. Alexandre Xavier Falcão e Jean Cousty

CAMPINAS
2025

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Belém, Felipe de Castro, 1994-
B411r Rumo a um arcabouço agnóstico de segmentação em superpixels / Felipe de Castro Belém. – Campinas, SP : [s.n.], 2025.

Orientador: Alexandre Xavier Falcão e Jean Cousty.
Coorientadores: Silvio Jamil Ferzoli Guimarães, Benjamin Perret.
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Computação.
Em cotutela com: Université Gustave-Eiffel, França.

1. Superpixels. 2. Segmentação de imagens. 3. Análise de imagem. 4. Multiescala. I. Falcão, Alexandre Xavier, 1966-. II. Guimarães, Silvio Jamil Ferzoli. III. Perret, Benjamin. IV. Cousty, Jean. V. Universidade Estadual de Campinas (UNICAMP). Instituto de Computação. VI. Título.

Informações complementares

Título em outro idioma: Towards an agnostic superpixel segmentation framework
Palavras-chave em inglês:
Superpixels
Image segmentation
Image analysis
Multiscale
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Alexandre Xavier Falcão [Orientador]
Jean Cousty
Silvio Jamil Ferzoli Guimarães
Benjamin Perret
Ewa Kijak
Nina Sumiko Tomita Hirata
Paulo André Vechiatto de Miranda
Thierry Géraud
Data de defesa: 11-02-2025
Programa de Pós-Graduação: Ciência da Computação

Objetivos de Desenvolvimento Sustentável (ODS)
Não se aplica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-6037-5977
- Currículo Lattes do autor: http://lattes.cnpq.br/0187395671366646



Universidade Estadual de Campinas
Instituto de Computação



Felipe de Castro Belém

Towards an Agnostic Superpixel Segmentation Framework

Rumo a um Arcabouço Agnóstico de Segmentação em Superpixels

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão (Supervisor)
  University of Campinas
- Prof. Dr. Jean Cousty (Supervisor)
  Université Gustave-Eiffel
- Prof. Dr. Silvio Jamil Ferzoli Guimarães (Co-supervisor)
  Pontifical Catholic University of Minas Gerais
- Prof. Dr. Benjamin Perret (Co-supervisor)
  Université Gustave-Eiffel
- Profa. Dra. Ewa Kijak
  Université de Rennes
- Profa. Dra. Nina Sumiko Tomita Hirata
  University of São Paulo
- Prof. Dr. Paulo André Vechiatto de Miranda
  University of São Paulo
- Prof. Dr. Thierry Géraud
  EPITA Research and Development Laboratory

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 11 de fevereiro de 2025

While having on one hand to respect the expectations and choices of the students, the educator also has the duty of not being neutral.
(FREIRE, Paulo)

Acknowledgements

First and foremost, I would like to thank my wife, Natália, for all the support during this journey. Without her, nothing would be possible in this doctorate. I extend this thanks to my relatives, especially to my parents, Lúcio and Leninha; my brother, Gustavo; my sister-in-law, Mariana; my mother- and father-in-law, Marília and Antônio; and also to Jéssica and Thiago. Also, I am grateful to all my friends, especially Matheus, Leonardo, Bárbara, Sarah, and Caroline, for helping me and for assisting in improving this thesis.

I am deeply grateful for the guidance and support of my advisors, Silvio, Falcão, Jean, and Benjamin. Their mentorship has been instrumental in my academic journey, from my Bachelor's studies, through my Master's, and now, in my PhD. Their expertise and encouragement have helped me develop as a researcher, and I am truly thankful for their contributions to my work.

This study was financed in part by Coordenação de Aperfeiçoamento de Ensino Superior (Capes) – Finance Code 001 –, by the Brazilian National Council for Scientific and Technological Development (CNPq), grants 407242/2021-0, 131000/2018-7, 306573/2022-9, 442950/2023-3, PQ 303808/2018-7, 310075/2019-0, and 304711/2023-3. Also, this study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number 2023/14427-8, 2014/12236-1, and 2013/07375-0 –, and by the Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) – Grant APQ-01079-23, Grant APQ-05058-23, PPM-00006-18, and PCE-00417-24. Their funding has been crucial in enabling me to pursue my research. I also thank Universidade Estadual de Campinas (UNICAMP), Pontifícia Universidade Católica de Minas Gerais (PUC-Minas), and École Supérieure d'Ingénieurs en Électrotechnique et Électronique (ESIEE - Paris) for providing me with the opportunity to develop my research within their institutions, particularly as a student or advisee. I would like to thank the colleagues I met during this journey from France, Tunisia, Algeria, Madagascar, Uruguay, Argentina, and China.

Finally, I am grateful for this opportunity, which was only viable due to government funding. Without such programs, I would never have been able to grow as a researcher and professor and develop a wide network to produce high-quality papers. From my personal experience, more funding is utterly necessary so that several other researchers may experience what I have experienced and to allow future researchers the necessary conditions for their integral dedication.

Resumo

Um problema clássico em segmentação de imagens envolve o particionamento de múltiplos objetos em partes disjuntas para que a delineação destes seja precisamente obtida pelo seu agrupamento. Essa tarefa, nomeada por segmentação em superpixels, apresenta tal propriedade enquanto reduz a ordem de magnitude da entrada, de pixels para grupos de pixels conexos e similares (*i.e.*, superpixels). Além disso, ao gerar grupos de pixels similares, superpixels também apresentam uma informação com mais significado que seus pares atômicos, enquanto reduz a redundância intrínseca de pixels similares e próximos. Consequentemente, algoritmos em superpixel vêm sendo utilizados amplamente como passo intermediário para a resolução de múltiplos problemas.

Entretanto, algoritmos do estado da arte se deparam com um grande desafio de segmentação efetiva e eficiente, independente das características do objeto ou fundo, e atendimento às demandas do usuário acerca da forma e disposição, por exemplo. Ou seja, não são capazes de serem adaptados facilmente e intuitivamente para qualquer domínio possível ou de gerar a saída esperada da segmentação, indicando que não são agnósticos de domínio. Essa dificuldade é normalmente resultante da concepção e otimização de soluções em domínios similares, que acabam por internalizar e especializar o algoritmo para segmentar um objeto particular em um contexto e saída específicos. Esta situação é normalmente encontrada em abordagens clássicas e em mais recentes, especialmente aquelas que consideram uma informação prévia do objeto, seja através de aprendizado profundo ou via representação intermediária, como mapas de saliência de objeto.

Por exemplo, se desejar uma sequência de segmentações com características específicas, diversas abordagens necessitariam de múltiplas execuções, uma para cada. Na realidade, tal série de segmentações, denominada *segmentação multiescala*, é uma abordagem usual para representar as características do objeto (ou da imagem) em diferentes escalas de percepção. Isto é, escalas mais finas, contendo uma grande quantidade de regiões, capturam os detalhes mais finos, enquanto aquelas mais rudimentares aglomeram redundâncias e provêem uma informação mais significativa acerca da região e seu entorno.

Além disso, caso o usuário esteja somente interessado em um objeto, apenas poucos algoritmos incorporam tal informação por meio de mapas de saliência. Nestes algoritmos, a localização do objeto é representada pelas intensidades dos pixels, de tal forma que tons mais claros indicam maior probabilidade de pertencimento ao objeto, e mais escuros possuem maior probabilidade de pertencimento ao fundo. Ainda que estes métodos apresentem alta acurácia em delineamento, eles também são costumeiramente mais lentos que abordagens clássicas. Métodos que consideram informação de objeto por meio de técnicas de aprendizado profundo também sofrem de deterioração de velocidade quando uma única segmentação é requisitada pelo usuário. Ademais, estes normalmente apresentam um desempenho moderado em delineamento, na qual é posto em questionamento visto o esforço gasto no aprendizado das características do objeto. No fim, mais pesquisas são necessárias para gerar superpixels usando aprendizado profundo.

Finalmente, outro problema surge caso o usuário possua uma segmentação multiescala, mas é ignorante da possibilidade de explorar propriedades hierárquicas para otimização e simplificação de tarefas. Pode-se notar que se é garantido que o objeto possa ser construído de maneira precisa através de uma simples seleção de nós em uma hierarquia, a tarefa de segmentação interativa seria demasiadamente facilitada. Mas, apenas alguns algoritmos de segmentação em superpixels são conhecidos por serem hierárquicos. De outra forma, a maioria dos algoritmos, quando é requisitada a geração de uma segmentação multiescala, não garantem um princípio crucial em segmentação hierárquica: o princípio da localidade. Tal princípio pode ser informalmente definido como a presença das bordas em escalas mais rudimentares nas escalas mais finas (*i.e.,* que possuem menos e mais regiões, respectivamente), indicando um aninhamento entre as regiões de ambos.

Até onde se sabe, não há proposta para estimar a hierarquicidade, ou a semelhança de uma segmentação multiescala para uma hierárquica. Se feito de maneira ingênua, é possível notar que a tarefa é combinatória. Também, se considerar a possibilidade de inserir ou remover escalas para maximizar a semelhança, o problema atinge um alto nível de dificuldade. Trabalhos existentes para avaliação de hierarquias normalmente buscam um objeto, o que significa a existência de uma máscara de objeto. Claramente, a referida tarefa é baseada em avaliar contornos e, para isso, pode-se ponderar os contornos baseado nos objetos definidos na máscara. Entretanto, o comportamento aninhado entre escalas não é relacionado a objeto, e ponderar a relevância das bordas não auxilia em determinar se uma segmentação multiescala favorece o aninhamento de regiões entre as escalas.

Nesta tese, abordamos ambos os desafios propondo várias contribuições. Para a tarefa de domínio agnóstico, propomos um novo arcabouço de segmentação em superpixels, denominada Superpixels through Iterative CLEarcutting (SICLE), que generaliza duas outras contribuições desta tese, Dynamic and Iterative Spanning Forest (DISF) e Object-based DISF (ODISF). No SICLE, três etapas independentes são definidas: (i) superamostragem de sementes; (ii) geração de superpixels usando o arcabouço da Image Foresting Transform (IFT); e (iii) remoção de sementes. A partir de (i), onde uma quantidade significativamente alta de sementes é selecionada, as etapas (ii) e (iii) são executadas para gerar superpixels a partir de um conjunto aprimorado de sementes até atingir o número desejado de superpixels.

Ao selecionar um número significativamente maior de pontos iniciais (*i.e.*, *sementes*) do que o número desejado de superpixels, o objetivo é aumentar a probabilidade de cada objeto possível conter pelo menos uma semente interna. Por outro lado, muitas abordagens do estado da arte selecionam um número de sementes aproximadamente igual ao número final de superpixels. Se o objeto e o número de sementes forem suficientemente pequenos, a estratégia de amostragem poderá não garantir essa propriedade para todos os objetos. Consequentemente, a falta de sementes está fortemente associada à deterioração do delineamento devido à concorrência reduzida.

Após a amostragem, o SICLE gera os superpixels usando a *Image Foresting Transform* (IFT) a partir do conjunto de sementes. A IFT é um arcabouço para o desenvolvimento de operadores de imagem com base na conectividade e pode ser calculada usando uma generalização do algoritmo de caminho mais curto de Dijkstra para considerar diferentes funções de conectividade. Assim, no IFT, cada superpixel é uma árvore cujos caminhos para sua semente são ótimos. Em outras palavras, uma árvore conquista pixels mais semelhantes a ela do que qualquer outra árvore. Dependendo da função de conectividade, o algoritmo pode não garantir a otimização teórica dos caminhos, mas ainda produz uma floresta de abrangência enraizada nas sementes, cuja aplicabilidade e desempenho são bem

relatados na segmentação de superpixels.

Após a amostragem, SICLE gera superpixels através da *Image Foresting Transform* (IFT) a partir do conjunto de sementes. A IFT é um arcabouço para a construção de operadores de imagens baseados em conectividade, e pode ser calculada utilizando uma generalização do algoritmo de Dijkstra para caminhos mínimos, para considerar diferentes funções de conectividade. Assim, na IFT, cada superpixel é uma árvore cujos caminhos até a semente são ótimos. Em outras palavras, uma árvore conquista os pixels mais similares à ela do que à qualquer outra árvore. A depender da função de conectividade, o algoritmo pode não garantir a otimalidade teórica dos caminhos, mas ainda é capaz de gerar uma floresta geradora enraizada nas sementes, cuja aplicabilidade e desempenho é bem conhecido em segmentação em superpixels.

Como resultado da superamostragem, o terceiro passo do arcabouço consiste em selecionar uma porção das sementes que não são necessárias para o acurado delineamento, enquanto mantém aquelas que são cruciais para tal objetivo. Neste trabalho, chamaremos as primeiras de *irrelevantes* e as últimas como *relevantes*. Visto que a precisa identificação da relevância de sementes é difícil, SICLE permite tal informação manifestar através das iterações. Isto é, ao "perturbar" as sementes através da remoção parcial de sementes irrelevantes, as relevantes continuam para uma nova competição pelos pixels "órfãos" e, portanto, cada semente possui uma nova chance de exibir sua relevância. Nesta estratégia, uma semente relevante é continuamente indentificada como tal em cada iteração.

Entretanto, caso um objeto específico seja desejado, pode-se notar que o critério para determinação de relevância de uma semente é prejudicada por tal subjetividade. Normalmente, a intenção é aumentar o número de sementes dentro ou próximas ao objeto de interesse. Em razão disso, o critério pode identificar falsos positivos, que podem ser percebidos como sementes relevantes para objetos irrelevantes. Logo, a relevância está sujeita à penalização sempre que tal informação é fornecida pelo mapa de saliência de objeto.

Como se pode notar, o critério de parada do SICLE consiste em obter o número desejado de superpixels. Na maioria dos métodos do estado da arte, um número obrigatório de iterações é demandado para produzir a segmentação e, em algumas circunstâncias, ater à tal quantidade pode ser desnecessário. Por exemplo, quando a quantidade desejada de superpixels é signficativamente baixa, a competição é relaxada, e conquistar pixels dissimilares é corriqueiro. Para isso, a abordagem padrão é realocar sementes para o mínimo do grupo para minimizar a dissimilaridade interna ao superpixel, seja relacionada às características ou à posição espacial, por exemplo. Porém, para maiores quantidades de superpixel, a competição é extrema, levando à superpixels altamente especializados ou, de outra forma, reduzindo o agrupamento de pixels dissimilares. Logo, realocar sementes pode não fornecer uma melhoria significativa.

Portanto, o objetivo do SICLE é considerar uma curva exponencial para determinar a quantidade de sementes a serem removidas a cada iteração. A motivação por trás de tal curva específica está associada ao comportamento intrínseco da competição na segmentação em superpixels. Partindo de um conjunto significativamente maior de sementes, as quais a maioria é irrelevante, remover uma grande porção usando um critério acurado afetaria minimamente o desempenho da segmentação. Entretanto, a medida que o número de sementes decai e se aproxima da quantidade final desejada de superpixels, remover uma grande porção, por exemplo uma parte do objeto, poderia acarretar em uma grande desestabilização na competição. Tal remoção favoreceria "vazamentos" severos e, consequentemente, a estratégia busca reduzir a quantidade de sementes a serem removidas até

obter a quantidade desejada de superpixels. Note que esta estratégia permite computações mais rápidas para maiores quantidades de superpixels enquanto executa a quantidade necessária de iterações para manter o efetivo delineamento para menores quantidades de superpixels. Ademais, a partir desta curva, SICLE é capaz de gerar uma segmentação multiescala em uma única execução ao simplesmente apresentar as segmentações intermediárias a medida que o número de sementes restantes é reduzido e atinge cada valor especificado pelo usuário.

O agnosticismo de domínio é alcançado no SICLE, principalmente através dos passos (ii) e (iii), os quais o usuário pode fornecer uma função objetivo para otimização. É possível notar que, visto que a superamostragem aumenta as chances de se selecionar uma semente em cada objeto, não há necessidade de se adaptar o passo (i) para novos domínios. Considerando a geração de superpixels, o passo (ii), sob diferentes funções de conectividade e de estimação do custo da aresta, fornece o ferramentário necessário para se produzir superpixels regulares ou irregulares baseado em algum fator de regularização, quando demandado. Ou seja, a IFT é responsável, principalmente, pela morfologia dos superpixels. Finalmente, quando se seleciona sementes relevantes, nota-se que a identificação acurada está associada diretamente com o domínio e com as características do objeto, indicando que viéses são cruciais para uma estimação altamente especializada e efetiva de relevância. Diferentes objetos com características distintas podem requerer diferentes estimações de relevância, e domínios diferentes podem demandar outros critérios de relevância.

Resultados experimentais mostram que as variantes do SICLE superam diversos métodos do estado da arte em diversos domínios considerando velocidade e acurácia, enquanto gera uma sequência de segmentações em uma única execução. Em outros termos, SICLE é capaz de manter seu desempenho efetivo independente do domínio e, se as condições são favoráveis, como o fornecimento de um mapa de saliência acurado, nossos métodos são capazes de atingir o limite superior de seu desempenho em apenas duas iterações. Além disso, versões compactas do SICLE podem explorar a qualidade do mapa de saliência dado para fornecer uma propriedade não antes vista na segmentação em superpixels: superpixels altamente compactos e com alta aderência às bordas.

Ainda assim, por mais que aparente, a referida multiescala não é garantida de ser hierárquica pelo SICLE visto as violações de localidade. Isto é, apesar do SICLE ser capaz de produzir uma sequência de segmentações estritamente decrescente, o que garante o princípio da localidade definido em segmentações hierárquicas, a presença das bordas da escala precedente não são garantidas na próxima escala. Por exemplo, ao alterar a competição para a segmentação subsequente, caminhos ótimos antes "bloqueados" por um superpixel irrelevante podem, agora, conquistar diversos pixels, incluindo aqueles previamente pertencentes à outro na escala anterior. Como consequência, bordas antes detectadas podem ser alteradas, violando assim o princípio da localidade e impedindo o aninhamento entre regiões.

Logo, buscando prover uma medida para determinar a semelhança de uma segmentação multiescala para com uma hierarquia, nós definimos os conceitos de núcleo e de cobertura, que podem ser brevemente definidos como as regiões na escala anterior que são, respectivamente, aninhados ou que sobrepõem a região considerada na escala seguinte. Baseado em ambos, nós estudamos e catalogamos oito casos possíveis quando analisadas as regiões entre pares de segmentações subsequentes: (i) separação completa; (ii) deflação e separação; (iii) deflação completa; (iv) estabilidade; (v) instabilidade; (vi) inflação completa; (vii) inflação e junção; (viii) junção completa. Nestes casos, uma hie-

rarquia é composta estritamente de junções completas ou de estabilidade para sequências com número decrescente de regiões, e é análogo para separação completa ou estabilidade quando o número de regiões é crescente.

A partir desses casos, nós concebemos três medidas para estimar a hierarquicidade de uma segmentação multiescala: (i) nestedness; (ii) inflation ratio; e (iii) refinement error. A primeira medida estima, baseado no núcleo de cada região, a razão de regiões na escala precedente aninhadas na escala subsequente. Ao contrário, a segunda medida calcula a razão da inflação das regiões na escala anterior para construir as regiões da seguinte. Para isso, são excluídas as contribuições das regiões pertencentes à algum núcleo da escala seguinte. Finalmente, a terceira medida é inspirada em uma métrica popular em segmentação em superpixels e mede a taxa para modificação das regiões da escala anterior para que todas estejam aninhadas na escala seguinte. Ou seja, ela avalia o esforço mínimo para alteração: ou removendo a interseção entre regiões quando é suficientemente baixa, ou juntando regiões quando a interseção é suficientemente alta.

Por definição, uma hierarquia apresentaria nestedness máximo, e valores mínimos de inflation ratio e refinement error. Tal propriedade está também presente entre segmentações subsequentes na hierarquia, significando que se quaisquer duas partições subsequentes exibem um comportamento aninhado, é possível concluir que a série completa exibe um comportamento aninhado e, finalmente, que a multiescala é uma hierarquia. Chegar em tal conclusão a partir das nossas contribuições é uma forma mais eficiente do que avaliar todas os pares possíveis entre escalas. Também, é possível notar que, para calcular tais medidas, assumimos que a escala precedente é a "máscara" de sua respectiva escala subsequente.

A partir dos nossos resultados, podemos verificar que nossas propriedades são verdadeiras e, mais importante, que é possível verificar se uma multiescala é uma hierarquia. Ademais, quando não é o caso, combinando os resultados de todas as medidas oferece uma ampla e clara perspectiva da natureza e gravidade das violações hierárquicas existentes na segmentação em multiescala, que a impede de ser hierárquica. Por exemplo, "quase hierarquias" podem ser identificadas em dois casos: alto nestedness e baixos inflation ratio e refinement error, e baixos nestedness, inflation ratio, e refinement error. Enquanto o primeiro apresenta uma perspectiva clara do comportamento aninhado, o último indica que, apesar do aninhamento não estar garantido, a inflação é mínima e a correção das bordas também é mínima.

Abstract

A classical problem in image segmentation involves partitioning multiple objects into disjoint parts so that their delineation can be accurately achieved by their grouping. Such task, named superpixel segmentation, offers such property while reducing the order of magnitude from the input, from pixels to groups of connected and similar pixels (*i.e.*, superpixels). Furthermore, by generating groups of similar pixels, superpixels also confer more semantic information than their atomic counterparts while reducing the intrinsic redundancy of nearby similar pixels. Consequently, superpixel algorithms have been widely used as an intermediary step for solving multiple problems.

However, state-of-the-art algorithms face a significant challenge of effective and efficient segmentation irrespective of the objects and background's characteristics and attending to the user's desires regarding, for instance, shape and disposition. Simply put, they often cannot be easily and straightforwardly adapted to any possible domain or to generate the expected segmentation output, meaning they are not domain-agnostic. Such difficulty is often a result of conceiving and optimizing solutions in similar domains, which internalizes and highly specializes the algorithm for segmenting a particular object in a specific context and output. This issue is commonly found in classical approaches and in more recent ones, especially those that consider prior object information either through deep-learning or through an intermediary representation, such as object saliency maps.

For example, if one desires a series of segmentations with specific characteristics, several approaches would require multiple executions, one for each. In fact, such a series of segmentations, named *multiscale segmentation*, is a common approach to representing the object's (or image's) characteristics in different scales of perception. That is, finer scales, containing a high quantity of regions, capture the finer details, while coarser ones agglomerate redundancies and provide more semantic information about a region and its surroundings.

Also, if the user is only interested in one object, only a few algorithms incorporate such information through saliency maps. In these algorithms, the object's location is represented by the pixel intensities such that brighter values indicate higher object probability and darker ones have higher background probability. Even though these methods present high delineation accuracy, they are often significantly slower than classical approaches. Methods that consider object information through deep-learning techniques also suffer from speed deterioration when a single segmentation is demanded by the user. Furthermore, they usually present fair delineation performance, which is questionable due to their effort in learning the object's characteristics. In the end, more research is required to generate superpixels using deep-learning.

Finally, another problem arises if the user possesses a multiscale segmentation but is unaware of the possibility of exploring hierarchical properties for optimization and task simplification. One can see that if it is ensured that the object can be accurately built with a simple node selection in the hierarchy, the interactive segmentation task is

significantly facilitated. Yet, only a few algorithms in superpixel segmentation are known to be hierarchical. In other terms, most algorithms, when required to generate a multiscale segmentation, cannot guarantee one crucial principle in hierarchical segmentation: the locality principle. Such principle can be informally defined as the existence of borders in the coarser scale to the finer one, indicating the nesting behavior regions of both.

To the best of our knowledge, there is no proposal for estimating the hierarchiness, or the resemblance of a multiscale segmentation to a hierarchical one. If done naïvely, one may see that the task is combinatorial. Furthermore, if considering the possibility of inserting or removing scales to maximize resemblance, the problem achieves a high level of difficulty. Existing works for evaluating hierarchies often are for object purposes, which means the existence of an object ground truth. Clearly, the referred task is contour-based evaluation, and for that, one may weigh the contours based on the objects defined in the ground truth. However, the nesting behavior between scales is not object-related, and weighting the relevance of the borders does not assist in determining the nesting behavior of the multiscale segmentation.

In this thesis, we address both challenges by proposing several contributions. For the domain-agnostic task, we propose a novel superpixel segmentation framework, named Superpixels through Iterative CLEarcutting (SICLE), which generalizes two other contributions of this thesis, Dynamic and Iterative Spanning Forest (DISF) and Object-based DISF (ODISF). In SICLE, three independent steps are defined: (i) seed oversampling; (ii) superpixel generation using the Image Foresting Transform (IFT) framework; and (iii) seed removal. From (i), where a significantly high amount of seeds is selected, steps (ii) and (iii) are performed for generating superpixels from a refined seed set until achieving the desired number of superpixels.

By selecting a significantly higher number of initial points (*i.e.*, *seeds*) than the desired number of superpixels, the goal is to increase the probability of every possible object containing at least one within. In contrast, many state-of-the-art approaches select a number of seeds approximately to the final number of superpixels. If the object and the number of seeds are sufficiently small, the sampling strategy may not guarantee such property for all the objects. Consequently, the lack of seeds is strongly associated with deterioration in delineation due to the reduced competition.

After sampling, SICLE computes superpixels using the *Image Foresting Transform* (IFT) from the seed set. The IFT is a framework for developing image operators based on connectivity and may be computed using a generalization of Dijkstra's shortest-path algorithm for considering different connectivity functions. Thus, in IFT, each superpixel is a tree whose paths to its seed are optimum. In other words, a tree conquers pixels more closely similar to it than any other tree. Depending on the connectivity function, the algorithm may not guarantee theoretical optimality of the paths, but still outputs a spanning forest rooted in the seeds, whose applicability and performance is well-reported in superpixel segmentation.

As a result of oversampling, the third step of the pipeline consists of selecting a portion of seeds that are not needed for accurate segmentation while maintaining those that are crucial for such a goal. In this work, we term the former as *irrelevant* seeds, whereas the latter as *relevant*. Since accurate identification of seed relevance is challenging, SICLE lets such information manifest throughout the iterations. That is, by "perturbing" seeds through partial removal of irrelevant ones, the remaining ones are left to compete for the "orphan" pixels, and, therefore, each seed has a new chance of exhibiting its relevance. In this strategy, a relevant seed is continuously identified as such in each iteration.

However, if a specific object is desired, one may see that the criterion for determining the relevance of a seed is undermined by such subjectivity. Often, the intent is to increase the number of seeds within or near the desired object. For this reason, the criterion may identify false positives, which can be seen as relevant seeds for irrelevant objects. Therefore, the relevance is subjected to penalization whenever such information is provided by the object saliency map.

As one may note, the stopping criterion of SICLE is to obtain the desired number of superpixels. In most state-of-the-art methods, a strict number of iterations are required to produce the segmentation, and, in some circumstances, attaining such quantity may be unnecessary. For instance, when the desired number of superpixels is significantly low, competition is relaxed, and conquering dissimilar pixels is usual. For that, the common approach is to reallocate seeds to the group's minimum to minimize intra-superpixel dissimilarity, feature-related or spatial-related, for instance. However, for higher superpixel quantities, the competition is extreme, leading to highly specialized superpixels or, in other terms, reducing the grouping of dissimilar pixels. Thus, reallocating seeds may not provide significant improvement.

Therefore, the goal of SICLE is to consider an exponential curve to determine the number of seeds to be removed in each iteration. The motivation behind such a specific curve is associated with the intrinsic competition behavior in superpixel segmentation. From a significantly greater set of seeds, most of which are irrelevant, removing a large portion using an accurate criterion would minimally impact the segmentation performance. However, as the number of seeds decreases and approximates the desired final number of superpixels, removing a large portion, for instance, an object part, could lead to a major destabilization in competition. Such removal would favor severe "leakings" and, consequently, the strategy is to decrease the number of seeds to be removed until reaching the desired quantity of superpixels. Note that this strategy allows faster computations for a high number of superpixels while performing the necessary number of iterations to sustain its accurate delineation for smaller superpixel quantities. Furthermore, from such a curve, SICLE is able to generate a multiscale segmentation on-the-fly by simply outputting the intermediary segmentations as the number of remaining seeds is reduced and achieves each value requested by the user.

Domain agnosticism is achieved in SICLE, primarily through steps (ii) and (iii), where the user may provide an objective function for optimization. One can see that since oversampling increases the chances of selecting a seed in every object, there is no need to adapt step (i) to new domains. In terms of superpixel generation, step (ii), through different connectivity and edge-cost functions, provides the necessary toolkit for producing either regular or irregular superpixels based on some regularity factor whenever it is demanded. Simply put, IFT handles mostly the superpixels' morphology. Finally, when selecting relevant seeds, one can see that accurate identification is associated directly with the domain and the object's characteristics, meaning biases are crucial for a highly specialized and effective relevance estimation. Different objects with distinct characteristics may require different relevance estimations, and different domains may demand other relevance criteria.

Experimental results show that SICLE variants surpass several state-of-the-art algorithms concerning speed and accuracy for distinct domains while generating a series of segmentations in a single execution. More precisely, SICLE is able to sustain its effective performance irrespective of the domain, and if conditions are favorable, such as providing an accurate saliency map, our methods are able to achieve its upper bound performance

in only two iterations. Moreover, SICLE compact variants may exploit the quality of the given saliency map to provide a property unseen in superpixel segmentation: highly compact and highly border-adherent superpixels.

Still, resembling as it may appear, the latter multiscale is not ensured to be hierarchical by SICLE due to locality violations. That is, although SICLE is capable of producing a strictly decreasing sequence of segmentations, which guarantees the causality principle stated for hierarchical segmentation, the borders of the preceding scale are not ensured to be in the next scale. As an example, by changing competition for the subsequent iteration, optimal paths previously "blocked" by an irrelevant superpixel may now be able to conquer several pixels, including those belonging to another in the previous scale. Consequently, previously detected borders may now be reshaped, violating the locality principle and impeding nestedness between regions.

Therefore, in order to provide a measure for determining the resemblance of a multiscale segmentation to a hierarchy, we defined the concepts of nucleus and cover, which can be briefly defined as those regions in the preceding scale that are, respectively, nested or overlapping the considered region of the succeeding scale. Based on both, we studied and cataloged eight possible cases when analyzing regions in between pairwise subsequent segmentations: (i) full split; (ii) deflate and split; (iii) full deflation; (iv) stability; (v) instability; (vi) full inflation; (vii) inflation and merge; (viii) full merge. In such cases, a hierarchy is strictly composed of full merge or stability operations for sequences with a decreasing number of regions, and it is analogous to a full split or stability of an increasing number of regions.

From such cases, we conceived three measures for estimating the hierarchiness of a multiscale segmentation: (i) nestedness; (ii) inflation ratio; and (iii) refinement error. The first measure estimates, based on the nucleus of each region, the ratio of regions in the preceding scale nested in the subsequent scale. On the other hand, the second measure computes the rate of inflation of the regions in the preceding scale to build the regions of the next one. For that, it excludes the contribution of those regions belonging to some nucleus of the subsequent scale. Finally, the third metric is inspired by a popular superpixel metric and measures the rate for modifying the regions of the preceding scale so that they all be nested in the subsequent scale. In simple terms, it evaluates the minimum effort to reshape: either removing the intersection between regions when it is sufficiently low or totally merging regions when the intersection is sufficiently high.

By definition, a hierarchy would present maximum nestedness, minimum inflation ratio and refinement error. Such property is present also between subsequent segmentations in the hierarchy, meaning that if every two subsequent partitions exhibit nesting behavior, one can conclude that the whole series exhibit a nesting behavior, and, finally, that the multiscale is a hierarchy. Achieving such conclusion through our contributions is a much more efficient approach than evaluating all possible pairwise comparison between scales. Also, as one may note, for computing these measures, we assume that the preceding scale is the "ground truth" of its respective subsequent scale.

From our results, we can verify that our properties hold and, most importantly, that it is possible to verify whether a multiscale is a hierarchy. Moreover, when this is not the case, combining the results of all measures offers a broad and clear perspective of the nature and extent of the hierarchical violations existing in the multiscale segmentation that prevent it from being hierarchical. For instance, "quasi-hierarchies" can be identified in two cases: high nestedness and low inflation ratio and refinement error, and low nestedness, inflation ratio, and refinement error. While the former offers a clearer perspective on the nesting

behavior, the latter indicates that, although nesting is not ensured, inflation is minimal and the need for correcting borders is, also, minimal.

Résumé

Un problème classique de la segmentation d'images consiste à diviser plusieurs objets en parties disjointes de sorte que leur délimitation puisse être réalisée avec précision par leur regroupement. Cette tâche, appelée segmentation en superpixels, offre cette propriété tout en réduisant l'ordre de grandeur de l'entrée, des pixels à des groupes de pixels connectés et similaires (*i.e.*, superpixels). En outre, en générant des groupes de pixels similaires, les superpixels confèrent également plus d'informations sémantiques que leurs homologues atomiques tout en réduisant la redondance intrinsèque des pixels similaires proches. Par conséquent, les algorithmes de superpixels ont été largement utilisés comme étape intermédiaire pour résoudre de nombreux problèmes.

Toutefois, les algorithmes de pointe sont confrontés à un défi de taille, à savoir une segmentation efficace et efficiente, quelles que soient les caractéristiques des objets et de l'arrière-plan et en tenant compte des souhaits de l'utilisateur en ce qui concerne, par exemple, la forme et la disposition. En d'autres termes, ils ne peuvent souvent pas être adaptés facilement et directement à n'importe quel domaine possible ou pour générer le résultat de segmentation attendu, ce qui signifie qu'ils ne sont pas agnostiques par rapport au domaine. Cette difficulté résulte souvent de la conception et de l'optimisation de solutions dans des domaines similaires, ce qui internalise et spécialise fortement l'algorithme de segmentation d'un objet particulier dans un contexte et un résultat spécifiques. Ce problème se retrouve fréquemment dans les approches classiques et dans les approches plus récentes, en particulier celles qui prennent en compte des informations préalables sur les objets, soit par l'apprentissage profond, soit par une représentation intermédiaire, telle que les cartes de saillance des objets.

Par exemple, si l'on souhaite obtenir une série de segmentations présentant des caractéristiques spécifiques, plusieurs approches nécessiteraient plusieurs exécutions, une pour chacune d'entre elles. En fait, une telle série de segmentations, appelée « segmentation multiéchelle », est une approche courante pour représenter les caractéristiques de l'objet (ou de l'image) à différentes échelles de perception. En d'autres termes, les échelles plus fines, qui contiennent une grande quantité de régions, capturent les détails les plus fins, tandis que les échelles plus grossières agglomèrent les redondances et fournissent davantage d'informations sémantiques sur une région et son environnement.

Par ailleurs, si l'utilisateur ne s'intéresse qu'à un seul objet, seuls quelques algorithmes intègrent cette information par le biais de cartes de saillance. Dans ces algorithmes, l'emplacement de l'objet est représenté par l'intensité des pixels, de sorte que les valeurs les plus claires indiquent une probabilité plus élevée pour l'objet et les plus foncées une probabilité plus élevée pour l'arrière-plan. Bien que ces méthodes offrent une grande précision de délimitation, elles sont souvent beaucoup plus lentes que les approches classiques. Les méthodes qui prennent en compte les informations sur les objets par le biais de techniques d'apprentissage profond souffrent également d'une détérioration de la vitesse lorsqu'une segmentation unique est demandée par l'utilisateur. En outre, elles présentent générale-

ment des performances de délimitation équitables, ce qui est discutable en raison de leur effort d'apprentissage des caractéristiques de l'objet. En fin de compte, des recherches supplémentaires sont nécessaires pour générer des superpixels à l'aide de l'apprentissage profond.

Enfin, un autre problème se pose si l'utilisateur dispose d'une segmentation multi-échelle mais ignore la possibilité d'explorer les propriétés hiérarchiques à des fins d'optimisation et de simplification de la tâche. On peut constater que si l'on s'assure que l'objet peut être construit avec précision par une simple sélection de nœuds dans la hiérarchie, la tâche de segmentation interactive est considérablement facilitée. Pourtant, seuls quelques algorithmes de segmentation en superpixels sont connus pour être hiérarchiques. En d'autres termes, la plupart des algorithmes, lorsqu'ils doivent générer une segmentation multi-échelle, ne peuvent pas garantir un principe crucial de la segmentation hiérarchique : le principe de localité. Ce principe peut être défini de manière informelle comme l'existence de frontières entre l'échelle la plus grossière et l'échelle la plus fine, indiquant le comportement d'imbrication des régions des deux échelles.

À notre connaissance, il n'existe aucune proposition pour estimer la hiérarchisation ou la ressemblance d'une segmentation multi-échelle avec une segmentation hiérarchique. Si l'on procède de manière naïve, on peut constater que la tâche est combinatoire. En outre, si l'on considère la possibilité d'insérer ou de supprimer des échelles pour maximiser la ressemblance, le problème atteint un niveau de difficulté élevé. Les travaux existants pour l'évaluation des hiérarchies sont souvent destinés à des objets, ce qui implique l'existence d'une vérité de terrain. Il est clair que la tâche visée est l'évaluation basée sur les contours, et pour cela, on peut pondérer les contours sur la base des objets définis dans la vérité de base. Cependant, le comportement d'imbrication entre les échelles n'est pas lié à l'objet, et la pondération de la pertinence des frontières n'aide pas à déterminer le comportement d'imbrication de la segmentation multi-échelle.

Dans cette thèse, nous relevons ces deux défis en proposant plusieurs contributions. Pour la tâche agnostique, nous proposons un nouveau cadre de segmentation des superpixels, appelé Superpixels through Iterative CLEarcutting (SICLE), qui généralise deux autres contributions de cette thèse, Dynamic and Iterative Spanning Forest (DISF) et Object-based DISF (ODISF). Dans SICLE, trois étapes indépendantes sont définies : (i) le suréchantillonnage des graines ; (ii) la génération de superpixels à l'aide de la transformation forestière de l'image (IFT) ; et (iii) l'élimination des graines. À partir de l'étape (i), au cours de laquelle un nombre significativement élevé de graines est sélectionné, les étapes (ii) et (iii) sont exécutées pour générer des superpixels à partir d'un ensemble de graines affiné jusqu'à ce que le nombre de superpixels souhaité soit atteint.

En sélectionnant un nombre de points initiaux (*i.e.*, *seeds*) significativement plus élevé que le nombre de superpixels souhaité, l'objectif est d'augmenter la probabilité que chaque objet possible contienne au moins un élément. En revanche, de nombreuses approches de pointe sélectionnent un nombre de graines correspondant approximativement au nombre final de superpixels. Si l'objet et le nombre de graines sont suffisamment petits, la stratégie d'échantillonnage peut ne pas garantir cette propriété pour tous les objets. Par conséquent, l'absence de graines est fortement associée à une détérioration de la délimitation en raison de la concurrence réduite.

Après l'échantillonnage, SICLE calcule les superpixels à l'aide de la *Image Foresting Transform* (IFT) à partir de l'ensemble de graines. L'IFT est un cadre permettant de développer des opérateurs d'images basés sur la connectivité et peut être calculée en utilisant une généralisation de l'algorithme du chemin le plus court de Dijkstra pour prendre

en compte différentes fonctions de connectivité. Ainsi, dans l'IFT, chaque superpixel est un arbre dont les chemins vers sa graine sont optimaux. En d'autres termes, un arbre conquiert des pixels plus proches de lui que n'importe quel autre arbre. Selon la fonction de connectivité, l'algorithme peut ne pas garantir l'optimalité théorique des chemins, mais il produit tout de même une forêt couvrante enracinée dans les graines, dont l'applicabilité et les performances sont bien connues dans le domaine de la segmentation des superpixels.

En raison du suréchantillonnage, la troisième étape du pipeline consiste à sélectionner une partie des semences qui ne sont pas nécessaires à une segmentation précise tout en conservant celles qui sont cruciales pour atteindre cet objectif. Dans ce travail, nous appelons les premières graines *irrelevant* et les secondes *relevant*. Comme l'identification précise de la pertinence des graines est difficile, SICLE laisse cette information se manifester tout au long des itérations. En effet, en « perturbant » les graines par l'élimination partielle de celles qui ne sont pas pertinentes, les graines restantes sont laissées en concurrence pour les pixels « orphelins » et, par conséquent, chaque graine a une nouvelle chance de montrer sa pertinence. Dans cette stratégie, une graine pertinente est continuellement identifiée comme telle à chaque itération.

Toutefois, si l'on souhaite obtenir un objet spécifique, on peut constater que le critère de détermination de la pertinence d'une graine est miné par une telle subjectivité. Souvent, l'objectif est d'augmenter le nombre de graines à l'intérieur ou à proximité de l'objet souhaité. Pour cette raison, le critère peut identifier des faux positifs, qui peuvent être considérés comme des graines pertinentes pour des objets non pertinents. Par conséquent, la pertinence est soumise à une pénalisation chaque fois que cette information est fournie par la carte de saillance de l'objet.

Comme on peut le constater, le critère d'arrêt de SICLE est d'obtenir le nombre souhaité de superpixels. Dans la plupart des méthodes de pointe, un nombre strict d'itérations est nécessaire pour produire la segmentation et, dans certaines circonstances, l'obtention d'une telle quantité peut s'avérer inutile. Par exemple, lorsque le nombre de superpixels souhaité est très faible, la concurrence est relâchée et la conquête de pixels dissemblables est habituelle. Pour cela, l'approche courante consiste à réattribuer les graines au minimum du groupe afin de minimiser la dissimilarité intra-superpixel, liée aux caractéristiques ou à l'espace, par exemple. Toutefois, pour des quantités de superpixels plus élevées, la concurrence est extrême, ce qui conduit à des superpixels hautement spécialisés ou, en d'autres termes, à une réduction du regroupement des pixels dissemblables. Par conséquent, la réaffectation des graines peut ne pas apporter d'amélioration significative.

Par conséquent, l'objectif de SICLE est de considérer une courbe exponentielle pour déterminer le nombre de graines à éliminer à chaque itération. La motivation derrière une telle courbe spécifique est associée au comportement de compétition intrinsèque dans la segmentation en superpixels. À partir d'un ensemble beaucoup plus important de graines, dont la plupart ne sont pas pertinentes, l'élimination d'une grande partie à l'aide d'un critère précis n'aurait qu'un impact minimal sur les performances de la segmentation. Cependant, lorsque le nombre de graines diminue et se rapproche du nombre final souhaité de superpixels, l'élimination d'une grande partie, par exemple d'une partie d'objet, peut entraîner une déstabilisation majeure de la concurrence. Une telle suppression favoriserait de graves « fuites » et, par conséquent, la stratégie consiste à diminuer le nombre de graines à supprimer jusqu'à atteindre la quantité souhaitée de superpixels. Il est à noter que cette stratégie permet des calculs plus rapides pour un nombre élevé de superpixels tout en effectuant le nombre nécessaire d'itérations pour maintenir une délimitation précise pour des quantités plus faibles de superpixels. De plus, à partir d'une telle courbe, SICLE est

capable de générer une segmentation multi-échelle à la volée en produisant simplement les segmentations intermédiaires au fur et à mesure que le nombre de graines restantes est réduit et atteint chaque valeur demandée par l'utilisateur.

L'agnosticisme par rapport au domaine est réalisé dans SICLE, principalement par les étapes (ii) et (iii), au cours desquelles l'utilisateur peut fournir une fonction objective pour l'optimisation. On peut constater que, puisque le suréchantillonnage augmente les chances de sélectionner une graine dans chaque objet, il n'est pas nécessaire d'adapter l'étape (i) à de nouveaux domaines. En ce qui concerne la génération de superpixels, l'étape (ii), grâce à différentes fonctions de connectivité et de coût des bords, fournit la boîte à outils nécessaire pour produire des superpixels réguliers ou irréguliers sur la base d'un certain facteur de régularité chaque fois que cela est nécessaire. En d'autres termes, l'IFT gère principalement la morphologie des superpixels. Enfin, lors de la sélection des semences pertinentes, on peut constater que l'identification précise est directement associée au domaine et aux caractéristiques de l'objet, ce qui signifie que les biais sont cruciaux pour une estimation hautement spécialisée et efficace de la pertinence. Des objets différents avec des caractéristiques distinctes peuvent nécessiter des estimations de pertinence différentes, et des domaines différents peuvent exiger d'autres critères de pertinence.

Les résultats expérimentaux montrent que les variantes de SICLE surpassent plusieurs algorithmes de pointe en termes de vitesse et de précision pour des domaines distincts, tout en générant une série de segmentations en une seule exécution. Plus précisément, SICLE est capable de maintenir ses performances effectives quel que soit le domaine, et si les conditions sont favorables, comme la fourniture d'une carte de saillance précise, nos méthodes sont capables d'atteindre leur limite supérieure de performance en seulement deux itérations. En outre, les variantes compactes de SICLE peuvent exploiter la qualité de la carte de saillance donnée pour fournir une propriété inédite dans la segmentation des superpixels : des superpixels très compacts et très cohérents avec les frontières.

Cependant, aussi ressemblant que cela puisse paraître, SICLE ne garantit pas que cette dernière multi-échelle soit hiérarchique en raison de violations de la localité. En d'autres termes, bien que SICLE soit capable de produire une séquence strictement décroissante de segmentations, ce qui garantit le principe de causalité énoncé pour la segmentation hiérarchique, les frontières de l'échelle précédente ne sont pas assurées d'être dans l'échelle suivante. Par exemple, en changeant la concurrence pour l'itération suivante, les chemins optimaux précédemment « bloqués » par un superpixel non pertinent peuvent désormais conquérir plusieurs pixels, y compris ceux appartenant à un autre pixel de l'échelle précédente. Par conséquent, les frontières précédemment détectées peuvent maintenant être remodelées, violant le principe de localité et empêchant l'imbrication entre les régions.

Par conséquent, afin de fournir une mesure pour déterminer la ressemblance d'une segmentation multi-échelle à une hiérarchie, nous avons défini les concepts de noyau et de couverture, qui peuvent être brièvement définis comme les régions de l'échelle précédente qui sont, respectivement, imbriquées ou qui chevauchent la région considérée de l'échelle suivante. Sur la base de ces deux concepts, nous avons étudié et catalogué huit cas possibles lors de l'analyse des régions situées entre les segmentations subséquentes par paire : (i) scission complète ; (ii) déflation et scission ; (iii) déflation complète ; (iv) stabilité ; (v) instabilité ; (vi) inflation complète ; (vii) inflation et fusion ; (viii) fusion complète. Dans ces cas, une hiérarchie est strictement composée d'opérations de fusion complète ou de stabilité pour les séquences avec un nombre décroissant de régions, et elle est analogue à une scission complète ou à la stabilité d'un nombre croissant de régions.

À partir de ces cas, nous avons conçu trois mesures pour estimer le caractère hiérar-

chique d'une segmentation multi-échelle : (i) l'imbrication ; (ii) le rapport d'inflation ; et (iii) l'erreur de raffinement. La première mesure estime, sur la base du noyau de chaque région, le ratio des régions de l'échelle précédente imbriquées dans l'échelle suivante. D'autre part, la seconde mesure calcule le taux d'inflation des régions de l'échelle précédente pour construire les régions de l'échelle suivante. Pour cela, elle exclut la contribution des régions appartenant à un noyau de l'échelle suivante. Enfin, la troisième métrique s'inspire d'une métrique populaire de superpixels et mesure le taux de modification des régions de l'échelle précédente afin qu'elles soient toutes imbriquées dans l'échelle suivante. En termes simples, elle évalue l'effort minimal à fournir pour remodeler : soit en supprimant l'intersection entre les régions lorsqu'elle est suffisamment faible, soit en fusionnant totalement les régions lorsque l'intersection est suffisamment élevée.

Par définition, une hiérarchie doit présenter une imbrication maximale, un taux d'inflation minimal et une erreur de raffinement minimale. Cette propriété est également présente entre les segmentations suivantes de la hiérarchie, ce qui signifie que si toutes les deux partitions suivantes présentent un comportement d'imbrication, on peut conclure que toute la série présente un comportement d'imbrication et, enfin, que la multi-échelle est une hiérarchie. Parvenir à une telle conclusion grâce à nos contributions est une approche beaucoup plus efficiente que l'évaluation de toutes les comparaisons possibles par paire entre les échelles. En outre, comme on peut le noter, pour calculer ces mesures, nous supposons que l'échelle précédente est la « vérité de base » de l'échelle suivante.

Nos résultats nous permettent de vérifier que nos propriétés sont valables et, surtout, qu'il est possible de vérifier si une multi-échelle est une hiérarchie. De plus, lorsque ce n'est pas le cas, la combinaison des résultats de toutes les mesures offre une perspective large et claire de la nature et de l'étendue des violations hiérarchiques existant dans la segmentation multi-échelle qui l'empêchent d'être hiérarchique. Par exemple, les « quasi-hiérarchies » peuvent être identifiées dans deux cas : imbrication élevée, faible ratio d'inflation et erreur de raffinement, et imbrication faible, ratio d'inflation et erreur de raffinement. Alors que le premier cas offre une perspective plus claire sur le comportement d'imbrication, le second indique que, bien que l'imbrication ne soit pas garantie, l'inflation est minimale et la nécessité de corriger les frontières est, elle aussi, minimale.

List of Figures

1.1 Example of image superpixel segmentation. (a) Original image. (b) Segmentation in 500 superpixels by [1]. (c) Achievable segmentation of the objects from (b). 28
1.2 Examples of superpixel segmentation in different domains. (a-b) Results (in cyan) obtained by a classic [54] and a deep-learning algorithm [74], respectively, overlapping the ground-truth borders (in red and yellow). . . 29
1.3 Example of saliency-based superpixel segmentation. (a) Ground-truth. (b) Object saliency map [64]. (c) Superpixel segmentation [16, 17], in cyan, overlapping the ground-truth borders (in red). 30
1.4 Comparison of superpixel segmentation approaches against a compact and irregular instances of SICLE (*i.e.*, SICLE-COMP and SICLE-IRREG, respectively). The object boundaries and superpixel borders are depicted, respectively, in red and cyan, and yellow circles indicate regions in which our proposal achieved a better delineation than its counterparts. Finally, it was required a quantity of 100 superpixels. 34
1.5 Timeline of the contributions of this thesis, where those depicted in bold are directly associated to the thesis, whereas the remaining ones are related to it. 35

2.1 IFT execution in a toy example considering two seeds of distinct labels (depicted by colored diamonds). (a) Original graph with precomputed edge-costs. (b) Initialization of path-costs and mappings. (c-e) Non-subsequential IFT executions. (f) Resulting optimum-path-forest. 40

4.1 Pipeline for superpixel generation using SICLE (black, red, and blue arrows) and considering the saliency generated by [64]. DISF's pipeline is indicated by black arrows, whereas ODISF's by black and red. A quantity of 1000 seeds were sampled for generating a segmentation with 10 superpixels. 49
4.2 Impact of α in superpixel generation on an (a) image considering (b) a saliency map generated by [64]. The $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}^{\alpha}_{\text{root}}$ were considered, $N_0 = 3000$ and $N_f = 100$, and the values of α were (c-f) 0, 0.33, 0.66, and 1. . . 53
4.3 Multiscale segmentation generated by SICLE demanding (a-d) 500, 300, 200, and 100 superpixels. The initial seed quantity was $N_0 = 500$, whilst $N_f = 100$, and $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}_{\text{sum}}$ were considered. 56
4.4 Seed relevance criterion impact on a (a) set of 500 seeds using $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}^{\alpha}_{\text{sum}}$. A number of 106 were selected using (b-e) $\boldsymbol{V}_{\text{size}}$, $\boldsymbol{V}_{\text{minsc}}$, $\boldsymbol{V}_{\text{maxsc}}$, and $\boldsymbol{V}_{\text{spread}}$. 57

4.5 Seed relevance penalization impact on a (a) set of 500 seeds considering the saliency (b) generated by [64]. The criterion used was $\boldsymbol{V}_{\text{spread}}$ for selecting 106 seeds and the penalization performed was (c-g) none, $\boldsymbol{B}_{\text{obj}}$, $\boldsymbol{B}_{\text{border}}$, $\boldsymbol{B}_{\text{osb}}$, and $\boldsymbol{B}_{\text{bobs}}$. 60

5.1 Non-hierarchical multiscale generated by SICLE requiring (a-d) $N_f \in \{200, 100, 50, 25\}$ considering $N_0 = 1000$, $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$, and $\boldsymbol{V}_{\text{minsc}}$. 61
5.2 Diagram depicting all eight possible cases for building a region $Y_j \in Y$ from the regions of the partition X. 62
5.3 (a-c) Three distinct partitions of a 8×8 square universe Υ. 64

6.1 Quantitative results on the train set of each dataset for SICLE-COMP for varying ι. 72
6.2 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying $\boldsymbol{V}_*$. 73
6.3 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying Ω. 74
6.4 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying N_0. 75
6.5 Quantitative results on the test set of each dataset when no object information is provided. 76
6.6 Qualitative results with an image from Parasites requiring 100 superpixels. The object boundaries and the superpixel borders are depicted in magenta and cyan, respectively. The images were cropped for visualization purposes. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines. 78
6.7 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying α and using the U^2-Net map [64]. 79
6.8 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying saliency quality: bounding box (BB), U^2-Net map [64], and ground-truth (GT). 80
6.9 Segmentation results with SICLE-COMP considering the ground-truth as saliency map. (a) original image whose object contours are depicted in red; (b-d) Results with $N_f \in \{50, 30, 20\}$. For all segmentations, $\alpha = 2.0$. 80
6.10 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying $\boldsymbol{B}_*$ and using the U^2-Net map [64]. 81
6.11 Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying Ω and using the U^2-Net map [64]. 82
6.12 Quantitative results on the test set of each dataset using the U^2-Net map [64]. 83
6.13 Qualitative results with an image from ECSSD requiring 100 superpixels and using the U^2-Net estimator [64]. The object boundaries and the superpixel borders are depicted in red and cyan, respectively. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines. 84
6.14 Experimental results of the proposed hierarchiness measures on Liver and Parasites datasets (resp. left and right columns), whose average is depicted as a diamond. 85
6.15 Pseudocolored segmentations using SICLE-COMP and SICLE-IRREG, respectively, identified by C_n and I_n, for $n \in \{100, 250, 500\}$ regions. 87

List of Tables

5.1 (a-b) Non-normalized matrices based from the partitions in Figure 5.3. . . 67

6.1 Speed performance, in seconds, of the fastest methods on the BSDS500 dataset when no object information is provided. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f. 77

6.2 Speed performance, in seconds, of the fastest methods on ECSSD dataset using the U^2-Net map [64]. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f. 83

Contents

1 Introduction **27**
- 1.1 Context . . . 27
- 1.2 Motivation . . . 28
- 1.3 Major Goals and Research Questions . . . 31
- 1.4 Contributions . . . 32
- 1.5 Overview . . . 36

2 Fundamentals **37**
- 2.1 Multiscale and Hierarchies . . . 37
- 2.2 Image and Graphs . . . 38
- 2.3 Connectivity and Segmentation . . . 39
- 2.4 Image Foresting Transform . . . 39

3 Related Works **43**
- 3.1 Superpixel Properties . . . 43
- 3.2 Classical Methods . . . 45
- 3.3 Deep-learning-based Methods . . . 46

4 Building an Agnostic Superpixel Framework **48**
- 4.1 Seed Oversampling . . . 49
- 4.2 Superpixel Segmentation . . . 51
- 4.3 Seed Removal . . . 53
  - 4.3.1 Seed Relevance . . . 55
  - 4.3.2 Seed Penalization . . . 57
- 4.4 Algorithm . . . 59

5 Evaluating Hierarchiness **61**
- 5.1 Nucleus and Cover . . . 62
- 5.2 Cases . . . 62
- 5.3 Measures . . . 64

6 Experimental Results **68**
- 6.1 Datasets . . . 68
- 6.2 Methods . . . 69
- 6.3 Evaluation . . . 71
- 6.4 Non-object-based Analysis . . . 71
  - 6.4.1 Ablation Study . . . 72
  - 6.4.2 Quantitative and Qualitative Results . . . 75
- 6.5 Object-based Analysis . . . 77

6.5.1 Ablation Study 77
6.5.2 Quantitative and Qualitative Results 81
6.6 Hierarchiness Analysis 84

7 Conclusion and Future Work **88**

Bibliography **91**

Chapter 1

Introduction

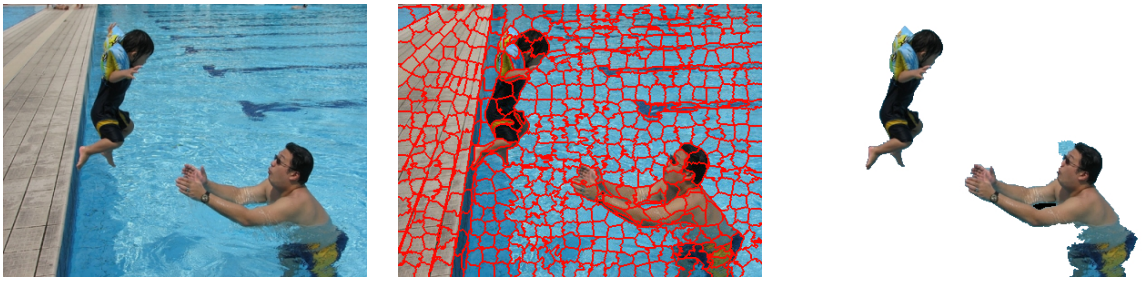
In this chapter, the first section (*i.e.*, Section 1.1) presents the context of this work, including the broad definitions of superpixels and hierarchies, while Section 1.2 details the challenges that state-of-the-art algorithms are unable to handle. From the latter, we present the main goals of this work alongside the research questions in Section 1.3. The contributions of this work, especially published papers, are described in Section 1.4. We end this chapter by presenting the overview of this work in Section 1.5.

1.1 Context

More formally, image segmentation can be described as the process of distinguishing objects in an image. In this work, every object crucial for solving a particular problem is simply termed as an object, and, in contrast, every object that is irrelevant to the same problem is termed as background. For instance, if a physician aims to discover a lung tumor (the object) in an image, other organs and tissues are seen as background. From this example, it is clear that the accurate distinguishment of objects is important and that errors may compromise subsequential tasks. In the previous example, a bad-quality segmentation can hamper the physician's diagnosis. A proper delineation of the objects favors several classification solutions, especially in medical contexts [33, 53, 93].

One of the most common approaches for segmenting objects, exemplified by Figure 1.1, is called *superpixel segmentation*, which can be informally associated with a puzzle-making solution. That is, the effective delineation of the object is achieved through the generation of numerous pieces that, when properly selected, build the desired objects. Mathematically speaking, *superpixels* is a collection of connected *picture elements* (*a.k.a.*, *pixels*) grouped by optimizing some objective function (*e.g.*, color similarity, spatial distance, *etc.*). Aside the previous property, a superpixel-wise representation presents a smaller magnitude than a pixel-wise one, being the major reason for their popular use in a variety of applications [28, 33, 53, 87, 90, 93].

However, for some applications, like *saliency detection* [47], a single segmentation is insufficient for estimating the object's "visual appeal" (*i.e.*, saliency). Thus, several segmentations of the same image, but with varying quantities of superpixels, are used to describe the object's characteristics with different resolutions. Coarser resolutions,



(a) (b) (c)

Figure 1.1 – Example of image superpixel segmentation. (a) Original image. (b) Segmentation in 500 superpixels by [1]. (c) Achievable segmentation of the objects from (b).

characterized by fewer regions, provide global information whilst reducing redundancy. In contrast, finer resolutions (*i.e.*, higher superpixel quantity) offer more detailed information about the object parts. Such sequence, termed as *multiscale* segmentation, in which each segmentation is now called a *scale* or *layer*, is widely used in literature for solving problems [44, 50, 56, 91]

A *hierarchical* segmentation is a multiscale one in which every region is contained in only one region of the subsequent scale. When this is not the case, we simply term as a *non-hierarchical* segmentation for distinction. Due to its nested behavior, several properties arise that can simplify, ensure, assist, and optimize several segmentation solutions. If one desires a hierarchical segmentation of an image that contains all possible scales, we say it is a *dense* hierarchy. Otherwise, if a subset suffices for solving a problem, then it is said to be a *sparse* hierarchy [37]. More formally, a sparse hierarchy can be defined as a proper subset of its corresponding dense hierarchy. Either way, hierarchical or not, both multiscale segmentation aims to effectively delineate the object in all scales generated.

1.2 Motivation

As exemplified, there are a variety of different applications that rely on superpixels in a specific *domain* that encapsulates: (i) data; (ii) hardware; and (iii) user interaction. Concerning data, we refer to availability and its characteristics (*e.g.*, natural or medical images), including those of the objects of interest. We also consider hardware limitations associated with the problem, such as the lack of GPUs. Finally, we consider the user's interaction towards a solution, like implementation and adaptation. In this work, we do not consider interactive methods.

In this work, we focus on two aspects: *bias* and *morphology*. We term as bias all prior knowledge of the data and its content, like the object's color, location, or texture, which can be useful for conceiving effective segmentation strategies. In contrast, morphology involves the set of morphological requirements (*e.g.*, shape, displacement) crucial for solving the task. Note that bias and morphology may differ significantly between domains and goals. Medical applications often present objects with well-known characteristics, while in natural domain, they may not. Also, segmentation benefits from irregular superpixels, whereas feature extraction usually requires compact regions. In the end, we argue that

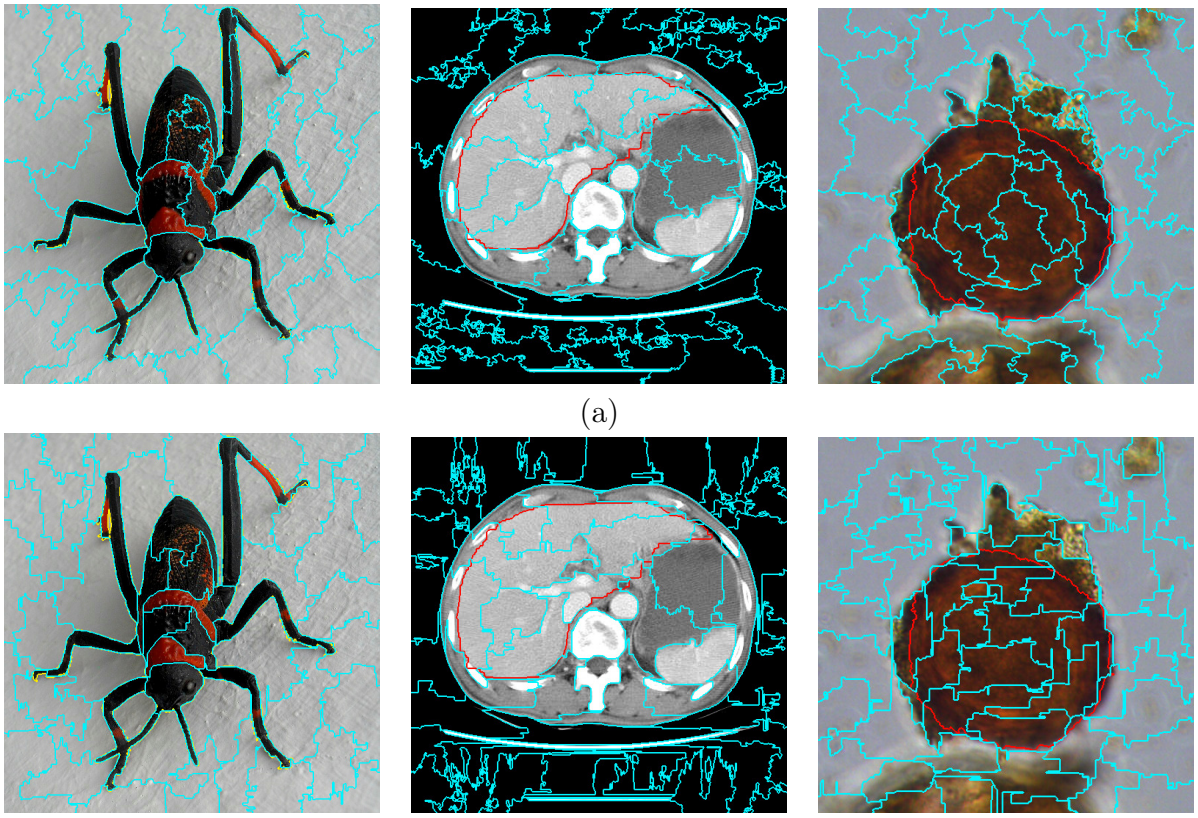


Figure 1.2 – Examples of superpixel segmentation in different domains. (a-b) Results (in cyan) obtained by a classic [54] and a deep-learning algorithm [74], respectively, overlapping the ground-truth borders (in red and yellow).

any method must provide the tools for adapting to any domain.

However, due to *(domain) internalization*, as we name it, adapting state-of-the-art superpixel algorithms is challenging. That is, such methods are highly optimized for specific domains since they were evaluated in several datasets with similar characteristics (*i.e.*, natural images). By assuming certain premises, object delineation may be compromised when delineating objects of different domains. Consequently, most approaches provide a set of parameters for either generating irregular or compact superpixels [1, 22, 54], appropriate for natural images, but may not be desirable in other domains. Even customizing displacement can be difficult to achieve for content-sensitive strategies [55, 81, 83] since they maximize superpixel resolution in regions with high feature variability. Thus, if the object is homogeneous, it will contain few superpixels, irrespective of the user's needs.

Some methods [7, 45, 62, 72, 74, 85, 88, 94] use deep-learning strategies and use fine-tuning for surrogating the drawbacks of domain internalization. Although it allows assimilating new biases, generating the desired morphology is difficult since it requires a well-tailored loss function to optimize. Also, even in possession of such functions, data availability (annotated or not) can be scarce in certain domains, especially medical. Furthermore, the need for specialized hardware (*i.e.*, GPUs) for achieving on-par speed performance with classical approaches is another major drawback of such algorithms. Finally, as illustrated by the mismatch between superpixel borders and object boundaries

in Figure 1.2, deep-learning-based strategies for superpixel segmentation require more research since their current reported performance in delineation is moderate compared to non-deep-learning ones.

Other works [16, 17, 84] consider prior object information, often represented by its *saliency* (*i.e.*, visual attention), for favoring the delineation of the desired object at the expense of poorly delineating those non-desired. Although the strict resultant morphology leads to accurate object delineation and high superpixel resolution within more salient regions, it impedes its usage for producing dispersed and compact superpixels. Other morphologies may also hamper its segmentation performance. Even though using such prior information may reduce the bias influence in delineation, these algorithms rely significantly on the information quality. Thus, when object information and domain biases do not converge (see Figure 1.3, their performance can severely deteriorate while trying to assimilate both.



(a) (b) (c)

Figure 1.3 – Example of saliency-based superpixel segmentation. (a) Ground-truth. (b) Object saliency map [64]. (c) Superpixel segmentation [16, 17], in cyan, overlapping the ground-truth borders (in red).

As one may note, the desired superpixel morphology may not respect the object characteristics. Requiring compact superpixels may impact the delineation of irregularly shaped objects, for instance. All the aforementioned methods try to solve such a dilemma during their superpixel generation through a single particular function, prioritizing either compactness or irregularity through a set of parameters. Therefore, satisfying more complex demands implies developing more complex functions, which such algorithms are often unable to achieve.

It is possible to argue in favor of exploiting several superpixel scales to generate an accurate segmentation result for the object. However, obtaining such a solution is combinatorial. Even when using hierarchical approaches, such as the *Superpixel Hierarchy* (SH) [80] algorithm, it is challenging to determine which combination of nodes (*i.e.,* cut) in the hierarchy provides such ideal output. Moreover, for hierarchical ones, errors generated during construction (*e.g.*, merging or splitting regions) are propagated, possibly aggravated, in the hierarchy. In contrast, non-hierarchical approaches are able to "correct" incorrect delineations by providing new and independent segmentations at each scale.

Although non-hierarchical superpixel methods achieve effective delineation results, they do not provide the optimizations and theoretical properties that hierarchical ones offer. Such impossibility results from violating two principles [43]. For instance, borders of coarser regions may not overlap those at finer scales, which violates the *locality principle*.

Also, in certain algorithms [36], different parameters may not build a sequence of segmentations in which the quantity of regions varies in a non-strict monotonical behavior, violating the *causality* principle.

If a non-hierarchical multiscale segmentation is presented in which violations exist, one may argue that it may be possible to extract information so that the objective is easily achievable. As an example, if there is a causality violation, reordering the series in a strictly monotonically decreasing order may be a solution. On the other hand, locality violation requires evaluation of its magnitude, such that it may be dismissed in non-crucial regions, especially when the object is accurately delineated. Thus, if one aims to hierarchically segment an object, it suffices to guarantee that its inner regions present a nested behavior between scales.

The previous procedure offers a straightforward verification of whether a multiscale segmentation is hierarchical or not. However, such binary measure does not provide insights into whether it is "almost" a hierarchy, or, simply put, whether the violations are minimal or severe. Note that, even using hierarchical structures, violations may occur when defining objects [61]. Finally, as one may note, if one considers the possibility of adding, removing, or reordering scales, such a problem becomes combinatorial. Even when considering two subsequent partitions in a multiscale segmentation, in which one could measure every two pairs in the sequence, the challenge stays unchanged. To the best of our knowledge, the existence of such a metric is unknown, and most of the works related to such task either require an object ground-truth for segmentation quality assessment [82, 73], or weight contours differently based on a ground-truth [63].

1.3 Major Goals and Research Questions

In this work, our main goal is to study and develop a superpixel segmentation framework that provides effective and efficient variants, irrespective of the application's domain (*i.e.*, domain-agnostic). Moreover, we investigate the resemblance of the multiscale segmentations to a hierarchy through a hierarchiness measure. Simply put, the goal of this thesis is to answer the following questions:

Question 1: *Is it possible to build a superpixel segmentation method that assimilates any bias?*

As one may note, the question of neglecting any bias is, by itself, ill-posed. Development and optimization are both tasks that benefit from pre-known information about the goal. Therefore, in this thesis, we rephrased it to study whether it is possible to assimilate any bias. Differently from the former, the new formulation allows a more feasible task since the construction of a suitable variant is reduced to the problem of providing the necessary information for an accurate result.

Question 2: *Is it possible to build a superpixel segmentation method that promotes any morphology?*

Similarly, the challenge of providing a segmentation that carries all possible morphologies is, again, ill-posed since most morphologies are conflicting. For instance, requiring regularly shaped superpixels while demanding optimal delineation is a dilemma posed by most researchers. Thus, in this work, the promotion of any morphology is seen as the maximization of different objective functions (*e.g.*, shape, and location) if one can provide such mathematical function.

Question 3: *Is it possible to build a superpixel segmentation method that achieve both latter with efficacy and efficiency for any object?*

We argue that the goal of delineation, as it is posed in several works, is an unfeasible task due to its subjectiveness. As an example, for an optic character recognition method, the accurate delineation of a post sign is one that segments the characters. In contrast, for a saliency detection algorithm, such optimal result involves segmenting the whole sign while disregarding the information within. Therefore, we claim that, like biases and morphologies, agnosticism is achieved by allowing proper but generic information about the object of interest. For that, one may argue for considering the object's features (*e.g.*, color) or shape, but even objects with the same classification (*e.g.*, dogs) present different characteristics. On the other hand, any object is located somewhere in the image, and from it, one may extract the object's features and shape information. Note that the converse may not be easy or feasible. Therefore, by allowing prior knowledge of the object's probable location, one may favor the accurate delineation of the object of interest while disregarding those that are considered irrelevant for solving the user's particular problem.

Question 4: *Is it possible to conceive a measure in which estimates the resemblance of a multiscale segmentation to a hierarchical one?*

As previously mentioned, an estimation that considers all possibilities within a multiscale segmentation to minimize its divergence to a hierarchy is a combinatorial problem. However, one may establish a strict order, such as decreasing the number of regions, and evaluate pairwise subsequent scales for calculating such measure since such sequence is closer to an algorithm's output from varying parameters rather than an optimal but independent region selection within scales. Note that since nesting behavior is maintained in such ordering, this approach would be able to detect whether a multiscale is hierarchical.

1.4 Contributions

As stated by several works in superpixel segmentation [71, 78, 77], including one survey of our own [10], many methods [1, 8, 17, 25, 52, 55, 75] use a three-step pipeline for segmenting images: (i) seed sampling; (ii) superpixel generation; (iii) seed recomputation. In (i), a set of initial and representative pixels (*i.e.*, *seeds*) are selected using some strategy, often a grid-like distribution. Then, in (ii), superpixels are generated with respect to

their own particular seed by optimizing some objective function. Then, step (iii) recomputes the seed set by, for instance, minimizing the intra-superpixel color dissimilarity, a popular approach of many methods [1, 52, 75]. Finally, steps (ii) and (iii) are repeated for a number of iterations until convergence is met, often being defined as reaching the maximum quantity of iterations or surpassing an error tolerance.

However, this pipeline has several drawbacks [12, 13, 14, 18, 21], being the major motivation for proposing a novel three-step pipeline: (i) seed oversampling; (ii) superpixel generation; and (iii) seed removal. Although alike, this new approach has two significant differences compared to the classical one. In (i), the number of seeds sampled is substantially higher than the desired final number of superpixels, those generated in step (ii) using the *Image Foresting Transform* (IFT) [35], while (iii) removes a portion of seeds classified as insignificant in the object delineation (*i.e.*, irrelevant). Clearly, the method halts when the number of seeds equals the number of superpixels. Although variants of this pipeline may produce a multiscale segmentation on the fly, it may not guarantee hierarchical properties stated in [42], such methods are able to correct eventual errors in each new and independent segmentation.

The *Dynamic and Iterative Spanning Forest* (DISF) [18] was the first method to recur to such a pipeline. By generating superpixels using a dynamic arc-cost estimation and ranking seed relevance based on the superpixel's size and contrast, DISF surpassed all state-of-the-art methods evaluated in different datasets and considering popular superpixel delineation metrics. Moreover, from such performance, it has been used in several works, such as interactive segmentation [11], and video supervoxel segmentation [46, 76].

Like most state-of-the-art algorithms, DISF is unaware of the user's desire to segment a specific object. For instance, if a mapping indicating the object's probable location (*i.e.*, object saliency map) is provided, DISF cannot modify nor improve its segmentation. Thus, similarly to [16, 17, 84], we generalized DISF to include such information, resulting in the *Object-based DISF* (ODISF) [14]. Different from its predecessor, ODISF provides a more accurate and relevant seed selection in step (iii) by favoring seeds within and near the object of interest. As a result, ODISF achieves improved delineation while being the fastest saliency-based method evaluated in the experiments.

Still, both DISF and ODISF do not allow intuitive user control over the number of iterations for segmentation, and they are unable to produce compact superpixels like their state-of-the-art counterparts. Moreover, ODISF cannot improve its delineation result if an accurate object estimation is provided since such information is limited to step (iii). Thus, we generalized both into a new superpixel segmentation framework named *Superpixels through Iterative CLEarcutting* (SICLE), resultant from the works in [20, 12, 13, 21]. SICLE overcomes all the aforementioned drawbacks and surpasses state-of-the-art methods in efficiency and efficacy in clearly distinct domains (*i.e.,* natural and medical image datasets), as shown in Figure 1.4. For instance, if certain conditions are met, only two iterations are needed for maximum delineation performance with maximum compactness, a property not achieved by the methods in the literature. From such performance, SICLE has shown to be an appropriate choice for resegmentation [49] when compared to other superpixel approaches.

One may see certain resemblances between the strategies of SICLE and Stochastic

watersheds [4], especially due to oversampling. However, as mentioned, SICLE recomputes and refines its seed set by removing seeds while maintaining the remaining ones in their previous locations. For Stochastic Watersheds, new random seeds are generated at each iteration to produce a probability density function, which SICLE does not compute. In fact, in SICLE, by providing an accurate seed relevance measure, the relevant contours are to be found in the first iteration (and sustained throughout the algorithm's execution). Finally, while Stochastic Watersheds performs a strict number of executions, SICLE may achieve effective delineation with fewer iterations than the maximum imposed by the user.

However, it is uncertain whether a multiscale segmentation, such as the one generated on the fly by SICLE and predecessors, resembles a hierarchy (locally or globally) so that one may exploit several optimizations and properties. Note that, for evaluating the nested behavior in between segmentations, one does not require a ground truth (often unavailable) nor to assess the segmentation quality. Thus, we explore [15] several cases when analyzing two consecutive segmentations. From those, we provide three measures for evaluating the hierarchiness between two subsequent partitions: (i) nestedness; (ii) refinement error; and (iii) inflation ratio. Through our proposals in an in-sequence pairwise comparison, it is possible to verify whether a multiscale segmentation is a hierarchy and, if not, to analyze the nature and extent of the hierarchical errors that prevent it from becoming hierarchical.

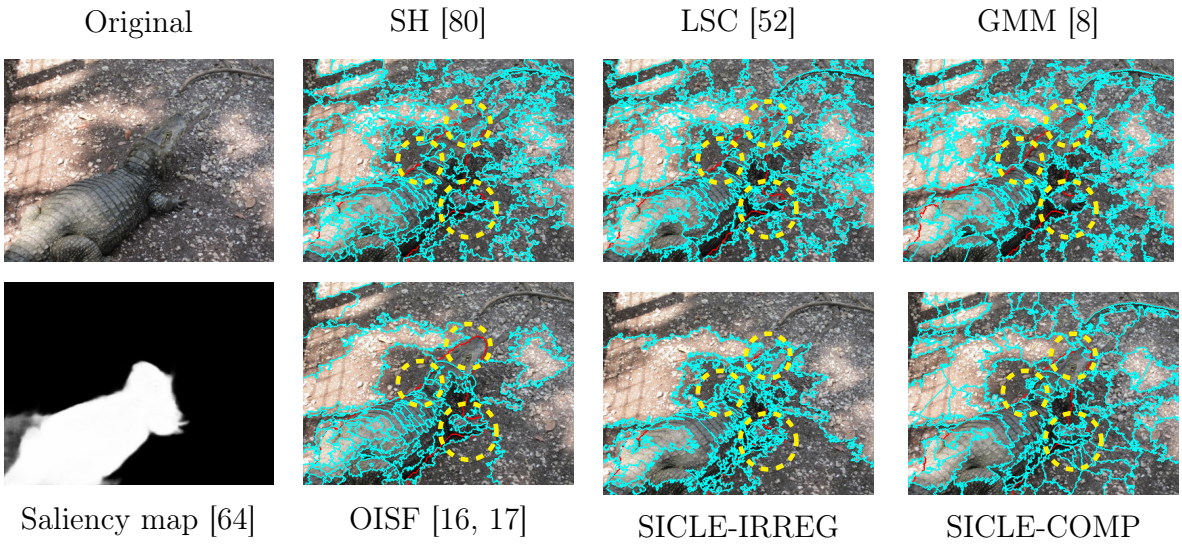


Figure 1.4 – Comparison of superpixel segmentation approaches against a compact and irregular instances of SICLE (*i.e.*, SICLE-COMP and SICLE-IRREG, respectively). The object boundaries and superpixel borders are depicted, respectively, in red and cyan, and yellow circles indicate regions in which our proposal achieved a better delineation than its counterparts. Finally, it was required a quantity of 100 superpixels.

In summary, this thesis has led to the following published, accepted, submitted and unpublished contributions listed below. Those that are directly associated with the thesis are depicted in bold, whereas the remaining ones are related to the thesis, such as applications.

[18] **F.C. Belém, S.J.F. Guimarães and A.X. Falcão**, "Superpixel Segmentation

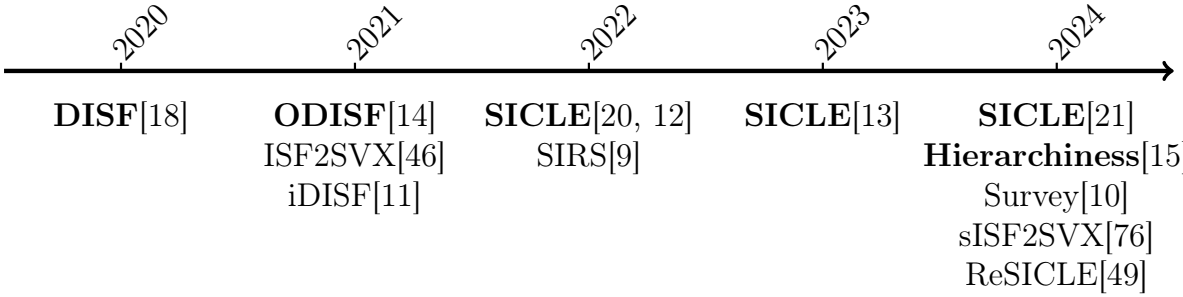


Figure 1.5 – Timeline of the contributions of this thesis, where those depicted in bold are directly associated to the thesis, whereas the remaining ones are related to it.

Using Dynamic and Iterative Spanning Forest". IEEE Signal Processing Letters, 2020.

[46] C. Jerônimo, F.C. Belém, S.A. Carneiro, Z.K.G. Patrocínio Jr., L. Najman, A.X. Falcão, S.J.F. Guimarães, "Graph-Based Supervoxel Computation from Iterative Spanning Forest". Discrete Geometry and Mathematical Morphology (DGMM), 2021.

[14] **F.C. Belém, B.Perret, J.Cousty, S.J.F. Guimarães and A.X. Falcão**, "Towards a Simple and Efficient Object-based Superpixel Delineation Framework". 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2021.

[11] I.B. Barcelos, F.C. Belém, P.A.V. Miranda, A.X. Falcão, Z.K.G. Patrocínio Jr., and S.J.F. Guimarães, "Towards Interactive Image Segmentation by Dynamic and Iterative Spanning Forest", Discrete Geometry and Mathematical Morphology (DGMM), 2021.

[9] I.B. Barcelos, F.C. Belém, L.M. João, A.X. Falcão and S.J.F. Guimarães, "Improving color homogeneity measure in superpixel segmentation assessment". 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2022.

[20] **F.C. Belém, B.Perret, J.Cousty, S.J.F. Guimarães and A.X. Falcão**, "Efficient Multiscale Object-based Superpixel Segmentation Framework". ArXiv Preprint, 2022.

[12] **F.C. Belém, I.B. Barcelos, L. M. João, B. Perret, J. Cousty, S.J.F. Guimarães and A.X. Falcão**, "Fast and Effective Superpixel Segmentation Using Accurate Saliency Estimation". Discrete Geometry and Mathematical Morphology (DGMM), 2022.

[13] **F.C. Belém, I.B. Barcelos, L. M. João, B. Perret, J. Cousty, S.J.F. Guimarães and A.X. Falcão**, "Novel Arc-Cost Functions and Seed Relevance Estimations for Compact and Accurate Superpixels". Journal of Mathematical Imaging and Vision, 2023.

[10] I.B. Barcelos, F.C. Belém, L.M. João, Z.K.G. Patrocínio Jr., A.X. Falcão and S.J.F. Guimarães, "A Comprehensive Review and New Taxonomy on Superpixel Segmentation". ACM Computer Surveys, 2024;

[76] D. Vieira, I.B. Barcelos, F.C. Belém, Z.K.G. Patrocínio Jr., A.X. Falcão and S.J.F. Guimarães, "Streaming Graph-based Supervoxel Computation Based on Dynamic

Iterative Spanning Forest". Progress In Pattern Recognition, Image Analysis, Computer Vision and Applications (CIARP), 2024.

[21] **F.C. Belém, B.Perret, J.Cousty, S.J.F. Guimarães and A.X. Falcão**, "Efficient Multiscale Object-based Superpixel Segmentation Framework". Submitted to the Journal of the Brazilian Computer Society, 2024.

[15] **F.C. Belém, F. Faria, A.X. Falcão, Z.K.G. Patrocínio Jr., S.J.F. Guimarães, B. Perret, and J. Cousty**, "Measuring Hierarchiness of Image Segmentations". To appear in the Brazilian Conference on Graphics, Patterns and Images, 2024.

[49] L. Lacerda, F.C. Belém, A.X. Falcão, Z.K.G. Patrocínio Jr., and S.J.F. Guimarães, "Seed-based Superpixel Re-Segmentation for Improving Object Delineation". To appear in the Iberoamerican Conference on Pattern Recognition, 2024.

1.5 Overview

This thesis is organized as follows. In Chapter 2, we present the mathematical definitions for a complete understanding of our work, from the concept of hierarchies and images to image graphs and segmentation. Then, in Chapter 3, we present recent and state-of-the-art methods in superpixel segmentation and broadly classify them by their strategy, whether they are classical or deep-learning-based. We describe the incremental work for developing SICLE in Chapter 4 and, in Chapter 5, we present the measures for estimating the hierarchiness between partitions in a multiscale segmentation. Experimental results for the aforementioned contributions are detailed and performed in Chapter 6, ending this thesis in Chapter 7 by drawing conclusions and future work.

Chapter 2

Fundamentals

Most works model the superpixel generation as a graph-based (directed or not) problem in which each superpixel is considered a component. That is, from a given input image, a graph is built in which the vertices are the image's pixels, and the edges indicate the immediate adjacents of each pixel. A superpixel segmentation algorithm then agglomerates connected pixels by their similarity (based on an objective function) into disjoint groups and, finally, performs a one-to-one correspondence between each vertice's group and its respective pixel. In this work, due to the *Image Foresting Transform* (IFT) [35], our core algorithm, each superpixel is built through competition, resulting in a tree rooted in a particular seed.

We start off in Section 2.1, introducing basic concepts for defining multiscale and hierarchical partitions of sets based on the works of [65, 67]. Section 2.2 presents the definitions of images and graphs, and, subsequently, in Section 2.3, we present the notion of connectivity, which is crucial for defining superpixels and segmentation. The IFT is detailed, and its algorithm is presented in the last section of this chapter (*i.e.*, Section 2.4).

2.1 Multiscale and Hierarchies

A *set* Υ can be defined as a collection of distinct *elements*, and it is referred to as *universe* if it contains all existing elements. The *size* or *cardinality* of Υ is denoted by $|\Upsilon|$ and is informally defined as the number of elements within it. Also, a *subset* $S \subseteq \Upsilon$ is a set in which, for all $x \in S$, $x \in \Upsilon$, and it is *proper* if $S \subseteq \Upsilon$ but $S \neq \Upsilon$.

Let $\mathbb{P}$ denote the power set, and $\Pi : \Upsilon \mapsto \mathbb{P}(\Upsilon)$ be a function which defines all *partitions* of Υ such that, $\forall\ P \in \Pi(\Upsilon)$, $\cup P = \Upsilon$, $P_i \cap P_j = \emptyset$ for any distinct $P_i, P_j \in P$, and $\emptyset \notin P$. The terms *block* or *region* refer to elements of a partition. The *identity partition* P is one in which every element in Υ is a block in P. Conversely, if P is composed of a single block, it is termed as the *universal partition*. Finally, if a subset $E \subseteq \Upsilon$ has a partition $Q \in \Pi(E)$ but Q is not necessarily a partition of Υ, then $Q \in \Pi^*(\Upsilon)$ is a *partial partition* of Υ, where $\Pi(\Upsilon)$ denotes the set of all partial partitions of Υ.

We may order the set $\Pi(\Upsilon)$ by size. Thus, given two partitions $P, Q \in \Pi(\Upsilon)$, we say that P is *bigger* than Q (denoted by $P \trianglerighteq Q$) if, and only if, $|P| \geq |Q|$. Then, in such case, Q is said to be *smaller* than P (denoted by $Q \trianglelefteq P$). From those, we can define a

multiscale partition $\langle \mathcal{M}, \trianglerighteq \rangle$ as a sequence of partitions $\mathcal{M} \subseteq \Pi(\Upsilon)$ (often called *layers* or *scales*) ordered in a non-increasing order of size. If $P \trianglerighteq Q$ in $\mathcal{M}$, then we say that P is a *downscale* of Q and *precedes* it, while Q is an *upscale* of P and *succeeds* it.

Similarly, it is possible to partially order the set $\Pi(\Upsilon)$ by *refinement* such that, for instance, P is considered *finer* than Q (denoted by $P \preceq Q$) if and only if, for every region $p \in P$, exists a unique $q \in Q$ in which $p \subseteq q$. Thus, Q is said to be *coarser* than P, and we write such information by $Q \succeq P$. One can see that $P \preceq Q$ implies that $P \trianglerighteq Q$, but the converse is not true. A *hierarchical* partition $\mathcal{H}$ is, then, a multiscale partition $\langle \mathcal{M}, \preceq \rangle$ ordered in non-increasing order of refinement, and can be classified either as *dense* or *sparse* based on the number of scales $|\mathcal{M}|$ [37].

2.2 Image and Graphs

An *image* I can be seen as a pair $(P, \boldsymbol{F})$ in which P denotes the set of *picture elements* (*pixels*) $p \in P \subset \mathbb{Z}^2$, whose *features*, such as the CIELAB colorspace, is a m-dimensional vector $\boldsymbol{F}(p) \in \mathbb{R}^m$, for $m \in \mathbb{N}_+$, mapped by a *feature function* $\boldsymbol{F}$. Thus, whenever $m = 1$ or $m > 1$, we say that I is a *grayscale* or *colored* image, respectively. As an example, an *object saliency map* is a grayscale image $O = (P, \boldsymbol{O})$ where $\boldsymbol{O}(p) \in [0, 1]$ that maps each pixel $p \in P$ to its proportional object likelihood (*i.e.,* its *saliency*), represented by its intensity. Therefore, higher intensities indicate higher object probabilities, whereas darker ones indicate higher background probability. In this work, the universe Υ is the set of pixels P. As one may note, we may interpret the pixel $p = \langle p_x, p_y \rangle$ as a bi-dimensional vector in which p_x and p_y define its *spatial coordinates*.

First, let $x = \langle x_1, x_2, \ldots, x_k \rangle \in \mathbb{R}^k$, for $k \in \mathbb{R}_+^*$, be a k-dimensional vector. We may compute its *p-norm* as presented in Equation 2.1

$$\|x\|_p = \left(\sum_{j=1}^{k} |x_j|^p \right)^{\frac{1}{p}} \tag{2.1}$$

Then, we may compute the *spatial* and *feature dissimilarity*, respectively, between any two pixels $p, q \in P$ using the *Manhattan distance* $\|p-q\|_1$ and $\|\boldsymbol{F}(p)-\boldsymbol{F}(q)\|_1$, or using the popular *Euclidean distance* $\|p-q\|_2$ and $\|\boldsymbol{F}(p)-\boldsymbol{F}(q)\|_2$. As an example, from its spatial dissimilarity, we can define the *neighborhood* $\boldsymbol{A}(p) \in \mathbb{P}(P)$ of p such that every $q \in \boldsymbol{A}(p)$ is termed as *adjacent (to p)*, and both p and q are said to be *adjacents*. Although there exist several strategies for defining $\boldsymbol{A}(p)$, most works deal with the 4- or 8-neighborhood that are formally defined by the generalization below (Equation 2.2):

$$\boldsymbol{A}^r(p) = \{q \in P : \|p - q\|_2 \leq r \in \mathbb{R}_+^*\} \tag{2.2}$$

considering the *radius* $r = 1$ and $r = \sqrt{2}$, respectively.

Given an image I, it is possible to build a *graph* $G = (V, E)$ in which $V \subseteq P$ and $E \subset V^2$ denote the *vertex* and the *edge* set, respectively. We can define such edges through $E = \{\{p, q\} : \exists\ p, q \in V$ where $p \in \boldsymbol{A}(q)$ or $q \in \boldsymbol{A}(p)\}$. Likewise, if $\{p, q\} \in E$, then p and q are adjacents. As one may note, the previous definition of E excludes duplicate

instances of $\{p, q\} \in E$, that we often refer as *parallel edges* and, in this work, we also exclude the possibility of existing $\{p, p\} \in E$ (*i.e.*, *auto-loops*). Therefore, G is a *simple* graph.

2.3 Connectivity and Segmentation

A *walk* ρ in a simple graph can be represented as a finite sequence of $k \in \mathbb{N}_+$ vertices $\langle v_1, v_2, \ldots, v_k \rangle$ in which $\langle v_i, v_{i+1} \rangle \in E$ for $1 \leq i < k$. In such case, v_1 and v_k are, respectively, ρ's *origin* and *terminus*, and, for simplicity, we may exhibit both by $\rho_{v_1 \rightsquigarrow v_k}$. We may also omit the origin and write ρ_{v_k} whenever v_1 is irrelevant to the context. For two subsequent vertices v_i, v_{i+1} in the sequence, we term v_i as the *predecessor* of v_{i+1}, and v_{i+1} as the *successor* of v_i. A walk ρ is referred to as *trivial* for $k = 1$, and as *nontrivial*, otherwise. Furthermore, a nontrivial walk $\rho_{v \rightsquigarrow u}$ between two vertices $u, v \in V$ can be either classified as *open*, if $v \neq u$, or *closed*, otherwise. From that, we can define a *path* as an open walk whose vertices do not repeat throughout the sequence. Similarly, a *cycle* is a closed walk $\rho_{v \rightsquigarrow v}$ in which all vertices, except v, do not repeat.

Given $G = (V, E)$, a *subgraph* G' is a pair (V', E') where $V' \subseteq V$ and $E' \subseteq E$. It is *spanning* if $V' = V$, and it is *induced* if, $\forall\ u, v \in V'$ in which $\{u, v\} \in E$, we have that $\{u, v\} \in E'$. Let $\Psi(G)$ be the set of all possible walks in G such that it defines $u, v \in V$ as *connected* by the existence of some walk $\rho_{v \rightsquigarrow u} \in \Psi(G)$. From that, we may say that a subset $C \subseteq V$ is *connected* whenever the induced subgraph by C is also connected. A *component* is a subgraph induced by a maximal set $C \subseteq V$ of connected vertices, which we will also refer to as *component*, for simplicity. If such a component does not contain closed walks, then it is called a *tree*. If G contains a single component, then it is *connected*; otherwise, it is *disconnected*. Finally, if every component of G is a tree, then G is a *forest*.

As one may note, the task of *(hard) segmentation* highly correlates with the aforementioned concepts. A *segmentation* $\{C_1, \ldots, C_k\} \in \Pi(V)$ is a set of $k \in \mathbb{N}_+$ non-empty disjoint components which spans and partitions V. If a set $S \subset V$ of initial vertices (*i.e.*, *seeds*) is provided, a *seed-based* segmentation is such one in which every seed $s \in S$ defines its unique non-empty disjoint component C_s. Often, in such cases, we say that s is the *root* of C_s and C_s is *rooted in* s. Therefore, a *superpixel segmentation* is a segmentation (seed-based or not) of a graph, and every component is a *superpixel*. Consequently, a *multiscale* and *hierarchical* superpixel segmentation is a sequence $\mathcal{M}$ of superpixel segmentations ordered by size and refinement, respectively.

2.4 Image Foresting Transform

The *Image Foresting Transform* (IFT) [35] is a framework for developing connectivity-based operators through a generalization of the Dijkstra's algorithm and, in this work, we are considering its seed-restricted variant. In simple terms, the IFT generates a forest from a graph $G = (V, E)$ whose trees (*i.e.*, superpixels) are built by assigning an optimum-cost path from a seed $s \in S$ to all the remaining vertices $v \in V \setminus S$.

In the IFT, optimum-paths are formed through *path concatenation* from the seeds.

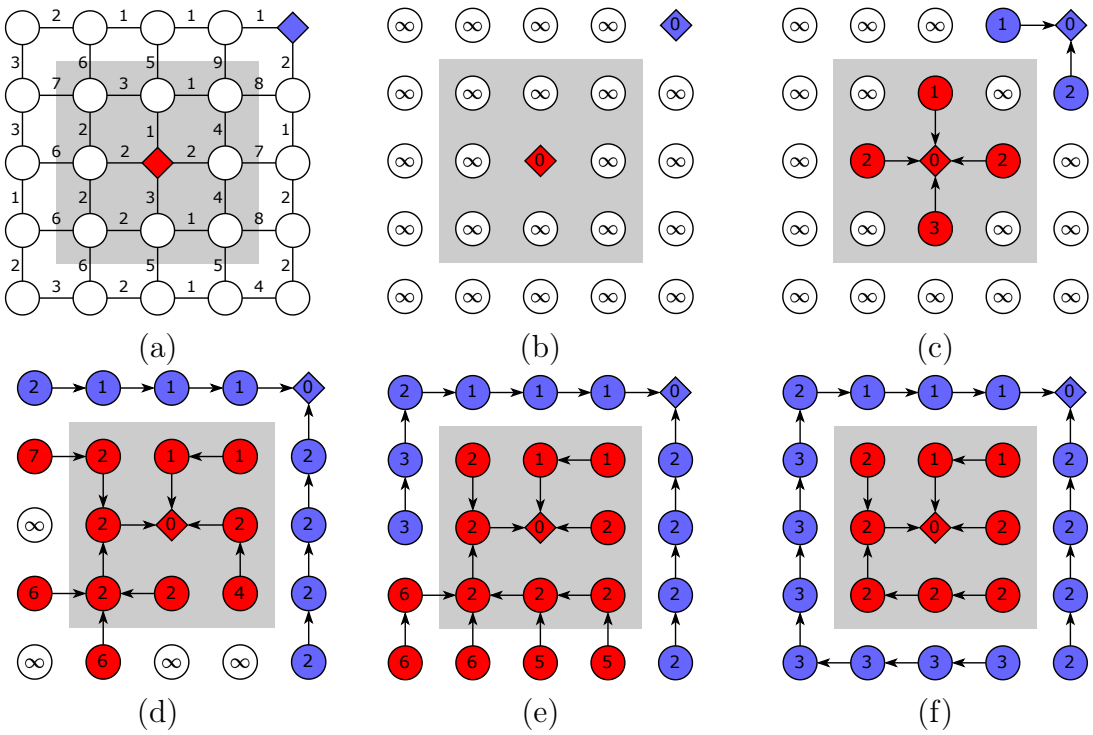


Figure 2.1 – IFT execution in a toy example considering two seeds of distinct labels (depicted by colored diamonds). (a) Original graph with precomputed edge-costs. (b) Initialization of path-costs and mappings. (c-e) Non-subsequential IFT executions. (f) Resulting optimum-path-forest.

That is, for $s \in S$ and $p, q \in V \setminus S$, the path resultant from concatenating $\rho_{s \rightsquigarrow p} \in \Psi(G)$ with $\{p, q\} \in E$ can be defined by $\rho_{s \rightsquigarrow q} = \rho_{s \rightsquigarrow p} \odot \{p, q\} = \langle s, \ldots, p, q \rangle$. Then, by recurring to a *connectivity function* $\boldsymbol{f}_*$, we can assign a *path-cost* $\boldsymbol{f}_*(\rho_q) \in \mathbb{R}_*$ to every path ρ_q. In this work, independently of $\boldsymbol{f}_*$, the cost of all trivial paths are defined by Equation 2.3:

$$\boldsymbol{f}_*(\rho_s = \langle s \rangle) = \begin{cases} 0, & \text{if } s \in S \\ \infty, & \text{otherwise} \end{cases} \tag{2.3}$$

Thus, all paths start from a seed $s \in S$. For non-trivial paths, one may recur to several path-cost functions already proposed in literature [17, 58, 75]. As an example, two popular functions, $\boldsymbol{f}_{\max}$ and $\boldsymbol{f}_{\text{sum}}$ compute the path-costs as defined in Equation 2.4:

$$\begin{aligned} \boldsymbol{f}_{\max}(\rho_v \odot \{v, u\}) &= \max\{\boldsymbol{f}_{\max}(\rho_v), \boldsymbol{w}_*(v, u)\} \\ \boldsymbol{f}_{\text{sum}}(\rho_v \odot \{v, u\}) &= \boldsymbol{f}_{\text{sum}}(\rho_v) + \boldsymbol{w}_*(v, u) \end{aligned} \tag{2.4}$$

in which $\boldsymbol{w}_*(v, u) \in \mathbb{R}_+^*$ denotes the *edge-cost function* that computes the cost of concatenating the edge $\{u, v\} \in E$ with a path $\rho_v \in \Psi(G)$. A path ρ_q is said to be *optimum* if, for any other $\varrho_q \in \Psi(G)$, $\boldsymbol{f}_*(\rho_q) \leq \boldsymbol{f}_*(\varrho_q)$. Although such property is guaranteed when using $\boldsymbol{f}_{\max}$, other connectivity functions, such as $\boldsymbol{f}_{\text{sum}}$, may not satisfy a set of conditions presented in [29] (*i.e.*, it is not *smooth*). Yet, works have demonstrated that the result still presents important properties for segmentation, especially for superpixel segmentation [58].

During execution, and with the use of a priority queue Q, the IFT minimizes a *cost map* $\boldsymbol{C}(v) = \min_{\rho_v \in \Psi(G)}\{\boldsymbol{f}_*(\rho_v)\}$, assigning then an optimum path ρ_v to every $v \in V$. When assigned, since $\rho_v = \rho_{s \rightsquigarrow u} \odot \{u, v\}$ is built through concatenation for some $s \in S$ and $u \in V$, it is known that u is the predecessor of v, and we may access such information through a *predecessor map* $\boldsymbol{P}$ such that $\boldsymbol{P}(v) = u$. For a seed s, we assign $\boldsymbol{P}(s)$ a distinctive marker ▲, since they possess no predecessor.

From $\boldsymbol{P}$, we can recursively map each vertex v to its root s in the optimum-path $\rho_{s \rightsquigarrow v}$, resulting in a *root map* $\boldsymbol{R}(v) = \boldsymbol{R}(s) = s$. In this work, each seed s has a unique label $\boldsymbol{L}(s) \in \mathbb{N}_+$ defined through a *label map* $\boldsymbol{L}$ such that, from $\boldsymbol{R}$ and for every $v \in V$, we have that $\boldsymbol{L}(v) = \boldsymbol{L}(\boldsymbol{R}(v)) = \boldsymbol{L}(s)$. Likewise, we can map v to its tree (rooted in $\boldsymbol{R}(v) = s$) $T_s = \{v \in V : \boldsymbol{R}(v) = s\}$. It is clear that $T_s \subseteq V$, in this context, is also a component. If $\boldsymbol{f}_*$ is smooth, then we may say that T_s is an *optimum-path tree* and, consequently, that the IFT generates an *optimum-path forest*. Otherwise, as one may note, the result consists of a spanning forest of the input graph G.

As one may note, a tree is also a superpixel in the IFT. For generating irregularly-shaped superpixels with accurate object delineation, one may recur to $\boldsymbol{f}_{\max}$ with the edge-cost function $\boldsymbol{w}_{\text{root}}$. Conversely, for achieving a balance between compacity and delineation performance, a proper configuration would be $\boldsymbol{f}_{\text{sum}}$ with $\boldsymbol{w}_{\text{sum}}$. Both edge-cost functions are defined below (Equation 2.5):

$$\begin{aligned} \boldsymbol{w}_{\text{root}}(v, u) &= \|\boldsymbol{F}(u) - \boldsymbol{F}(\boldsymbol{R}(v))\|_2 \\ \boldsymbol{w}_{\text{sum}}(v, u) &= (\iota\|\boldsymbol{F}(u) - \boldsymbol{F}(\boldsymbol{R}(v))\|_2)^{\beta} + \|v - u\|_2 \end{aligned} \tag{2.5}$$

in which $\iota \in \mathbb{R}_+^*$ and $\beta \in \mathbb{R}_+^*$ control the superpixel's irregular shape and the superpixel's adherence to boundaries, respectively.

Given that IFT recurs to the Dijkstra's algorithm, its complexity is $\mathrm{O}(|V| \log |V|)$, but several works manage to circumvent such computational cost by forcing integer costs [35], by allowing differential computations [30, 34, 69] or even by using specialized hardware, such as GPUs [40]. Nevertheless, the general IFT algorithm can be summarized below in Algorithm 1 and exemplified in Figure 2.1, using $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$ functions:

Algorithm 1 General IFT algorithm

Input: Graph $G = (V, E)$, seed set $S \subset V$, connectivity function $\boldsymbol{f}_*$
Output: Label map $\boldsymbol{L}$, root map $\boldsymbol{R}$, predecessor map $\boldsymbol{P}$, cost map $\boldsymbol{C}$
Auxiliary: Priority queue Q

$Q \leftarrow \emptyset$, $l \leftarrow 1$
for all $v \in V$ **do**
    $\boldsymbol{R}(v) = v$, $\boldsymbol{P}(v) = \blacktriangle$, $\boldsymbol{L}(v) = 0$, $\boldsymbol{C}(v) = \infty$
    if $v \in S$ **then**
        $\boldsymbol{C}(v) = 0$, $\boldsymbol{L}(v) = l$, $l \leftarrow l + 1$
        $Q \leftarrow v$ with priority $\boldsymbol{C}(v)$
while $Q \neq \emptyset$ **do**
    $v \leftarrow \operatorname{argmin}_{q \in Q}\{\boldsymbol{C}(q)\}$
    $Q \leftarrow Q \setminus \{v\}$
    for all $u \in \boldsymbol{A}(v)$ **do**
        $c \leftarrow \boldsymbol{f}_*(\rho_v \odot \{v, u\})$
        if $c < \boldsymbol{C}(u)$ **then**
            if $\boldsymbol{C}(u) \neq \infty$ **then** $Q \leftarrow Q \setminus \{u\}$
            $\boldsymbol{R}(u) = \boldsymbol{R}(v)$, $\boldsymbol{P}(u) = v$, $\boldsymbol{L}(u) = \boldsymbol{L}(v)$, $\boldsymbol{C}(u) = c$
            $Q \leftarrow u$ with priority $\boldsymbol{C}(u)$
return $(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{P}, \boldsymbol{C})$

Chapter 3

Related Works

The main goal of superpixel algorithms is to represent the objects of interest accurately by the union of their compounding parts. For that, such methods often generate a high quantity of superpixels during partitioning. As a consequence, these methods and oversegmentation ones are often considered to be equivalent, leading to concepts and definitions being used interchangeably. For instance, the popular Felzenszwalb-Huttenlocher algorithm (FH) [36], which has led to hierarchical approaches [43], was long used as a superpixel method by running in an "oversegmentation" mode [71]. In fact, several works based on the watershed transformation [23, 24], especially those in which the *Power Watershed* [31] generalizes, could be classified likewise. In this work, we adopt the same procedure presented in [71], and distinguish both by their ability to control the number of superpixels generated.

In this chapter, we introduce and debate over the most important properties of any superpixel segmentation in Section 3.1. In this work, since we do not desire to perform a survey in superpixel segmentation, we attain ourselves to the most popular, most recent, and most effective state-of-the-art algorithms. For simplicity, we broadly classify them in either classical or deep-learning-based approaches, being each one discussed in Sections 3.2 and 3.3, respectively. For a thorough discussion regarding superpixel segmentation, we suggest reading notable surveys such as [10, 71, 77, 78].

3.1 Superpixel Properties

As one may note, although producing a segmentation with a significantly high number of superpixels commonly presents a high object delineation accuracy, the proper selection of superpixels (for building the objects) becomes burdensome. On the other hand, by generating few superpixels, the latter task becomes easier, but the delineation is often of poor quality. Similarly, for delineation, one may prefer superpixels with irregular shapes. Conversely, regularly-shaped superpixels are most suitable for feature extraction. Simply put, the superpixel segmentation outcome is highly correlated with the domain, which involves biases and morphologies (as specified in Section 1). Alongside most researchers [1, 16, 22, 51, 54, 71, 80], we define the following list of properties as desirable for any superpixel segmentation algorithm:

— **User Control:** The user must be able to control the number of superpixels generated by the algorithm;
— **Connectivity:** Every superpixel must be a connected set of pixels;
— **Boundary Adherence:** The most relevant boundaries must be detected by the borders of the superpixels;
— **Efficiency:** Superpixels should be generated efficiently;
— **Simplicity:** The superpixel method must be straightforward to use;
— **Partition:** Every pixel must be uniquely associated to a superpixel (*i.e.*, disjoint superpixels).;
— **Parsimony:** These properties must be maintained with as few superpixels as possible;

Although some authors [66, 71, 77] indicate compacity and regularity as desirable characteristics, we see both as intrinsically associated with the application domain. That is, in certain contexts and as previously exemplified, the requirement of a superpixel being compact might not improve the application performance. Actually, requiring compact superpixels often prejudices object delineation [71, 75]. In [80], the authors suggest that multiscale segmentation is an important property of superpixel segmentation. Although such a requirement is application-driven, since the user may or may not demand multiple segmentations, we argue that it can be associated with the efficiency principle.

In terms of simplicity, we understand that such a property involves a greater number of circumstances than the mere concept of reducing the number of optimizable parameters. When considering hardware limitations, a simple method can segment with minimal computational power (*e.g.*, not requiring GPUs). Also, a simple method should be a viable solution even when data is a scarce resource (*e.g.*, medical applications). Finally, it is clear that any superpixel method should be easy to: (i) comprehend; (ii) implement; and (iii) adapt, irrespective of the domain. As one may see, limiting the user to simply tune a set of parameters does not imply in simplicity, especially if such are not independent nor intuitive.

Regarding boundary adherence, it is important to notice that the concept of relevant borders is subjected to a preconception of the object of interest. For instance, if a specific object is desired, one may argue that its boundaries are relevant, whereas those of other objects are not. On the other hand, when there is no clear indication of the object of interest, the relevant borders are defined through subjectivity. That is, even parts of a certain object can be considered as possible objects of interest, such as the characters of a post sign, and the criterion for determining the relevance of their boundaries is often based on the application's objective (*e.g.*, optical character recognition or salient object detection). Therefore, we argue that any superpixel method must provide the necessary tools for adapting its strategy to consider the user's notion of relevant borders (or equivalently, of the object of interest), whenever provided, and effectively segment the desired object.

3.2 Classical Methods

As in [10], we can categorize classical methods based on their strategy for generating superpixels (*i.e.*, their main processing): (i) neighborhood; (ii) boundary-evolution; (iii) dynamic center; (iv) path; and (v) hierarchy. It is important to notice that, although many approaches exist which do not fall into any of the previous categories, we argue that they are not as representative (*i.e.*, comprises few algorithms). We cite the *Gaussian Mixture Model* (GMM) [8], the *Entropy Rate Superpixels* (ERS) [54], and the *Differential Evolutionary Superpixels* (DES) [41] as examples of such uncategorized methods.

The first category often requires an initial seed set and recurs to a non-spatially connected neighborhood for generating superpixels. It is the category with the largest number of algorithms, especially due to the popularity of the *Simple Linear Iterative Clustering* (SLIC) [1] method. In SLIC, superpixels are generated using an adapted K-means in the feature space, which combines color and spatial coordinates to promote regularity. Furthermore, SLIC is fast and, although it presents fair delineation, its simplicity favored the development of more effective approaches based on it [39, 55, 81, 92]. As an example, the *Linear Spectral Clustering* (LSC) [52] generates superpixels with high boundary adherence through a weighted K-means in a deliberately tailored 10-dimensional feature space. One common property of all neighborhood-based approaches is the ability to generate compact superpixels independently of the characteristics of the input image. However, by allowing the conquering of non-connected pixels, these methods often require a post processing step for assuring connectivity and, consequently, they may not guarantee the desired superpixel quantity.

The strategy of boundary-evolution algorithms can be summarized as the refinement of the superpixel borders initially defined from a crude but initial partition. This approach commonly leads to highly efficient algorithms, such as *Superpixels Extracted via Energy-Driven Sampling* (SEEDS) [22] and *Iterative Boundaries for implicit Identification for Segmentation* (IBIS) [25]. In SEEDS, superpixels are generated through a coarse-to-fine strategy using a hill-climbing optimization algorithm, whereas IBIS uses the distance measure from [1] to detect and merge non-homogeneous blocks with the superpixel with highest similarity. Another example is the *Efficient Topology Preserving Segmentation* (ETPS) [86], which is inspired by SEEDS, also refines superpixel borders from a grid partitioning, but its pixel-level energy function manages to measure different superpixel properties (*e.g.*, shape regularity and color homogeneity). Since only pixels at superpixel borders are manipulated, these algorithms are the most efficient, and moreover, connectivity is often guaranteed. On the other hand, they usually cannot guarantee the exact number of superpixels desired by the user.

Algorithms such as *Simple Non-Iterative Clustering* (SNIC) [2], are classified as dynamic-center-update approaches since the distance function of its clustering process is based on the features and dynamically updating its centers. For instance, SNIC is a seed-based method which starts off from a grid distribution. However, during clustering, the centroids are updated as superpixels grow. Another example is the *Dynamic Random Walks* (DRW) [48] approach, which, by limiting the walking range, manages to generate superpixels by combining boundary information (*e.g.*, gradient information or boundary

probability). Although such a category presents higher boundary adherence than the previous methods, it results in less compact superpixels.

At the expense of not easily controlling the compacity of their superpixels [71], path-based strategies often lead to top delineation performance. Their strategy can be summarized as clustering pixels by growing paths from the initial seed set. Consequently, a superpixel segmentation is usually a forest rooted in the seed set. From their results, those based on the *Image Foresting Transform* (IFT) [35] framework are the most effective delineation methods. The *Iterative Spanning Forest* (ISF) [75] is a superpixel segmentation framework that partitions the image through iterations of IFT executions on improved seed sets. In contrast to other methods, the ISF steps can be independently defined, which obviously facilitates its adaptability to other applications [28, 59, 70]. For instance, the authors in [16, 17] proposed a generalization of ISF, which incorporates prior object information represented by object saliency maps for improved delineation. Such framework, named *Object-based ISF* (OISF), surpassed several state-of-the-art methods in delineation but has shown high computational complexity during sampling while being significantly influenced by the map's detection quality. The *Eikonal-based Region Growing for efficient Clustering* (ERGC) [27] is an example of a path-based algorithm that does not recur to IFT. For generating superpixels, ERGC uses the Fast Marching algorithm from a regularly-distanced seed set. Then, by enforcing compactness, it builds homogeneous superpixels efficiently and effectively. The *Waterpixels* [57] is another path-based method that builds superpixels through the watershed transformation. Although not directly associated with the IFT, certain works have shown equivalences between watershed and IFT for a specific set of conditions [3, 31, 32].

With the benefit of generating segmentations with distinct superpixel quantity (*i.e.*, scales) in linear time, hierarchical clustering approaches create superpixels by merging pixels and/or superpixels (*i.e.*, bottom-up) or by the converse, dividing image regions (*i.e.*, top-down). Thus, their results often present homogeneous regions with high boundary adherence but, as with path-based methods, present high superpixel irregularity. On the other hand, if one error is present in any scale of the hierarchy, it tends to be propagated and aggravated to the following ones. As examples of hierarchical-based algorithms, we cite the *Recursive ISF* (RISF) [37], which runs several ISFs for building a sparse superpixel hierarchy. On the other hand, the *Superpixel Hierarchy* (SH) [80] recurs to the Borůvka algorithm for computing a minimum spanning forest for each scale. Finally, the *Deep Affinity Network-Hierarchical ERS* (DAL-HERS) [62] aggregates multiscale information to learn pairwise pixel affinities so that the HERS algorithm builds a hierarchical structure like SH, using the Borůvka method, by maximizing the entropy rate.

3.3 Deep-learning-based Methods

In 2018, a new wave of superpixel methods that use deep-learning techniques emerged. Initial proposals recurred to features extracted by a deep-learning algorithm in an attempt to achieve improved delineation. One of such is the *Deep-FLIC* [7], which clusters pixels using original and deep features through the *Fast and Linear Iterative Clustering*

(FLIC) [92]. However, as studied in [74], the simple incorporation of deep features is insufficient for accurate delineation. Therefore, they propose a new loss function, named *SEgmentation-Aware Loss* (SEAL), for learning pixel affinities for improving the delineation performance of its underlying algorithm, ERS.

Other works have developed end-to-end trainable networks for generating superpixels, which we may broadly classify into two categories: (i) fully convolutional and (ii) encoder-decoder. The *Superpixel Sampling Networks* (SSN) [45] is an example of the first group and, to the best of our knowledge, the first end-to-end trainable superpixel network. From its architecture, SSN extracts the image features and cluster pixels using a differential version of SLIC for producing a likelihood-mapping of a pixel belonging to a superpixel (*i.e.*, soft pixel-superpixel assignment). We may cite the *Lifelong Non-iterative Superpixels* (LNSNet) [94] as an example of an unsupervised superpixel algorithm. In LNSNet, superpixels are generated using a lifelong learning strategy and a non-iterative clustering module. Certain algorithms, like the *Regularized Information Maximization* (RIM) [72] integrate the soft pixel-superpixel assignment in the convolutional process and, being unsupervised, train its network based on clustering entropy, spatial smoothness and reconstruction.

Algorithms of the second category build superpixels through an encoder-decoder architecture. The *Superpixel Embedding Network* (SEN) [38] is one of such, and generates superpixels by clustering pixels using a differential mean-shift model, while using SNIC for generating pseudo-ground-truths for training. In *Fully Convolutional Network* (FCN) [85] and in *Association Implantation* (AINet) [79], the authors proposed a U-shaped network for generating the soft-association map in a supervised way. However, the latter uses an association module to associate each pixel with its surrounding superpixels in a grid shape.

As a drawback of end-to-end trainable superpixel methods, most are unable to directly output the hard pixel-superpixel association map, implying the need for a post-processing step. In contrast, the authors in [89] overcome such an issue by proposing an interpolation network named *Superpixel Interpolation Network* (SIN). In SIN, features are extracted from multiple convolutions, and subsequently, multiple interpolations are applied to generate the pixel-superpixel association map while enforcing connectivity.

It is clear that deep-learning algorithms present an interesting strategy toward domain-agnosticism since most of their performance is based on optimization derived from image convolutions and a given ground truth. Yet, a severe drawback of these methods is, clearly, the quantity of data available for training. For supervised ones, it is worsened by the necessity of annotations, which is often a limited resource, especially in the medical domain. Also, these methods are difficult to comprehend, implement, and, to some extent, adapt. Even if fine-tuning is performed, their boundary adherence is moderate in their reported domain. Therefore, application-based properties such as compacity and displacement can be challenging and may significantly impact the delineation result. Lastly, given that such methods often require dedicated hardware to improve their speed, we argue that deep-learning-based approaches are unable to attain the simplicity property.

Chapter 4

Building an Agnostic Superpixel Framework

Although several state-of-the-art superpixel methods achieve high efficacy and efficiency in segmenting natural images, changing the domain significantly impacts their performance, often negatively. Such behavior can be explained by the prior incorporation of specific biases and morphology, which are not valid throughout domains. To overcome such a challenge, it is crucial to define a superpixel segmentation strategy that is not based on domain-specific biases and morphologies to achieve consistent and accurate delineations. Such strategy, we argue, must consider intrinsic characteristics of superpixel segmentation, which, clearly, are domain agnostic. Therefore, when a domain is defined, such a method must allow the incorporation of prior information for improving its performance.

Popularized by SLIC [1], the most used pipeline for superpixel segmentation involves three steps: (i) initial seed sampling; (ii) seed-based superpixel generation; and (iii) seed recomputation. The strategy goes as follows. Starting by selecting a quantity of seeds that approximates to the desired number of superpixels, each one will generate a unique superpixel in the subsequent step. In order to (theoretically) improve the delineation, seeds are reallocated to a more favorable position, often one that minimizes intra-superpixel dissimilarity, generating an improved seed set. Steps (ii) and (iii) are performed either until a convergence criterion is met or a strict number of iterations is performed.

Although it is widely used, this popular pipeline presents a significant set of drawbacks. In step (i), by selecting a number of seeds approximately to the desired quantity of superpixels, it is not assured that the object and its surroundings will contain a sufficient amount of seeds that will lead to its effective delineation. We will term seeds with this causality property as *relevant* seeds, whereas the remaining ones as *irrelevant*. Also, by reallocating seeds in step (iii), the effective delineation obtained in the previous iteration is no longer guaranteed. Regarding the number of iterations (or convergence), one can see that an algorithm that recurs to such a strategy may unavoidably perform more iterations than required with little to no gain in delineation performance. Finally, as a result of such a pipeline, outputting a multiscale segmentation demands executing the same algorithm, but with varying numbers of seeds, several times.

In this chapter, we present *Superpixels through Iterative CLEarcutting* (SICLE) [20,

12, 13, 21], an agnostic superpixel segmentation framework resultant from this thesis, for building effective solutions in any domain. Differently from SLIC and alike, SICLE's three-step pipeline approaches segmentation differently (see Figure 4.1). In step (i), detailed in Section 4.1, the number of seeds sampled is significantly greater than the desired number of superpixels. Superpixels are then generated using the IFT framework, as described in Section 4.2. Section4.3 describes the necessary steps for removing irrelevant seeds while sustaining the effective performance of the previous iteration until reaching the desired number of superpixels (defined by the user). The algorithm and its complexity are presented in Section 4.4. Optionally, one may provide prior object information, represented as object saliency maps, for reshaping, adapting, or improving segmentation for a specific object. Finally, SICLE generalizes two other frameworks, namely *Dynamic and Iterative Spanning Forest* (DISF) [18] and *Object-based DISF* (ODISF) [14], which are also contributions of this thesis.

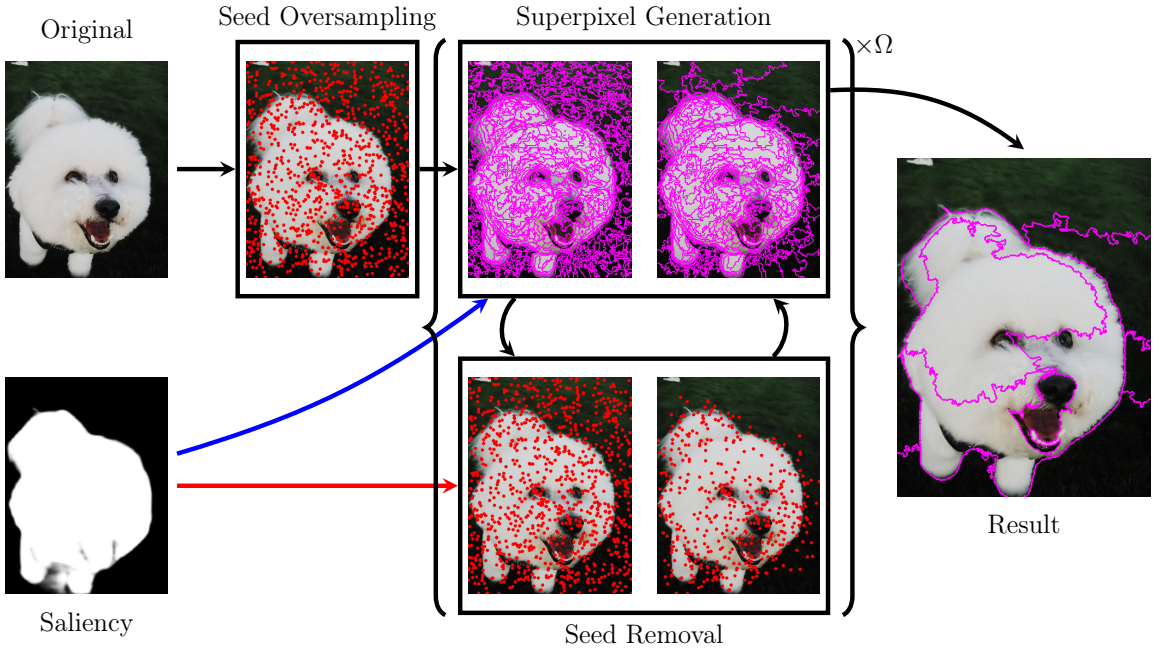


Figure 4.1 – Pipeline for superpixel generation using SICLE (black, red, and blue arrows) and considering the saliency generated by [64]. DISF's pipeline is indicated by black arrows, whereas ODISF's by black and red. A quantity of 1000 seeds were sampled for generating a segmentation with 10 superpixels.

4.1 Seed Oversampling

As previously mentioned, selecting a quantity $N_0 \in \mathbb{N}_+$ of initial seeds $S \subset V$ approximately to the final number of superpixels $N_f \in \mathbb{N}_+$ may miss important object parts. As a result, most methods recur to an equally-distanced seed distribution to exploit most of the image's area, such as in a grid distribution [1, 52, 55, 75]. However, in several cases, the object is many times smaller than the image dimensionality, making it significantly difficult to ensure that the object contains at least one seed within. In fact, since fewer

superpixels are often desirable, such a problem is severely aggravated.

Mathematically speaking, the common state-of-the-art approach demands precision using an imprecise sampling strategy. Conversely, other works use prior object information for more accurate seed distribution, often promoting higher concentrations within the object of interest [16, 19]. However, these approaches demand high computational power, leading to a significant decrease in speed performance in their respective solutions. In this sense, object-based algorithms use high-precision strategies but at the expense of being slower, while classical approaches paradoxically opt for a fast but imprecise algorithm. Therefore, precision and efficiency are inversely correlated.

We argue that *oversampling* offers a better solution since it pursues recall over precision. That is, by selecting a significantly higher number of seeds than the desired number of superpixels (*i.e.*, $N_0 \gg N_f$), we may infer that every object has at least one seed within if N_0 is sufficiently large. Let us denote $R \subseteq V$ as the set of all relevant seeds of the image. Clearly, for each selected seed $s \in V$, the chances of s being relevant are of $|R \setminus S|/|V \setminus S|$. As one may note, the goal of selecting relevant seeds is sustained if $s \in R$, but only the divider is impacted when $s \notin R$. Thus, wrongly adding non-relevant seeds assists on increasing the chances of the subsequent one to be relevant.

Moreover, it is possible to note that seeds that are spatially close and have similar features, such as color and texture, ought to be similarly relevant. Therefore, as we argue, selecting only a proper subset of R suffices for obtaining an accurate delineation, and oversampling assists in finding such seeds. Note that, even when using object saliency maps for guidance in sampling, oversampling would still contain all seeds (up to redundancy) for achieving similar segmentation. However, such strategy results in a handful of irrelevant seeds, which are dealt in step (iii) (*c.f.*, Section 4.3) for ensuring N_f superpixels.

By favoring recall, we may recur to fast and imprecise algorithms, such as the previously mentioned GRID sampling [1, 52, 55, 75], in an oversampling strategy. In this strategy, the goal is to select equally-distanced seeds in a grid-like pattern. While most works assume that the images' axis are of equal length, we provide a different formulation that takes into account any image dimensionality. Let $\rho_* \in]0, 1[$ be the proportion of a given axis over all image's axis. The width's and height's length and proportion, respectively, are $W, H \in \mathbb{N}_+$, and $\rho_W = W/(W+H)$ and $\rho_H = H/(W+H)$. From that, we may determine the number of seeds to be sampled by $N_g = (\rho_W \cdot c)(\rho_H \cdot c)$ where $c \in \mathbb{R}+$ is a constant $c = \sqrt{N_0/(\rho_W \cdot \rho_H)}$. As one may notice, $N_g \approx N_0$. While most works determine the stride between seeds by $\sqrt{N_0/N_f}$, in our approach, the stride is defined proportionally by the axis length: $W/(c \cdot \rho_W)$ and $H/(c \cdot \rho_H)$. Furthermore, we do not shift seeds to the lowest gradient position in the neighborhood for an attempt at better positioning since such a goal is questionable when oversampling.

Still, one may question the benefits of GRID (*i.e.*, equally-distributed seeds) in oversampling. For low values of N_0, equally-distanced seeds aid in sampling most of the image's area. Conversely, as N_0 increases, the stride between seeds decreases sharply, and consequently, seeds tend to be closer to one another, indicating that the image's area is covered by number rather than positioning. Furthermore, if the user provides a non-rectangular mask representing a region of interest to be segmented, one can see that ensuring equidistance amongst seeds while best approximating N_0 can be challenging, if

not impossible, in certain cases. On the other hand, a random seed oversampling, hereafter named RND, may be used to surrogate the previous difficulties. Not only is it simple to implement in any circumstance, being one less step to be selected and optimized by the user, but, backed by the expected relevance redundancy in R, distinct yet equivalently accurate object delineations may be achieved. Note that, since oversampling captures relevant seeds, the goal of achieving a specific segmentation morphology, such as equidistant superpixels, is delegated to step (iii). Finally, although random seed oversampling is not a novel strategy, as seen in the work of [4] for watershed segmentation, it is a strategy that has yet much to be explored for superpixel segmentation.

4.2 Superpixel Segmentation

For generating superpixels, we use the IFT since its effectiveness is widely known in superpixel segmentation [16, 17, 37, 75]. One combination of connectivity function $\boldsymbol{f}_*$ and edge-cost function $\boldsymbol{w}_*$ which yielded accurate object delineation in an interactive segmentation task [26, 69] involves computing the edge-costs dynamically. More formally, let $\boldsymbol{F}(T_s) = \sum_{v \in T_s} \boldsymbol{F}(v)/|T_s|$ be the *mean features* of a *growing* tree T_s rooted in $s = \boldsymbol{R}(v) \in S$. Then, the *Dynamic IFT* (DynIFT) $\boldsymbol{w}_{\text{dyn}}$ can be defined as follows (Equation 4.1):

$$\begin{aligned} \boldsymbol{f}_{\max}(\rho_v \odot \{v, u\}) &= \max\{\boldsymbol{f}_{\max}(\rho_v), \boldsymbol{w}_{\text{dyn}}(v, u)\} \\ \boldsymbol{w}_{\text{dyn}}(v, u) &= \|\boldsymbol{F}(T_s) - \boldsymbol{F}(u)\|_2 \end{aligned} \tag{4.1}$$

where $u, v \in V$ and $\boldsymbol{R}(v) = s$. As one may note, although $\boldsymbol{w}_{\text{dyn}}$ is not smooth, the IFT still outputs a spanning forest.

However, since the tree's mean feature is being built during execution, $\boldsymbol{w}_{\text{dyn}}$ is highly unstable. For different tie-breaking policies for the IFT execution, the mean feature $\boldsymbol{F}(T_s)$ of a tree T_s may differ significantly, which may undesirably lead to distinct delineations. Furthermore, one may question the effectiveness of $\boldsymbol{w}_{\text{dyn}}$ in superpixel segmentation. That is, since superpixels minimize dissimilarity by design, it is expected that $\boldsymbol{F}(T_s) \approx \boldsymbol{F}(s)$, especially for high values of N_f, which are common for superpixel segmentation. For such reasoning, one may suggest using $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$ over $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{dyn}}$ with the arguments of using immutable features and being equivalent in delineation performance.

Both functions are effective but are unable to produce compact superpixels as $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}_{\text{sum}}$ do. Also, although all three may allow differential computations [34, 69] for faster executions of the IFT, they are unaware of the user's desire to segment a specific object. Simply put, like state-of-the-art methods, such functions try to delineate all possible relevant borders in the image, which is an ill-posed task. As one may note, the concept of the object of interest depends on the application domain: for the same computerized scan tomography, each physician may search for different organs with respect to the diagnosis.

In the *Object-based Iterative Spanning Forest* (OISF) [16] framework, the authors used $\boldsymbol{f}_{\text{sum}}$ with a novel edge-cost function for considering prior object information represented by its saliency $\boldsymbol{O} : V \mapsto [0, 1]$, while capable of generating compact superpixels. Equation 4.2

defines such function as:

$$\boldsymbol{w}_{\text{obj}}(v,u) = (\iota\gamma^{\|\boldsymbol{O}(\boldsymbol{R}(v))-\boldsymbol{O}(u)\|_1}\|\boldsymbol{F}(u)-\boldsymbol{F}(\boldsymbol{R}(v))\|_2 + \gamma(\|\boldsymbol{O}(\boldsymbol{R}(v))-\boldsymbol{O}(u)\|_1))^{\beta} + \|v-u\|_2 \tag{4.2}$$

where $\{u,v\} \in E$ and $\gamma \in \mathbb{R}_+$ is a saliency adherence factor. Intuitively speaking, such factor proportionately increases the cost of every path that crosses borders in the given saliency.

As one may note, $\boldsymbol{w}_{\text{obj}}$ is significantly more complex than $\boldsymbol{w}_{\text{root}}$ and $\boldsymbol{w}_{\text{sum}}$, implying that optimizing parameters may be challenging. Furthermore, from the reported results, such edge-cost function achieves significant improvement for low values of N_f superpixels. However, as N_f increases, it sharply converges to an upper bound in delineation. Such an event can be explained by the strong adherence to the saliency borders, necessary for few superpixels but extreme for highly specialized superpixels (*i.e.*, high N_f). This condition can be severely aggravated by incorrect saliency estimations, which would reinforce incorrect borders and deteriorate segmentation. Therefore, for a sufficiently high N_f, such a function may perform worse than $\boldsymbol{w}_{\text{root}}$ and $\boldsymbol{w}_{\text{sum}}$, even considering object saliency maps.

Thus, the goal is to extend the properties of $\boldsymbol{w}_{\text{root}}$ and $\boldsymbol{w}_{\text{sum}}$ for considering prior object information with parsimony. For this specific latter condition, we may recur to a *saliency confidence factor* $\alpha \in \mathbb{R}_+$, which indicates the degree of trust the user has towards the saliency variation $\|\boldsymbol{O}(\boldsymbol{R}(p)) - \boldsymbol{O}(q)\|_1 \in [0,1]$ between the root of p and q. That is, $\alpha\|\boldsymbol{O}(\boldsymbol{R}(p)) - \boldsymbol{O}(q)\|_1$. From that, we propose the generalization of $\boldsymbol{w}_{\text{root}}$ (*c.f.*, Equation 2.5), as shown in Equation 4.3:

$$\boldsymbol{w}^{\alpha}_{\text{root}}(p,q) = (\|\boldsymbol{F}(\boldsymbol{R}(p)) - \boldsymbol{F}(q)\|_2)^{1+\alpha\|\boldsymbol{O}(\boldsymbol{R}(p))-\boldsymbol{O}(q)\|_1} \tag{4.3}$$

If $\alpha = 0$, one can see that $\boldsymbol{w}^{\alpha}_{\text{root}} = \boldsymbol{w}_{\text{root}}$ and avoid incorporating the information of a poor-quality saliency map. For this property, we term as *(saliency) robustness*. Conversely, for $\alpha > 0$, the cost of crossing saliency borders is increased, assisting in leaking prevention and improving delineation.

Yet, $\boldsymbol{w}^{\alpha}_{\text{root}}$ is unable to produce compact superpixels as $\boldsymbol{w}_{\text{sum}}$ does. Therefore, we also propose a generalization $\boldsymbol{w}^{\alpha}_{\text{sum}}$ of $\boldsymbol{w}_{\text{sum}}$ using a different strategy. Since $\boldsymbol{w}_{\text{sum}}$ elevates the cost to a power of β, the boundary factor, and one recurs to such function to produce regular superpixels, the goal is to proportionally disregard regularity in specific, but crucial, locations (*i.e.*, saliency variations). Therefore, our proposal is defined as follows (Equation 4.4)

$$\boldsymbol{w}^{\alpha}_{\text{sum}}(p,q) = ((\iota + \alpha\|\boldsymbol{O}(\boldsymbol{R}(p)) - \boldsymbol{O}(q)\|_1)\|\boldsymbol{F}(q) - \boldsymbol{F}(\boldsymbol{R}(p))\|_2)^{\beta} + \|p-q\|_2 \tag{4.4}$$

Similarly, if $\alpha = 0$, one can see that $\boldsymbol{w}^{\alpha}_{\text{sum}} = \boldsymbol{w}_{\text{sum}}$ and promotes robustness, while for $\alpha > 0$, the regularity constraint is diminished over the color dissimilarity when a path crosses saliency borders, favoring delineation accuracy. Finally, when $\boldsymbol{O}$ is not provided, setting $\boldsymbol{O}(p) = 1$ for every $p \in V$ leads to equivalent behavior of $\alpha = 0$ for both edge-cost functions. Figure 4.2 illustrates the impacts of varying α when considering $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$. It is possible to note that when $\alpha = 0$, several leakings are present in the segmentation, which are slightly corrected when α is 0.33 and more significantly for 0.66. However, for

$\alpha = 1$, the contours of the segmentations highly resemble those in the saliency map.

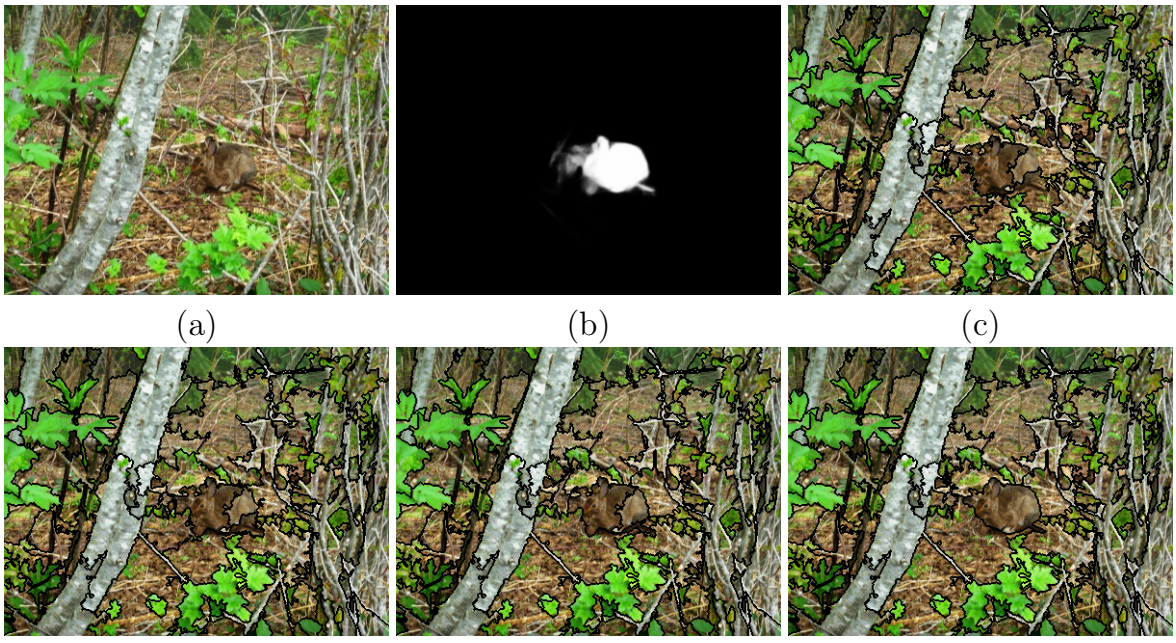


Figure 4.2 – Impact of α in superpixel generation on an (a) image considering (b) a saliency map generated by [64]. The $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}^{\alpha}_{\text{root}}$ were considered, $N_0 = 3000$ and $N_f = 100$, and the values of α were (c-f) 0, 0.33, 0.66, and 1.

4.3 Seed Removal

As previously mentioned, several state-of-the-art algorithms sample $N_0 \approx N_f$ seeds and recompute each seed for an improved seed set for the next iteration. The most common approach involves selecting the superpixel *medoid* for the subsequent iteration. More formally, the medoid is the pixel $p \in T_s$, where $s = \boldsymbol{R}(p)$ with minimum feature dissimilarity such that $\|\boldsymbol{F}(T_s) - \boldsymbol{F}(p)\|_2 \leq \|\boldsymbol{F}(T_s) - \boldsymbol{F}(q)\|_2$ for any other $q \in T_s$. Some works [75] select the *centroid*, which can be similarly defined as the pixel p whose spatial position is closest to the superpixel's center of mass $\boldsymbol{c}(T_s) = \sum_{v \in T_s} v/|T_s|$ (*i.e.*, $\|p - \boldsymbol{c}(T_s)\|_2 \leq \|q - \boldsymbol{c}(T_s)\|_2$). Either way, the goal is to minimize intra-superpixel dissimilarity based on some criterion. Finally, this update process occurs for a strict number of iterations defined by the user.

Often, this strategy results in more accurate delineation since favoring homogeneity prevents, clearly, conquering dissimilar pixels, which could hamper delineation (*i.e.*, object borders). However, such improvement is tied to the initial number of seeds N_0. That is, as the number of superpixels decreases (and thus affects N_0 in most works), conquering dissimilar pixels is usual since the lack of competition does not promote increased specificity of the superpixel's features. Consequently, it is not possible to ensure a final set of relevant seeds, and the accuracy of state-of-the-art methods significantly deteriorates for a reduced number of superpixels. Conversely, if N_0 increases, shifting may not impact

segmentation since a grouping of similar pixels converges into a similar point and produces similar superpixels. Concerning the number of iterations, we argue that imposing a strict number of iterations for segmentation, as done by many works [75, 1, 55, 52], is inefficient since it depends on N_f. As mentioned, in high competition environments (*i.e.*, high N_f), superpixels are highly specialized, and several iterations would impact minimally the delineation since the seed reallocation is also subtle. On the other hand, in low competition environments, caused by lowering N_f, more iterations are needed for convergence since the conquering of dissimilar pixels imposes a major difficulty in finding optimal minima for delineation.

In SICLE, however, such concern is mitigated by oversampling. That is, backed by the premise that oversampling builds a seed set S that contains a subset R of relevant seeds for accurate segmentation, the goal is to conceive a strategy for selecting N_f seeds from R. If such an ideal approach is known, one can see that SICLE would require only $k > 1$ iterations to produce $k - 1 \in \mathbb{N}_+$ segmentations with maximum delineation performance. Simply put only two iterations for a single segmentation and $k+1$ iterations for a multiscale segmentation with k scales.

It is challenging to obtain the smallest subset R of relevant seeds for generating segmentations with maximum accuracy with a minimum number of superpixels $N_f = |R|$. It is expected that R be smaller for domains with homogeneous objects and backgrounds (due to higher redundancy) and greater as the complexity of both increases. However, if $N_f < |R|$, relevant seeds are unavoidably removed to satisfy the user's demand and, therefore, not being able to assure nor sustain its expected result. In conclusion, as we argue, such a strategy is effective for superpixel segmentation, not for segmentation.

Since the ideal selection of seeds is often unknown, and removing $N_0 - N_f$ seeds in a single iteration could hamper delineation, one may opt for removing only a portion of irrelevant seeds until reaching N_f seeds for the last IFT segmentation. The goal is to let each seed $s \in S$ compete amongst its pairs and recur to the characteristics of its respective superpixel T_s for computing the relevance of s. From such comparison, we order and maintain $\boldsymbol{M}(i) \in \mathbb{N}_+$ most relevant seeds, removing the remaining ones, for a novel "relevance testing" (or "perturbance"). Thus, in this approach, the most relevant seeds are consistently labeled as such throughout the iterations.

We term as *seed preservation curve* $\boldsymbol{M}$ a strictly monotonically decreasing curve that defines the number of seeds to maintain at each iteration $i \in \mathbb{N}$, starting from $\boldsymbol{M}(0) = N_0$, until reaching $\boldsymbol{M}(i) = N_f$. Although one may suggest any possible curve, we exploit an intrinsic behavior caused by oversampling. At $i = 1$, it is expected that, aside from numerous irrelevant seeds, many regions contain several equally-relevant seeds. Consequently, by ordering the seeds by their relevance, the portion of the least relevant ones should present instances of the previous cases and, thus, removing a large quantity should impact the object delineation minimally. Throughout the iterations, the number of seeds decreases, and the least relevant ones are more defined by relation amongst its pairs (*i.e.,* less relevant than), than by being unambiguously irrelevant (*i.e.*, almost zero relevance). Also, fewer seeds are directly associated with reducing the seed competition, implying that any substantial removal could severely impact delineation, causing major "leakings". In this case, removing a small quantity of seeds should prevent superpixel growth instability

and minimize errors. In conclusion, we observe that an exponential function $\boldsymbol{M}$ best fits for estimating the portion of $\boldsymbol{M}(i)$ seeds to maintain for the next iteration $i+1$. As an example, one may define $\boldsymbol{M}$ as (Equation 4.5):

$$\boldsymbol{M}(i) = \max\{N_0 \cdot e^{-i}, N_f\} \tag{4.5}$$

, as performed in [14, 18], where e is the Euler's number.

However, this strategy does not limit the number of iterations to a user-defined maximum number $\Omega \geq 2$, if such a number is given. For instance, for an unsatisfactory segmentation due to N_0, one may increase N_0 for an improved result but expect the same number of iterations. From the latter equation, such modification inevitably leads to more iterations even for the same final quantity N_f of superpixels. Therefore, we propose a generalization of the latter equation, as presented in Equation 4.6

$$\boldsymbol{M}(i) = \max\{(N_0)^{1-\omega \cdot i}, N_f\} \tag{4.6}$$

, where $\omega \in \mathbb{R}_+$ is a *seed decay factor*. For limiting the number of executions to Ω, one may define $\omega = 1/(\Omega - 1)$. On the other hand, for equivalating to Equation 4.5, one can define $\omega = \log_{N_0} e$, since:

$$\begin{aligned}
\boldsymbol{M}(i) &= \max\{(N_0)^{1-(\log_{N_0} e)\cdot i}, N_f\} \\
&= \max\{(N_0 \cdot \left(N_0^{(\log_{N_0} e)}\right)^{-i}, N_f\} \\
&= \max\{(N_0 \cdot e^{-i}, N_f\}
\end{aligned}$$

For producing a multiscale segmentation, the user may provide the non-increasing sequence of desired scales such that $\boldsymbol{M}(i)$ at each iteration produces the i-th scale. Furthermore, as Figure 4.3 shows, since only a portion of seeds is removed at each iteration i, while the remaining ones are maintained for iteration $i+1$, one can see that SICLE tends to produce significantly similar segmentations, mimicking the behavior of a hierarchical superpixel segmentation.

4.3.1 Seed Relevance

If a *seed relevance function* $\boldsymbol{V}_*(s) \in \mathbb{R}_+$ can accurately measure whether a seed $s \in S$ is relevant (*i.e.*, $s \in R$), then SICLE would require only two iterations for obtaining a segmentation with N_f superpixels generated from only relevant seeds. However, as we argue, it is impossible to conceive a single $\boldsymbol{V}_*$ that is ideal for all possible domains, especially due to significant background and, most importantly, object variations. Therefore, we say that it is more profitable to allow SICLE, through $\boldsymbol{V}_*$, to be easily adapted for any domain. Simply put, we allow the user to provide a $\boldsymbol{V}_*$, which is highly specialized for its application domain since, again, relevance is subjective to the user.

Yet, certain assumptions and premises on superpixel segmentation may assist in providing an effective, but not ideal, selection of relevant seeds in several cases. For instance, based on the IFT framework, the size of a superpixel T_s whose seed is $s \in S$ denotes



(a) (b) (c) (d)

Figure 4.3 – Multiscale segmentation generated by SICLE demanding (a-d) 500, 300, 200, and 100 superpixels. The initial seed quantity was $N_0 = 500$, whilst $N_f = 100$, and $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}_{\text{sum}}$ were considered.

the ability of such seed to conquer pixels (thus, bigger implies better positioning). On the other hand, small superpixels can be associated with local minima, being unable to achieve bigger sizes. Therefore, the *size criterion* $\boldsymbol{V}_{\text{size}}(s)$ can be mathematically defined by Equation 4.7:

$$\boldsymbol{V}_{\text{size}}(s) = |T_s|/|V| \tag{4.7}$$

However, size alone does not provide sufficient information on the relevance of a superpixel T_s. For instance, homogeneous regions tend to produce big but significantly similar superpixels, whereas heterogeneous zones often indicate the encounter of boundaries between objects or parts of complex objects. Thus, we recur to the *tree-neighborhood* $\boldsymbol{A}(T_s)$, which can be defined as follows (Equation 4.8):

$$\boldsymbol{A}(T_s) = \{T_t : q \in \boldsymbol{A}(p) \text{ for some } p \in T_s \text{ and } q \in T_t\} \tag{4.8}$$

where $s, t \in S$. Intuitively speaking, two trees are *neighbors* if their extensions meet in some point.

From this neighborhood, we may now evaluate whether T_s is surrounded by similar or dissimilar superpixels $T_t \in \boldsymbol{A}(T_s)$. Since we aim to identify superpixels in heterogeneous zones, we may quantify such relevance based on the contrast within the neighborhood. Likewise, contrast alone is insufficient for determining relevance. As mentioned, heterogeneous superpixels may indicate local minima or parts of complex objects, whereas homogeneous superpixels can also be a significant part of the object or a significant part of the background. Therefore, one may combine contrast and size, disregarding small superpixels, and conceive the *maximum-contrast and size criterion* $\boldsymbol{V}_{\text{maxsc}}(s)$ and the *minimum-contrast and size criterion* $\boldsymbol{V}_{\text{minsc}}(s)$, formally defined as (Equation 4.9):

$$\begin{aligned} \boldsymbol{V}_{\text{maxsc}}(s) &= \boldsymbol{V}_{\text{size}}(s) \max_{T_t \in \boldsymbol{A}(T_s)} \{\|\boldsymbol{F}(T_s) - \boldsymbol{F}(T_t)\|_2\} \\ \boldsymbol{V}_{\text{minsc}}(s) &= \boldsymbol{V}_{\text{size}}(s) \min_{T_t \in \boldsymbol{A}(T_s)} \{\|\boldsymbol{F}(T_s) - \boldsymbol{F}(T_t)\|_2\} \end{aligned} \tag{4.9}$$

That is, the former aims to identify big superpixels in any distinctive border since it maximizes dissimilarity amongst the neighbors. Therefore, the proximity to a possible border is sufficient to be relevant. On the other hand, the latter tries to identify big superpixels located in highly heterogeneous regions since the most relevant seed, with the highest relevance value, is computed by the highest minimum neighborhood dissimilarity.

Finally, in some applications, the user expects that superpixels are not concentrated in any region but spread evenly in the image. For that, the previous functions are of no use since they select seeds close to probable object borders. Therefore, we propose the *spread criterion* $\boldsymbol{V}_{\text{spread}}(s)$ as defined by Equation 4.10:

$$\boldsymbol{V}_{\text{spread}}(s) = \boldsymbol{V}_{\text{size}}(s) \min_{T_t \in \boldsymbol{A}(T_s)} \{\|\boldsymbol{c}(T_s) - \boldsymbol{c}(T_t)\|_2\} \tag{4.10}$$

Intuitively speaking, $\boldsymbol{V}_{\text{spread}}$ favors seeds that are far apart from all neighbors based on their centroids $\boldsymbol{c}$, promoting an even distribution. If a maximum operator was considered, it would only require one far neighbor for a seed to achieve high relevance, which is undesirable. Finally, Figure 4.4 shows and compares the results obtained for each $\boldsymbol{V}_*$ for selecting relevant seeds.

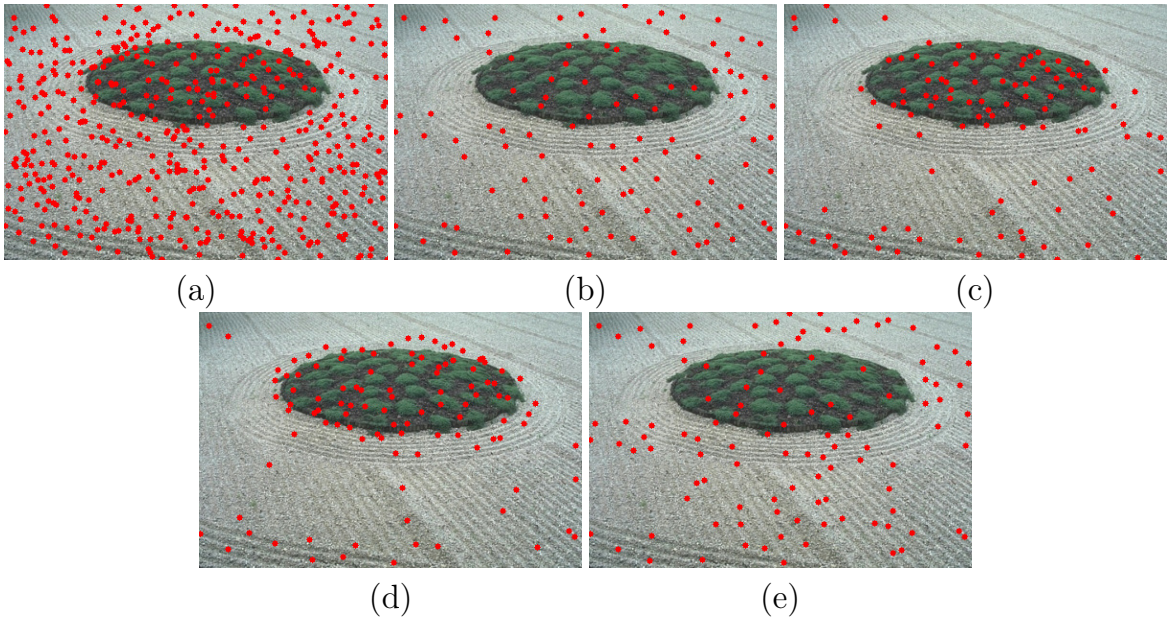


Figure 4.4 – Seed relevance criterion impact on a (a) set of 500 seeds using $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}^{\alpha}_{\text{sum}}$. A number of 106 were selected using (b-e) $\boldsymbol{V}_{\text{size}}$, $\boldsymbol{V}_{\text{minsc}}$, $\boldsymbol{V}_{\text{maxsc}}$, and $\boldsymbol{V}_{\text{spread}}$.

4.3.2 Seed Penalization

Still, the relevance $\boldsymbol{V}_*(s)$ of a seed s is, again, subjective since it is strongly tied to which object is desired, making it irrelevant when it does not assist in the expected morphology of the object of interest. As inferred, only a subset of R is needed for effective segmentation, but considering all objects in the image. One may note that if a single O object is desired, the necessary set of relevant seeds R_O may be much smaller, indicating not only more irrelevant seeds to be removed but the ability to sustain effective delineation

for smaller values of N_f. Therefore, one may consider SICLE with object information for segmentation (rather than superpixel segmentation) if, and only if, the object O is simple (*e.g.*, non-texturized and homogeneous), leading to smaller R_O, and $N_f \geq |R_O|$. Otherwise, it is no longer able to ensure accurate segmentation for low values of $N_f < |R_O|$.

If no specific object is desired, then one may define $\boldsymbol{O}(p) = 1$ for all $p \in V$. But, if an object is desired, then one could expect more superpixels within or near it in order to promote its accurate delineation. Clearly, superpixels that do not present such characteristics are of no use to the user's needs and, even though they may present high "relevance", they should be penalized by some *seed penalization function* $\boldsymbol{B}_*(s) \in \mathbb{R}_+$ based on the saliency $\boldsymbol{O}$ provided by the user. For the previous case of prioritizing seeds within or near the object, one may recur to the *object penalization* $\boldsymbol{B}_{\text{obj}}(s) \in \mathbb{R}_+$ as presented by Equation 4.11:

$$\boldsymbol{B}_{\text{obj}}(s) = \max\{\boldsymbol{O}(T_s), \max_{T_t \in \boldsymbol{A}(T_s)}\{\|\boldsymbol{O}(T_s) - \boldsymbol{O}(T_t)\|_2\}\} \tag{4.11}$$

where $\boldsymbol{O}(T_s) = \sum_{v \in T_s} \boldsymbol{O}(v)/|T_s|$ is the *mean saliency* of the superpixel T_s. Simply put, $\boldsymbol{B}_{\text{obj}}$ penalizes seeds s whose superpixels is mostly in the background and far from the object borders.

Although one may agree that concentrating superpixels within and near the object can improve delineation significantly, it also favors internal superpixels, which exert limited influence on delineation. That is, when irregularity and delineation are of utmost importance, the goal is to populate the object's borders with superpixels (rather internally) and, for that, we propose the *border penalization* $\boldsymbol{B}_{\text{border}}(s) \in \mathbb{R}_+$ (Equation 4.12):

$$\boldsymbol{B}_{\text{border}}(s) = \max_{T_t \in \boldsymbol{A}(T_s)}\{\|\boldsymbol{O}(T_s) - \boldsymbol{O}(T_t)\|_2\} \tag{4.12}$$

Differently from $\boldsymbol{B}_{\text{obj}}$, $\boldsymbol{B}_{\text{border}}$ disregards $\boldsymbol{O}(T_s)$, penalizing every superpixel T_s which is internal to the object or to the background.

Clearly, by focusing on maximizing delineation at any cost, both previous penalizations lead to an uneven superpixel distribution (*i.e.*, near the object), which, even enforcing compacity, would lead to irregularly-sized superpixels, which may not be desirable. Conversely, if saliency is provided, it is expected to favor the object delineation to the detriment of the others. Thus, it is essential to provide a penalization that minimizes the impacts on the overall segmentation compactness whilst improving the object delineation.

One possible approach is to take advantage of the user's usual goal. That is, since more seeds are expected to be within the object, and such is often small with respect to the image, object superpixels tend to be small in size. This condition may improve delineation at the expense of promoting irregularity. Based on similar reasoning, one may favor spread background superpixels since they tend to be small in number but big in size. This strategy (of spreading) allows for even growth between superpixels when compacity is desired. More formally, the *object and spread background penalization* $\boldsymbol{B}_{\text{osb}}(s) \in \mathbb{R}_+$ can be defined as follows (Equation 4.13):

$$\boldsymbol{B}_{\text{osb}}(s) = \max\{\boldsymbol{O}(T_s), (1 - \boldsymbol{O}(T_s)) \min_{T_t \in \boldsymbol{A}(T_s)}\{\|\boldsymbol{c}(T_s) - \boldsymbol{c}(T_t)\|_2\}/d\} \tag{4.13}$$

where $d \in \mathbb{R}_+$ is the maximum distance between any two vertices in the graph. In simple terms, $\boldsymbol{B}_{\text{osb}}$ penalizes background superpixels with close neighbors.

By the same assumption for $\boldsymbol{B}_{\text{border}}$, it is possible to infer that superpixels within the object offer negligible influence on the overall compactness of the segmentation due to their size. Thus, prioritizing internal object superpixels may not favor either delineation or compactness. On the other hand, we may focus on those object superpixels near the object borders for improving delineation, with minimal or no impact on compactness. For those in the background, we may favor spreading as $\boldsymbol{B}_{\text{osb}}$ promotes. In summary, the *border of the object and background spreading penalization* $\boldsymbol{B}_{\text{bobs}} \in \mathbb{R}_+$ can be formally described by Equation 4.14

$$\boldsymbol{B}_{\text{bobs}}(s) = \max\{\boldsymbol{B}_{\text{border}}(s), (1 - \boldsymbol{O}(T_s)) \min_{T_t \in \boldsymbol{A}(T_s)} \{\|\boldsymbol{c}(T_s) - \boldsymbol{c}(T_t)\|_2\}/d\} \tag{4.14}$$

Therefore, the final relevance of a seed $s \in S$ is determined by $\boldsymbol{V}_*(s)\boldsymbol{B}_*(s)$ computed considering the characteristics of its respective superpixel T_s. When $\boldsymbol{O}$ is not provided, we may obtain $\boldsymbol{B}_*(s) = 1$ for every seed by setting $\boldsymbol{O}(p) = 1$ for every $p \in V$. On the other hand, if the user desires to concentrate superpixels in the background, it is either possible through conceiving a novel penalization or, more simply, defining the new saliency as $1 - \boldsymbol{O}(p)$. We provide examples to illustrate the impact of penalization in Figure 4.5.

4.4 Algorithm

From all three steps, one can see that SICLE is bounded by the IFT execution, which is $\mathrm{O}(|V| \log |V|)$. Seed oversampling requires sampling N_0 seeds, which, from both GRID and RND approaches, is bounded by $\mathrm{O}(N_0)$. On the other hand, the seed removal step needs to compute all seed relevances, rank using a heap (for instance), and select $\boldsymbol{M}(i)$ most relevant seeds at iteration i, which demands a complexity of $\mathrm{O}(|S| \log |S|)$.

For generating a multiscale segmentation $\mathcal{M}$ of k scales, SICLE requires a single run (*i.e.*, $\mathrm{O}(1)$). Mathematically speaking, the number of iterations $t \in \mathbb{N}_+$ to achieve N_f, and, subsequently, perform the last iteration, can be known beforehand through:

$$\begin{aligned} N_0^{1-\omega t} = N_0/N_0^{\omega t} = N_f \text{ and } N_0^{\omega t} = N_0/N_f \\ \omega t = \log_{N_0}(N_0/N_f) = 1 - \log_{N_0} N_f \text{ thus } t = \lceil (1 - \log_{N_0} N_f)/\omega \rceil \end{aligned}$$

Therefore, the number of iterations to be performed is $t+1$. At each iteration $i \in [1, t+1]$, $\mathcal{M}$ includes the result of the IFT segmentation as a partition by assigning $\boldsymbol{L}(\boldsymbol{R}(p))$ to every $p \in V$, as presented in Algorithm 2.

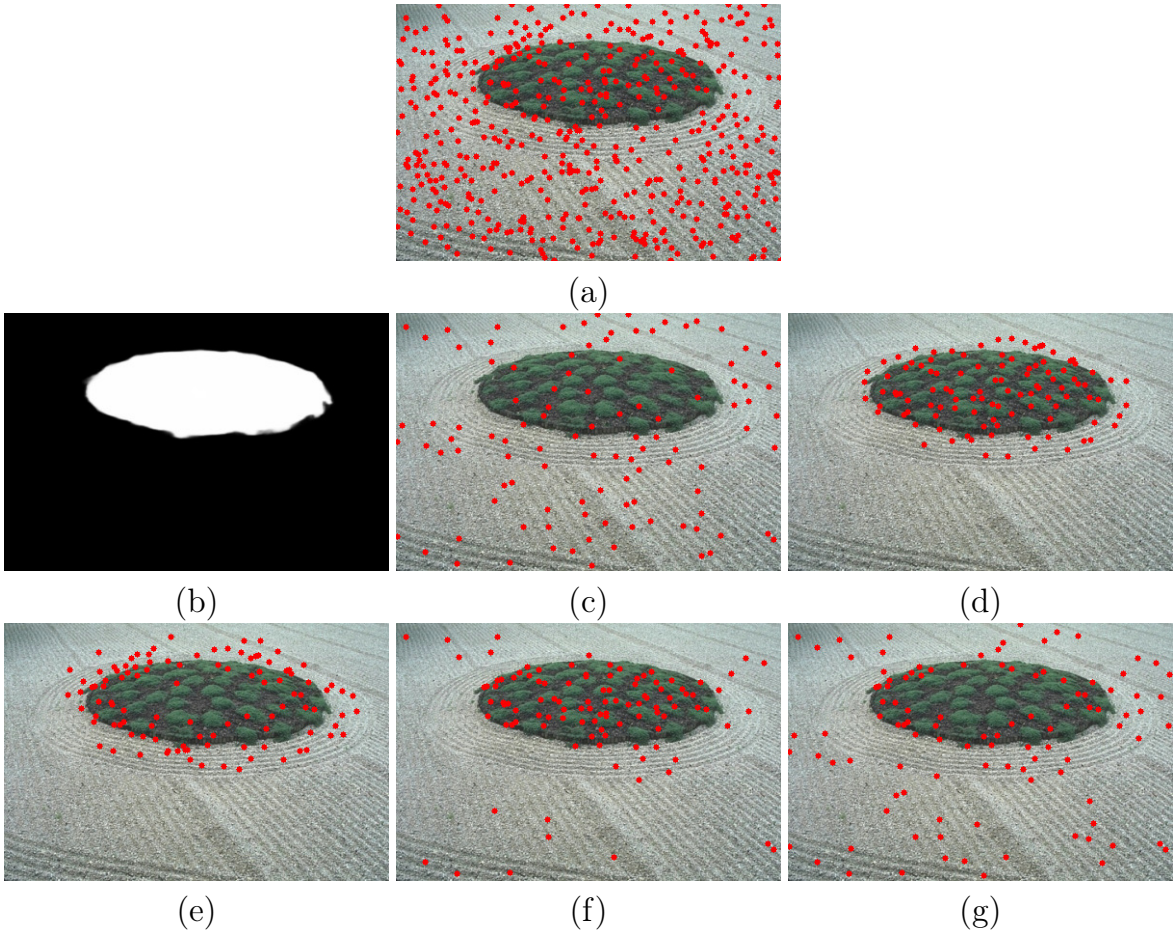


Figure 4.5 – Seed relevance penalization impact on a (a) set of 500 seeds considering the saliency (b) generated by [64]. The criterion used was $\boldsymbol{V}_{\text{spread}}$ for selecting 106 seeds and the penalization performed was (c-g) none, $\boldsymbol{B}_{\text{obj}}$, $\boldsymbol{B}_{\text{border}}$, $\boldsymbol{B}_{\text{osb}}$, and $\boldsymbol{B}_{\text{bobs}}$.

Algorithm 2 SICLE Algorithm

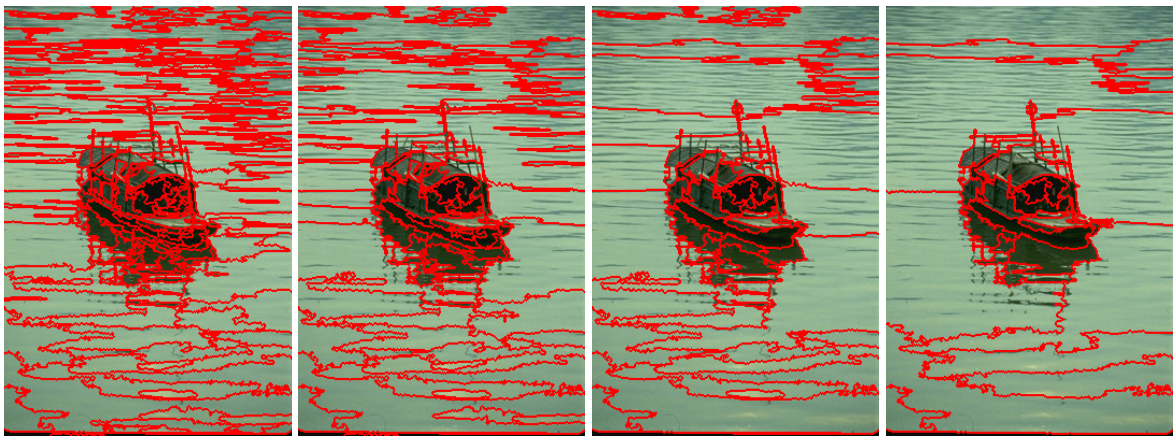
Input: Graph $G = (V, E)$, initial seed quantity N_0, final number of superpixels N_f, connectivity function $\boldsymbol{f}_*$,, seed relevance function $\boldsymbol{V}_*$, seed penalization function $\boldsymbol{B}_*$
Output: Multiscale segmentation $\mathcal{M}$
Auxiliary: Priority queue Q

$S \leftarrow \text{SeedOversampling}(N_0)$
$i = 1$
while $|S| \geq N_f$ **do**
  $Q \leftarrow \emptyset$
  $(\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{P}, \boldsymbol{C}) \leftarrow \text{IFT}(G, S, \boldsymbol{f}_*)$
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{P_s : P_s = \{p \in V : \boldsymbol{R}(p) = s\} \; \forall \; s \in S\}$
  for all $s \in S$ **do**
    $Q \leftarrow s$ with priority $\boldsymbol{B}_*(s)\, \boldsymbol{V}_*(s)$
  while $|S| > \boldsymbol{M}(i)$ **do**
    $s \leftarrow \arg\min_{s \in Q}\{\boldsymbol{B}_*(s)\, \boldsymbol{V}_*(s)\}$
    $S \leftarrow S \setminus \{s\}$, $Q \leftarrow Q \setminus \{s\}$
  $i \leftarrow i + 1$
return $\mathcal{M}$

Chapter 5

Evaluating Hierarchiness

Several superpixel methods are unable to produce a multiscale segmentation on the fly, irrespective of their efficiency and efficacy. Although SICLE is capable of such a feature, it does not ensure that such a series is hierarchical (Figure 5.1), providing an even greater set of properties already stated in this work, since the borders of a previous scale are not necessarily guaranteed in the subsequent one. Yet, since SICLE mimics and favors the generation of similar segmentations by only removing seeds (rather than reallocation), it is uncertain whether such multiscale segmentation, to some extent, may provide the set of theoretical properties of hierarchies either partially or locally. For that, a set of measures is deemed necessary for providing such insight for any multiscale segmentation.



(a) (b) (c) (d)

Figure 5.1 – Non-hierarchical multiscale generated by SICLE requiring (a-d) $N_f \in \{200, 100, 50, 25\}$ considering $N_0 = 1000$, $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$, and $\boldsymbol{V}_{\text{minsc}}$.

In this chapter, we introduce the concepts of *nucleus* and *cover* in Section 5.1, which are crucial for identifying a specific case, amongst eight different ones, as detailed in the subsequent section (*i.e.*, Section 5.2). From such, it is possible to evaluate how a region in a partition $S_{i+1} = Y$ is built from the regions of $S_i = X$ in a multiscale segmentation $\mathcal{M} = \{S_1, \ldots, S_k\}$ for $1 \leq i < k \in \mathbb{N}$. Finally, we propose three measures in Section 5.3 for estimating the hierarchiness between regions, partitions and, in the long run, of a multiscale segmentation.

5.1 Nucleus and Cover

Given a layer $X = \{X_1, \ldots, X_r\}$ and its successor $Y = \{Y_1, \ldots, Y_c\}$, where $r = |X|$ and $c = |Y|$, within $\mathcal{M}$, we may define the *nucleus* of a coarser region $Y_j \in Y$ as the group of blocks $X_i \in X$ at the finer scale totally included in the former (*i.e.*, nested). More formally, the nucleus $\eta(X, Y_j)$ can be defined by Equation 5.1

$$\eta(X, Y_j) = \{X_i \in X : X_i \subseteq Y_j\} \tag{5.1}$$

in which $1 \leq i \leq r$ and $1 \leq j \leq c$. In contrast, the *cover* of a coarser region Y_j can be seen as the set of regions X_i in the preceding scale necessary for building the former. In this sense, building is the partial or total selection of elements in each region in the cover so that the result is equivalent. Equation 5.2 properly defines the cover $\widehat{\eta}(X, Y_j)$ as:

$$\widehat{\eta}(X, Y_j) = \{X_i \in X : X_i \between Y_j\} \tag{5.2}$$

where $X_i \between Y_j$ indicates that $X_i \cap Y_j \neq \emptyset$ (*i.e.*, overlap). When this is not the case, we say that X_i and Y_j are *disjoint*. Finally, all previous concepts are also applicable to any region $X_i \in X$ with respect to Y.

As one may note, the nucleus $\eta(X, Y_j)$ may not build Y_j, implying that $\cup\eta(X, Y_j) \subseteq Y_j$. Also, since every region $X_i \in X$ contained in Y_j leads to $X_i \between Y_j$, we obtain $\cup\eta(X, Y_j) \subseteq \cup\widehat{\eta}(X, Y_j)$. Finally, since the cover $\widehat{\eta}(X, Y_j)$ may present regions that partially cover Y_j, aside from those totally comprised in it, we have that $Y_j \subseteq \cup\widehat{\eta}(X, Y_j)$. Thus, $\cup\eta(X, Y_j) \subseteq Y_j \subseteq \cup\widehat{\eta}(X, Y_j)$ and, consequently, $|\cup\eta(X, Y_j)| \leq |Y_j| \leq |\cup\widehat{\eta}(X, Y_j)|$.

5.2 Cases

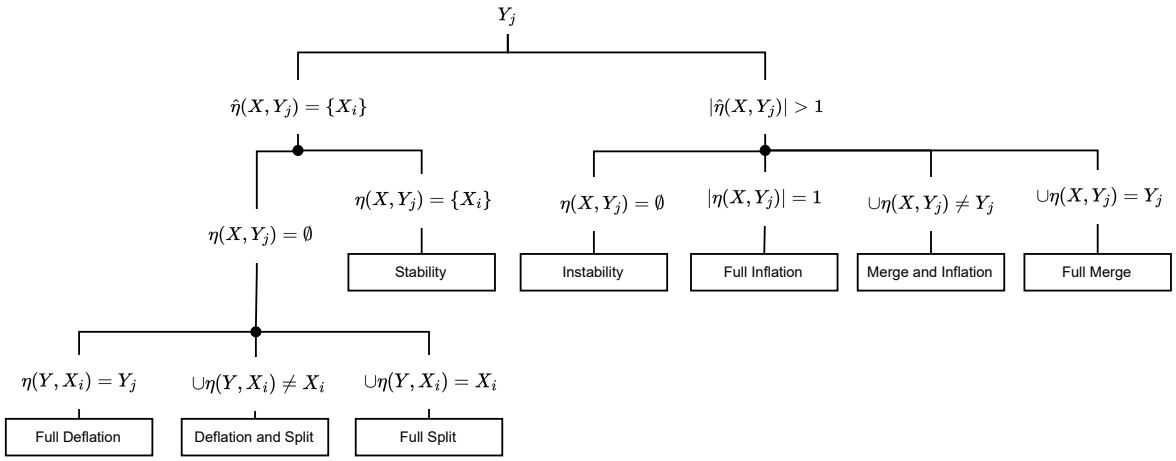


Figure 5.2 – Diagram depicting all eight possible cases for building a region $Y_j \in Y$ from the regions of the partition X.

In Figure 5.2, we are able to identify eight possible cases. When the cover $|\widehat{\eta}(X, Y_j)| > 1$, we infer that a single region $X_i \in \widehat{\eta}(X, Y_j)$ is insufficient for building Y_j, indicating that

Y_j has *grown*, with respect to X. However, it is uncertain whether such growth leads to the total (or partial) inclusion of other regions in X. Thus, we recur to the nucleus $\eta(X, Y_j)$ for distinction. When $\cup\eta(X, Y_j) = Y_j$, it is possible to state that $\eta(X, Y_j) \in \Pi(Y_j)$ and, consequently, possible to totally build it by merging the regions of the nucleus. For that situation, we name it as *full merge*.

On the other hand, if $|\eta(X, Y_j)| = 1$, only a region X_i is totally comprised in Y_j, meaning the remaining $Y_j \setminus X_i$ elements were aggregated to Y_j during its growth. Yet, it was insufficient to merge the overlapping regions. We name such case as *full inflation* since Y_j is built by fully inflating X_i only. When the nucleus $|\eta(X, Y_j)| > 1$ is composed of more than one region, but $\cup\eta(X, Y_j) \neq Y_j$ indicates that their merging is insufficient for building Y_j, requiring inflation, we term such case as *merge and inflation*.

In contrast with previous cases, associated with the growth of the regions in X to build Y_j, when the cover $\widehat{\eta}(X, Y_j)$ equals a single element $\{X_i\}$, no growth occurred. That is, we may establish a one-to-one correspondence between X_i and Y_j if cover and nucleus are equal (*i.e.*, $\widehat{\eta}(X, Y_j) = \eta(X, Y_j) = \{X_i\}$), indicating that the latter did not altered in between segmentations. For that, we simply name it as *stability*.

However, if the cover is $\widehat{\eta}(X, Y_j) = \{X_i\}$ but the nucleus is $\eta(X, Y_j) = \emptyset$, we say that Y_j was built by *shrinking* X_i. As one may note, since $Y_j \subseteq X_i$ the previous concepts are insufficient for determining the different possible cases when such an event occurs. Thus, we analyze the nucleus $\eta(Y, X_i)$ for distinction. For instance, we may define the dual of full merge, hereafter termed as *full split*, by verifying that $\cup\eta(Y, X_i) = X_i$. Simply put, X_i, a single region, was partitioned into regions of Y_j.

Similarly, *full deflation* is characterized by $\eta(Y, X_i) = \{Y_j\}$, indicating that X_i has eroded (or shrunk) into a single region of Y_j. Note that such a case is the dual of full inflation. For the merge and inflation case, we define its dual, *deflation and split*, by the inability of X_i to be built by the multiple regions of Y that compose its nucleus. Mathematically speaking, $\cup\eta(Y, X_i) \neq X_i$ for $|\eta(Y, X_i)| > 1$.

Still, in one specific case, the distinction between growth and shrinkage is not as distinguishable as in the previous cases. A region Y_j may require several elements of X to be built, meaning $|\widehat{\eta}(X, Y_j)| > 1$, but $\eta(X, Y_j) = \emptyset$, indicating that none is contained in the former. This case of *instability* poses a major challenge in accurately defining the sequence of the aforementioned operations, which resulted in Y_j. For instance, one could argue that this event may be termed as *translation* whenever growth and shrinkage are applied with equal magnitude. However, such simplification ignores the possible reshaping of the region in between segmentations. As a consequence, it encapsulates several distinct possible outcomes which, as we argue, do not assist in determining the hierarchiness of a multiscale segmentation.

We can exemplify each one of the aforementioned cases from the partitions presented in Figure 5.3 based on their nucleus η and cover $\widehat{\eta}$. For instance, while verifying whether B is an upscale of A, we can see that $\widehat{\eta}(A, B_1) = \{A_1, A_2, A_3, A_4\}$ and that $\eta(A, B_1) = \{A_1\}$, indicating that B_1 is built from full inflation. For B_2, it is possible to see that it was built from instability since $\eta(A, B_2) = \emptyset$ and $\widehat{\eta}(A, B_2) = \{A_2, A_4\}$. Like B_2, B_3 was also built from instability. In contrast, B_7 is equivalent to A_6 (*i.e.*, stable) since $\widehat{\eta}(A, B_7) = \eta(A, B_7) = \{A_6\}$. As an example, B_4 is a product of shrinking A_4 since $\eta(A, B_4) = \emptyset$

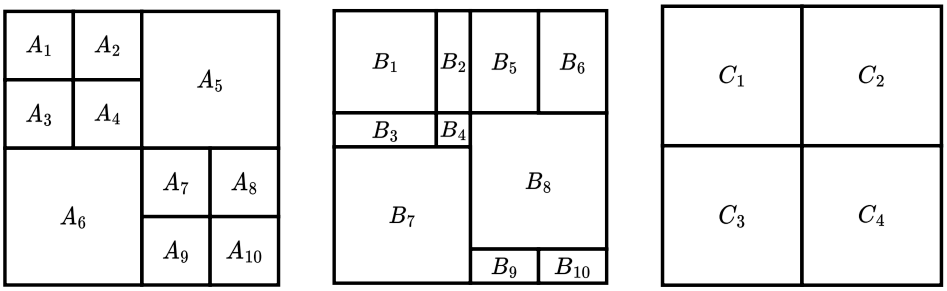


Figure 5.3 – (a-c) Three distinct partitions of a 8×8 square universe Υ.

and $\widehat{\eta}(A, B_4) = \{A_4\}$, requiring the verification that $\eta(B, A_4) = \{B_4\}$, indicating that B_4 was constructed by fully deflating A_4. One can verify the same case for B_9 and B_{10}. Since $\eta(A, B_5) = \emptyset$ and $\widehat{\eta}(A, B_5) = \{A_5\}$, we can see that B_5 (B_6, likewise) is built from deflation and split given that $\cup\eta(B, A_5) = \cup\{B_5, B_6\} \neq A_5$. Finally, we see that B_8 is built from merging and inflation as a result of $\cup\eta(A, B_8) = \cup\{A_7, A_8\} \neq B_8$.

In contrast, when analyzing from partition A to C, it is possible to see that $A \preceq C$ since every region of the latter is built either by stability or full merging — simply put, every region $A_i \in A$ is uniquely nested in some region $C_j \in C$. For instance, the case of full merge can be seen from C_1, which $\cup\eta(A, C_1) = \cup\{A_1, A_2, A_3, A_4\} = C_1$, and it is analogous for C_4. For C_2 and C_4, one can verify that $A_5 = C_2$ and $A_6 = C_4$, indicating stability. To illustrate the case of full split, we may analyze the converse of C_1 or C_4, that is, from partition C to A. As an example, based on $\eta(C, A_1) = \emptyset$ and $\widehat{\eta}(C, A_1) = \{C_1\}$, we infer that C_1 has shrunk, and from $\cup\eta(A, C_1) = C_1$, we conclude that a full split has occurred. Finally, since $A_5 = C_2$ and $A_6 = C_4$, and that $A_2, A_3, A_4, A_7, A_8, A_9, A_{10}$ are also built from full split, we may state that $C \succeq A$.

5.3 Measures

From the clear distinction between cases, we are then able to propose measures for evaluating two subsequent partitions $X, Y \in \Pi(\Upsilon)$ of Υ in a multiscale segmentation $\mathcal{M}$. First, we recur to a $r \times c$ matrix $\mathbf{M}(X, Y) = [\mathbf{M}(X, Y)_{ij}]$, considering $|X| = r$ and $|Y| = c$, that computes the intersection in between all regions of X and Y. More formally, for any two $X_i \in X$ and $Y_j \in Y$, we have that $\mathbf{M}(X, Y)_{ij} = |X_i \cap Y_j|$. It is clear that $\mathbf{M}(X, Y)_{ij} = \mathbf{M}(Y, X)_{ji}$ for any $X_i \in X$ and $Y_j \in Y$, implying that $\mathbf{M}(X, Y) = \mathbf{M}(Y, X)^\intercal$, where $\intercal$ denotes the *transpose* of the matrix.

From $\mathbf{M}$, we may normalize its values by $|\Upsilon|$, resulting in the matrix $\mathbf{M}_\Upsilon(X, Y)$ whose values $\mathbf{M}_\Upsilon(X, Y)_{ij} \in [0, 1]$ are defined by $\mathbf{M}_\Upsilon(X, Y)_{ij} = |X_i \cap Y_j|/|\Upsilon|$. Consequently, we

have the set of properties presented in Equation 5.3 as follows:

$$\begin{aligned} \sum_{i=1}^{r} \mathbf{M}(X,Y)_{ij} &= |Y_j| \\ \sum_{j=1}^{c} \mathbf{M}(X,Y)_{ij} &= |X_i| \\ \sum_{i=1}^{r}\sum_{j=1}^{c} \mathbf{M}(X,Y)_{ij} &= |\Upsilon| \\ \sum_{i=1}^{r}\sum_{j=1}^{c} \mathbf{M}_{\Upsilon}(X,Y)_{ij} &= 1 \end{aligned} \tag{5.3}$$

Likewise, we may normalize the matrix $\mathbf{M}(X,Y)$ by each element $X_i \in X$ for depicting its percentage within each intersection. Thus, the normalized matrix $\mathbf{M}_X(X,Y) = [\mathbf{M}_X(X,Y)_{ij}]$ defines its values $\mathbf{M}_X(X,Y)_{ij} \in [0,1]$ by $|X_i \cap Y_j|/|X_j|$. Consequently, not only $\sum_{j=1}^{c} \mathbf{M}_X(X,Y)_{ij} = 1$ but, also, $\mathbf{M}_X(X,Y)_{ij} = 1$ if, and only if, $X_i \subseteq Y_j$. As one may note, it is analogous to the normalized matrix $\mathbf{M}_Y(X,Y) = [\mathbf{M}_Y(X,Y)_{ij}]$ where $\mathbf{M}_Y(X,Y)_{ij} = |X_i \cap Y_j|/|Y_i| \in [0,1]$, which weights the intersection as a percentage of the region Y_j built from X. Thus, $\sum_{i=1}^{r} \mathbf{M}_Y(X,Y)_{ij} = 1$ and $\mathbf{M}_Y(X,Y)_{ij} = 1$ if, and only if, $Y_j \subseteq X_i$. Finally, it is clear that $X_i = Y_j$ when $\mathbf{M}_X(X,Y)_{ij} = \mathbf{M}_Y(X,Y)_{ij}$, and if X_i and Y_j are disjoint, for any of the previous matrices, its corresponding value will be zero.

For any two regions $X_i \in X$ and $Y_j \in Y$, we may quantify the property of $X_i \in \eta(X,Y_j)$ by a *nestedness* measure $\boldsymbol{NE}(X_i,Y_j) \in \{0,1\}$ defined by Equation 5.4

$$\boldsymbol{NE}(X_i,Y_j) = \mathbf{1}(X_i \subseteq Y_j) \equiv \mathbf{1}(\mathbf{M}_X(X,Y)_{ij} = 1) \tag{5.4}$$

in which $\mathbf{1}(\cdot)$ is *indicator function* that outputs "0" if the input boolean expression is *false*, or "1", otherwise. On the other hand, it is possible to measure the *inflation ratio* of Y_j through $\boldsymbol{IR}(X_i,Y_j) \in \{0,1\}$ by evaluating whether $X_i \in \widehat{\eta}(X,Y_j)$ but $X_i \notin \eta(X,Y_j)$. Equation 5.5 formally defines such measure as:

$$\begin{aligned} \boldsymbol{IR}(X_i,Y_j) &= \mathbf{1}(X_i \in \widehat{\eta}(X,Y_j) \setminus \eta(X,Y_j)) \\ &\equiv \mathbf{1}((0 < \mathbf{M}_X(X,Y)_{ij} < 1) \ \wedge \ (0 < \mathbf{M}_Y(X,Y)_{ij} < 1)) \end{aligned} \tag{5.5}$$

considering $\wedge$ the logical operator AND.

Inspired from [60], we propose a measure for computing the minimum proportional "effort" for altering X_i to either equate to Y_j or be disjoint from it. Thus, by computing the lowest quantity of operations (*i.e.*, adding or removing elements) necessary, the *refinement error* $\mathbf{RE}(X_i,Y_j) \in [0,1]$ can be defined by Equation 5.6:

$$\boldsymbol{RE}(X_i,Y_j) = \min\left\{|X_i \cap Y_j|, |X_i \setminus Y_j|\right\}/|X_i| \tag{5.6}$$

in which we may compute through Equation 5.7:

$$\begin{aligned}\boldsymbol{RE}(X_i, Y_j) &= \min\left\{|X_i \cap Y_j|, |X_i \setminus Y_j|\right\} / |X_i| \\ &= \min\left\{|X_i \cap Y_j|/|X_i|, (|X_i| - |X_i \cap Y_j|)/|X_i|\right\} \\ &\equiv \min\left\{\mathbf{M}_X(X,Y)_{ij}, 1 - \mathbf{M}_X(X,Y)_{ij}\right\}\end{aligned} \tag{5.7}$$

Still, if $\mathbf{M}_X(X,Y)_{ij} > 0$, we may impose a harsher penalization by adapting the latter equation to Equation 5.8:

$$\boldsymbol{RE}(X_i, Y_j) = \min\left\{1, (1 - \mathbf{M}_X(X,Y)_{ij})/\mathbf{M}_X(X,Y)_{ij}\right\} \tag{5.8}$$

As one may note, all previous measures evaluate in between regions $Y_j \in Y$ and $X_i \in X$. However, if one desires to apply each measure for Y_j but considering all regions in X, it is possible to weight the influence of each particular result of X_i by its intersection $|X_i \cap Y_j|$ with Y_j. Equation 5.9 presents such possibility of obtaining normalized measures:

$$\begin{aligned}\boldsymbol{NE}(X, Y_j) &= \sum_{i=1}^{r} \boldsymbol{NE}(X_i, Y_j)(|X_i \cap Y_j|/|Y_j|) \equiv \sum_{i=1}^{r} \boldsymbol{NE}(X_i, Y_j)\mathbf{M}_Y(X,Y)_{ij} \\ \boldsymbol{IR}(X, Y_j) &= \sum_{i=1}^{r} \boldsymbol{IR}(X_i, Y_j)(|X_i \cap Y_j|/|Y_j|) \equiv \sum_{i=1}^{r} \boldsymbol{IR}(X_i, Y_j)\mathbf{M}_Y(X,Y)_{ij} \\ \boldsymbol{RE}(X, Y_j) &= \sum_{i=1}^{r} \boldsymbol{RE}(X_i, Y_j)(|X_i \cap Y_j|/|Y_j|) \equiv \sum_{i=1}^{r} \boldsymbol{RE}(X_i, Y_j)\mathbf{M}_Y(X,Y)_{ij}\end{aligned} \tag{5.9}$$

Similarly, it is possible to perform an analogous adaptation for evaluating both partitions $X, Y \in \Pi(\Upsilon)$ of Υ elements as shown in Equation 5.10:

$$\begin{aligned}\boldsymbol{NE}(X, Y) &= \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{NE}(X_i, Y_j)(|X_i \cap Y_j|/|\Upsilon|) \equiv \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{NE}(X_i, Y_j)\mathbf{M}_\Upsilon(X,Y)_{ij} \\ \boldsymbol{IR}(X, Y) &= \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{IR}(X_i, Y_j)(|X_i \cap Y_j|/|\Upsilon|) \equiv \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{IR}(X_i, Y_j)\mathbf{M}_\Upsilon(X,Y)_{ij} \\ \boldsymbol{RE}(X, Y) &= \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{RE}(X_i, Y_j)(|X_i \cap Y_j|/|\Upsilon|) \equiv \sum_{i=1}^{r}\sum_{j=1}^{c} \boldsymbol{RE}(X_i, Y_j)\mathbf{M}_\Upsilon(X,Y)_{ij}\end{aligned} \tag{5.10}$$

When evaluating $X, Y \in \mathcal{M}$, if $X \preceq Y$ in the multiscale $\mathcal{M}$, one can see that every $X_i \in X$ is uniquely nested in some region $Y_j \in Y$ and, consequently, Y_j can be built solely from its own nucleus $\eta(X, Y_j)$. Therefore, $\boldsymbol{NE}(X,Y) = 1$ and $\boldsymbol{IR}(X,Y) = 0$, since every region in the cover is also in the nucleus. Furthermore, $\boldsymbol{RE}(X,Y) = 0$, since there is no multiple region overlap due to nesting. On the other hand, when $X \succeq Y$, $\boldsymbol{NE}(X,Y) = 0$ but $\boldsymbol{NE}(Y,X) = 1$, and $\boldsymbol{RE}(X,Y) = \boldsymbol{RE}(Y,X) = 0$ since such error is resultant rather from causality than from locality violations. Likewise, $\boldsymbol{IR}(X,Y) = \boldsymbol{IR}(Y,X) = 0$.

In the case of $\mathcal{M}$ is not a hierarchy, it is possible to face different situations. If the results present moderate to high values of $\boldsymbol{NE}$ but high values of $\boldsymbol{IR}$ and $\boldsymbol{RE}$, one can infer that the nesting behavior is related to small but numerous regions, whereas the errors are

related to particular bigger regions. On the other hand, for high values of ***NE*** and small values of ***IR*** and ***RE***, one should expect an "quasi-hierarchy" since violations are minimal and nesting is high. Likewise, for small values of ***IR***, ***RE*** and ***NE***, the former indicates that it can also be perceived as a "quasi-hierarchy" since locality violations are not severe, possibly being the reason for such significant small performance of ***NE***. Finally, for high values of ***IR***, ***RE***, and low values of ***NE***, it is clear to see that such a condition exhibits the worst-case scenario: high locality errors caused by inflation and reduced nesting behavior. Thus, as one may infer, correcting such partitions would be cumbersome and, most likely, unfruitful.

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}
A_1	4	0	0	0	0	0	0	0	0	0
A_2	2	2	0	0	0	0	0	0	0	0
A_3	2	0	2	0	0	0	0	0	0	0
A_4	1	1	1	1	0	0	0	0	0	0
A_5	0	0	0	0	6	6	0	4	0	0
A_6	0	0	0	0	0	0	16	0	0	0
A_7	0	0	0	0	0	0	0	4	0	0
A_8	0	0	0	0	0	0	0	4	0	0
A_9	0	0	0	0	0	0	0	2	2	0
A_{10}	0	0	0	0	0	0	0	2	0	2

(a)

	C_1	C_2	C_3	C_4
A_1	4	0	0	0
A_2	4	0	0	0
A_3	4	0	0	0
A_4	4	0	0	0
A_5	0	16	0	0
A_6	0	0	16	0
A_7	0	0	0	4
A_8	0	0	0	4
A_9	0	0	0	4
A_{10}	0	0	0	4

(b)

Table 5.1 – (a-b) Non-normalized matrices based from the partitions in Figure 5.3.

With the aid of Table 5.1, we may compute our proposed measures with respect to the partitions presented in Figure 5.3. Considering B_8 which is built from merge and inflation, we have that $\boldsymbol{NE}(A, B_8) = 8/16 = 0.5$, $\boldsymbol{IR}(A, B_8) = 8/16 = 0.5$, and $\boldsymbol{RE}(A, B_8) = 8/16 = 0.5$, indicating that 50% of B_8 was built from inflation and the remaining 50% from merging. As one may note, the refinement error result is equal to the inflation ratio since, as defined, inflation implies on multiple region overlap (thus, errors to be refined). As an example of full inflation, we may consider B_1, whose $\boldsymbol{NE}(A, B_1) = 4/9 \approx 0.44$, $\boldsymbol{IR}(A, B_1) = 5/9 \approx 0.56$, and $\boldsymbol{RE}(A, B_1) = 5/9 \approx 0.56$. In contrast, for C_1, which is built solely from full merge, $\boldsymbol{NE}(A, C_1) = 16/16 = 1$, $\boldsymbol{IR}(A, C_1) = 0/16 = 0$, and $\boldsymbol{RE}(A, C_1) = 0/16 = 0$, as expected. Finally, when evaluating A to B, we have that $\boldsymbol{NE}(A, B) = 28/64 = 0.4375$, $\boldsymbol{IR}(A, B) = 36/64 = 0.5625$, and $\boldsymbol{RE}(A, B) = 36/64 = 0.5625$. From A to C, given $A \preceq C$, we obtain the following results: $\boldsymbol{NE}(A, C) = 64/64 = 1$, $\boldsymbol{IR}(A, C) = 0/64 = 0$, and $\boldsymbol{RE}(A, C) = 0/64 = 0$.

For two segmentations $X, Y \in \mathcal{M}$ of a given image containing $|\Upsilon|$ pixels, the effort to compute the matrix $\mathbf{M}(X, Y)$ is $\mathrm{O}(|\Upsilon|^2)$. One can see that obtaining the other matrices (*i.e.*, $\mathbf{M}_Y(X, Y)$, $\mathbf{M}_X(X, Y)$, and $\mathbf{M}_\Upsilon(X, Y)$) requires $\mathrm{O}(|X||Y|)$ for normalizing each value $\mathbf{M}(X, Y)_{ij}$ of the former matrix. Calculating any measure between regions $Y_j \in Y$ and $X_i \in X$ is constant when $\mathbf{M}(X, Y)$ is provided. On the other hand, when, evaluating between Y_j and every $X_i \in X$, the complexity becomes $\mathrm{O}(|X|)$. Finally, when evaluating two segmentations X and Y, we have a complexity of $\mathrm{O}(|X||Y|)$ irrespective of the measure considered.

Chapter 6

Experimental Results

In order to evaluate the performance of our approach and, consequently, validade our research questions, benchmarking is crucial. The analysis we perform in this chapter is backed by the data we have collected throughout our study and assisted by the visual examples which we provide. The main motivation is to define a setup that enables us to conclude, that SICLE and the hierarchiness measures are able to produce the expected outcome.

Thus, this chapter is organized as follows. Section 6.1 presents the datasets and the motivation behind their selection, aside from detailing the split for optimization and the saliency estimator considered. Subsequently, in Section 6.2, we present the baselines for both SICLE and hierarchiness, detailing, again, our criteria for selection. In Section 6.3, we formalize the popular metrics for superpixel segmentation. Based on this setup, we, in Section 6.4, analyze the performance of SICLE whenever an object saliency map is absent, starting from an ablation study up to comparing with our baselines. Similarly, Section 6.5 is organized equivalently, but considering the existence of saliency maps. Finally, we evaluate the behavior of our hierarchiness measures in the last Section 6.6.

6.1 Datasets

Mainly for evaluating SICLE, our goal is to provide a wide-ranging performance analysis in generating superpixels. For that, we argue that a set of significantly distinct datasets is necessary and, therefore, we chose five datasets, three of them being natural image datasets, but with different objects (and goals in delineation). The last two are medical image datasets, one aims delineating an organ, whereas the other involves parasitology.

The first natural image dataset is the popular contour-driven *Berkeley Segmentation Dataset* (BSDS500) [6], composed of 500 images. Thus, the goal is to segment all relevant edges, defined by a set of users, within the image. However, such ground truth is not applicable whenever a single object is desired and, consequently, is not a viable option for our object-based analysis. To cope with this drawback, we chose the well-known *Extended Complex Scene Saliency* (ECSSD) [68] dataset, which contains 1000 natural images with complex objects and backgrounds, being the major reason for selection. Differently from

BSDS500, the ground truth in ECSSD for any image defines the most salient objects rather than the most relevant contours. The third dataset is the *Insects* [58] dataset, which contains 130 images of various insects and bugs. The motivation behind Insects is due to the thin legs of the objects, which are challenging to delineate accurately.

Concerning the two medical datasets, one of them is *Liver* [75], a dataset containing 40 images of computerized tomography slices of the human liver. Delineating such an object is a difficult task since it is monochromatic, meaning that color distinction is reduced to shades of gray, and its borders are smooth, requiring a more sofisticated detection rather than mere contrast. In contrast, *Parasites* [16] is a dataset containing 72 colored images of intestinal parasite eggs (*i.e.*, helmynth). Yet, although colored, segmenting parasite eggs demands expertise since impurities, which may be attached, resemble the object and, furthermore, the borders are also smooth, but due to the aqueous solution for microscopy analysis.

To evaluate SICLE, we have considered all five datasets. For each one, we randomly selected 30% for optimization and 70% for testing, except for BSDS500, which already establishes such a division. For producing the saliency maps, we opted for the U^2-Net estimator [64], fine-tuned using the same training set and its default parameters. On the other hand, to evaluate the hierarchiness measures, we selected the full Liver and Parasites datasets. The motivation lies in both characteristics of the datasets: the objects are difficult to delineate, thus more difficult to sustain borders when the approach is not hierarchical, and both objects and backgrounds do not change in their own characteristics (*i.e.* stable). The latter characteristic may favor providing similar, if equal, partitioning within different scales, possibly improving the hierarchiness of the multiscale segmentation. Although one may argue in favor of using natural image datasets over Parasites and Liver, we argue that our analysis could be prejudiced with such a suggestion since backgrounds and objects themselves vary significantly within the same dataset.Thus, it is significantly more challenging to comprehend the nesting behavior in such a setup.

6.2 Methods

From our analysis, we aim to provide off-the-shelf SICLE variants for effective delineation with irregular or compact superpixels, hereafter named SICLE-IRREG and SICLE-COMP, respectively, which will be thoroughly optimized and defined in the ablation study. Moreover, in order to improve speed, we opted for computing *differential IFTs* (DIFTs) [34], which, in simple terms, focus on competing for non-conquered vertices and correcting inconsistencies for changing the seed set S. Thus, since SICLE only removes seeds, those that were maintained tend to produce similar, if not equal, superpixels. For those that were removed, DIFT will assign to some "surviving" seed and deal with eventual optimum-path inconsistencies on the superpixel's subtree. Such an approach is significantly faster than performing several sequential IFTs, especially when a multiscale segmentation with many scales is requested. Finally, our code is publicly available

online [1] [2].

The baselines for comparing SICLE in terms of superpixel segmentation were selected based on their novelty, efficiency or efficacy: (i) ERS [54] [3]; (ii) SLIC [1] [4]; (iii) SH [80] [5]; (iv) IBIS [25] [6]; (v) LSC [52] [7]; (vi) GMM [8] [8]; (vii) OISF [16, 17] [9]; and (viii) DAL-HERS [62] [10]. For all the aforementioned methods, we used their default parameter setting, and we evaluated considering $N_f \in \{50, 100, 250, 500, 750, 1000\}$. Concerning compact superpixels, SLIC, IBIS, LSC, and GMM are top approaches in the literature, whereas SH and ERS are highly irregular but with effective delineation. In terms of speed, one may refer to SLIC and IBIS as the fastest methods, while OISF, although one of the slowest methods, is the single object-based method amongst baselines. In order to compare with recent deep-learning approaches in superpixel segmentation, DAL-HERS is an appropriate choice.

We select two methods for establishing an upper and lower bound for the hierarchiness measures proposed. For the former, we chose SH, since it is a hierarchical superpixel segmentation. On the other hand, for the latter, we provide a method that minimizes nestedness as much as possible. Thus, we implemented a *(Random) Voronoi* superpixel segmentation: it generates N_f random seeds and computes a Voronoi tesselation for obtaining a segmentation with N_f superpixels. Finally, we recur to SICLE-IRREG and SICLE-COMP as case-study since SICLE mimics hierarchical segmentations by not altering the seed displacement. We emphasize that, in this set of experiments, no saliency map was used. For all methods, we demanded $N_f = \{750, 500, 250, 100\}$ for analysis.

Although one may suggest including other methods as baselines, such as SLIC or the popular *Felzenszwalb-Huttenlocher* method (FH) [36], in our analysis, we argue that such inclusion is not beneficial. First, SLIC alters the seed displacement as N_f changes, prejudicing nestedness by design. As one may note, such a condition is covered by Voronoi. Second, as pointed out by [43], FH may not guarantee the causality principle for different values of k. Clearly, by avoiding causality violations, we may focus on analyzing the locality violations, which are the goals of our proposals. Therefore, in contrast to both exemplified, using SICLE provides a straightforward, simple, and interesting case for our analysis. Thus, in our opinion, all four methods offer a broad and thorough perception of our proposed measures.

1. https://github.com/LIDS-UNICAMP
2. https://github.com/IMScience-PPGINF-PucMinas
3. https://github.com/mingyuliutw/EntropyRateSuperpixel
4. https://www.epfl.ch/labs/ivrl/research/slic-superpixels/
5. https://github.com/semiquark1/boruvka-superpixel
6. https://github.com/xapha/IBIS
7. https://jschenthu.weebly.com/projects.html
8. https://github.com/ahban/GMMSP-superpixel
9. https://github.com/LIDS-UNICAMP/OISF
10. https://github.com/hankuipeng/DAL-HERS

6.3 Evaluation

Concerning superpixel segmentation, we measured the performance of all methods, using popular metrics, in a 64-bit Intel(R) Core(TM) i7-4790S PC with a CPU Speed of 3.20GHz. First, let $\mathcal{G} = \{G_1, \ldots, G_k\}$ be the set of objects defined in the ground truth. Clearly, $\mathcal{G} \in \Pi(V)$. Then, let $\varpi^r(\mathcal{G}) \subset V$ be defined as follows (Equation 6.1):

$$\varpi^r(\mathcal{G}) = \{p \in V : p \in G_i \in \mathcal{G} \text{ and } \exists\ q \in \boldsymbol{A}^r(p) \text{ such that } q \notin G_i\} \tag{6.1}$$

Simply put, ϖ^r extracts the borders of the regions with respect to a radius r, by verifying, for every vertex, if it contains an adjacent vertex that does not belong to the same group.

Then, we may intuitively define the *Boundary Recall* (BR) [71] as the ratio of superpixel borders overlapping object boundaries, with respect to a radius r. For that, let $\mathcal{S} = \{S_1, \ldots, S_n\} \in \Pi(V)$ be the superpixel segmentation containing n superpixels. Mathematically speaking, we may define $\boldsymbol{BR}(\mathcal{G}, \mathcal{S}) \in [0, 1]$ as (Equation 6.2):

$$\boldsymbol{BR}(\mathcal{G}, \mathcal{S}) = \frac{1}{|V|} \sum_{p \in V} \mathbf{1}(p \in \varpi^r(\mathcal{G}) \text{ and } p \in \varpi^r(\mathcal{S})) \tag{6.2}$$

where $\mathbf{1}$ is an *indicator function* that outputs "0" if the input boolean expression is false, and "1", otherwise. Similarly to [71], we set $r = 0.0025d\sqrt{2}$, being d the image's diagonal size. For this metric, higher is better.

For evaluating "leakings" in delineation, we recur to the *Under-segmentation Error* (UE) [60], which measures the error (Equation 6.3) from multiple object overlap by the superpixels:

$$\boldsymbol{UE}(\mathcal{G}, \mathcal{S}) = \frac{1}{|V|} \sum_{G_i \in \mathcal{G}} \sum_{S_j \in \mathcal{S}} \min\{|S_j \cap G_i|, |S_j \setminus G_i| \tag{6.3}$$

given $S_j \cap G_i \neq \emptyset$. Unlike $\boldsymbol{BR}$, for $\boldsymbol{UE}$, lower is better.

In terms of evaluating compactness, we use the popular *Compactness* (CO) [71] one. Differently from both $\boldsymbol{UE}$ and $\boldsymbol{BR}$, $\boldsymbol{CO}$ defines compactness by evaluating whether the superpixels, weighted by their size, present a compact shape, being maximum when it is a circle. Therefore, not only is it not a ground-truth-based metric, but it also aims for higher values. Formally, it is defined by Equation 6.4:

$$\boldsymbol{CO}(\mathcal{S}) = \frac{1}{|V|} \sum_{S_j \in \mathcal{S}} \frac{4\pi |S_j|}{|\varpi^{\sqrt{2}}(\mathcal{S}) \cap S_j|^2} \tag{6.4}$$

Note that the divider is an approximation for the square of the perimeter of the superpixel S_j.

6.4 Non-object-based Analysis

Since building and evaluating all possible SICLE variations is burdensome, we selected default parameters based on previous works as a starting point for optimization. Thus, for SICLE-IRREG, we chose $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}^{\alpha}_{\text{root}}$ functions [14, 18, 26], while for SICLE-COMP, we

have considered $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}^{\alpha}_{\text{sum}}$ [75]. For both, we set the maximum number of iterations $\Omega = 10$, as performed in several works [1, 37, 52, 75], and we empirically set the initial seed quantity $N_0 = 8000$ for the RND sampling strategy. Clearly, in this case, we set $\alpha = 0$.

6.4.1 Ablation Study

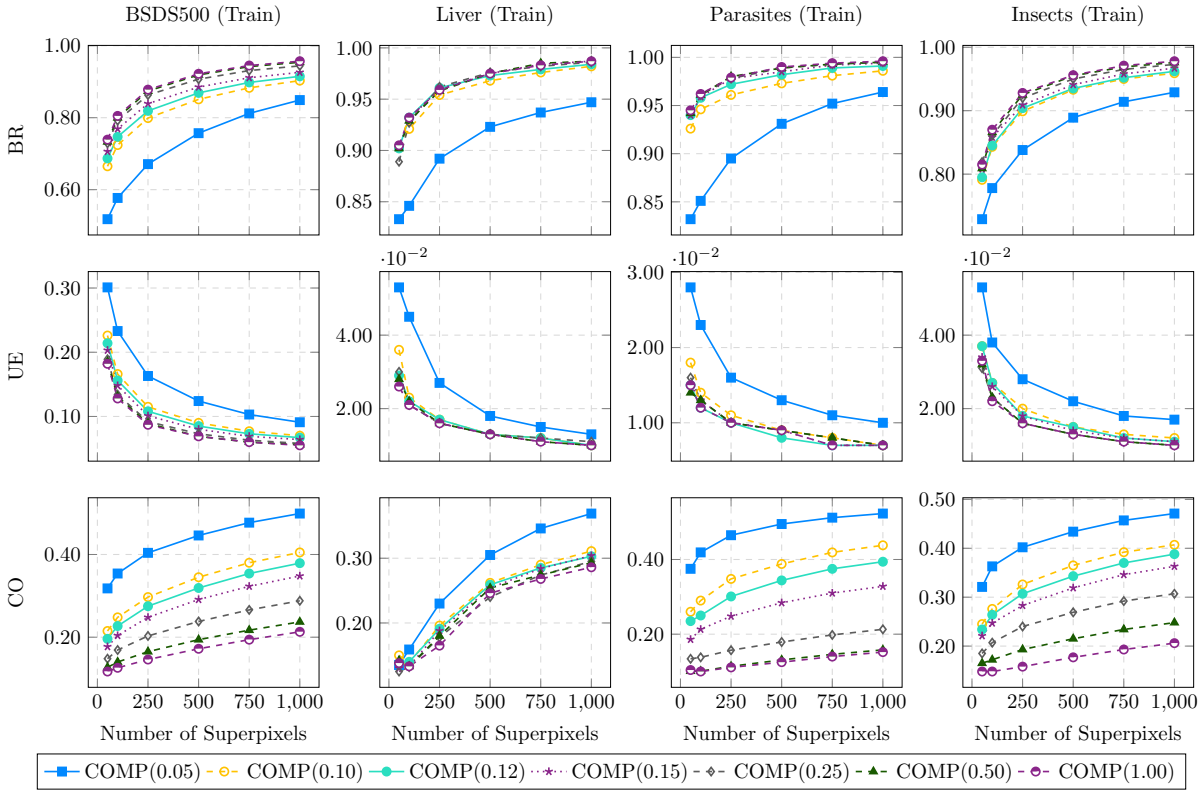


Figure 6.1 – Quantitative results on the train set of each dataset for SICLE-COMP for varying ι.

Differently from SICLE-IRREG, SICLE-COMP possesses two parameters: ι, the irregularity factor, and β, the boundary adherence factor. As in [16, 37, 75], we set $\beta = 12$ since we always demand superpixels adherent to object boundaries. However, we optimized $\iota \in \{0.05, 0.10, 0.12, 0.15, 0.20, 0.25, 0.50, 1.00\}$ for selecting such value which offers effective delineation and fair compacity (Figure 6.1). Clearly, for $\iota = 0.05$, SICLE-COMP achieves top compacity at the expense of worst boundary adherence. It is interesting to note that a slight variation from 0.05 to 0.10 severely impacts the performance of our compact variant, leading to less compactness but increased superpixel delineation. But, as ι grows, the deterioration of $\boldsymbol{CO}$ is more significant than the subtle increase in $\boldsymbol{BR}$. Due to this reason, we opt for $\iota = 0.12$, as in [75], since it still improves its delineation in Parasites and suffers a tolerable impact on compactness (as defined by our optimization goal).

Figure 6.2 shows our experiment for determining the best seed relevance criterion

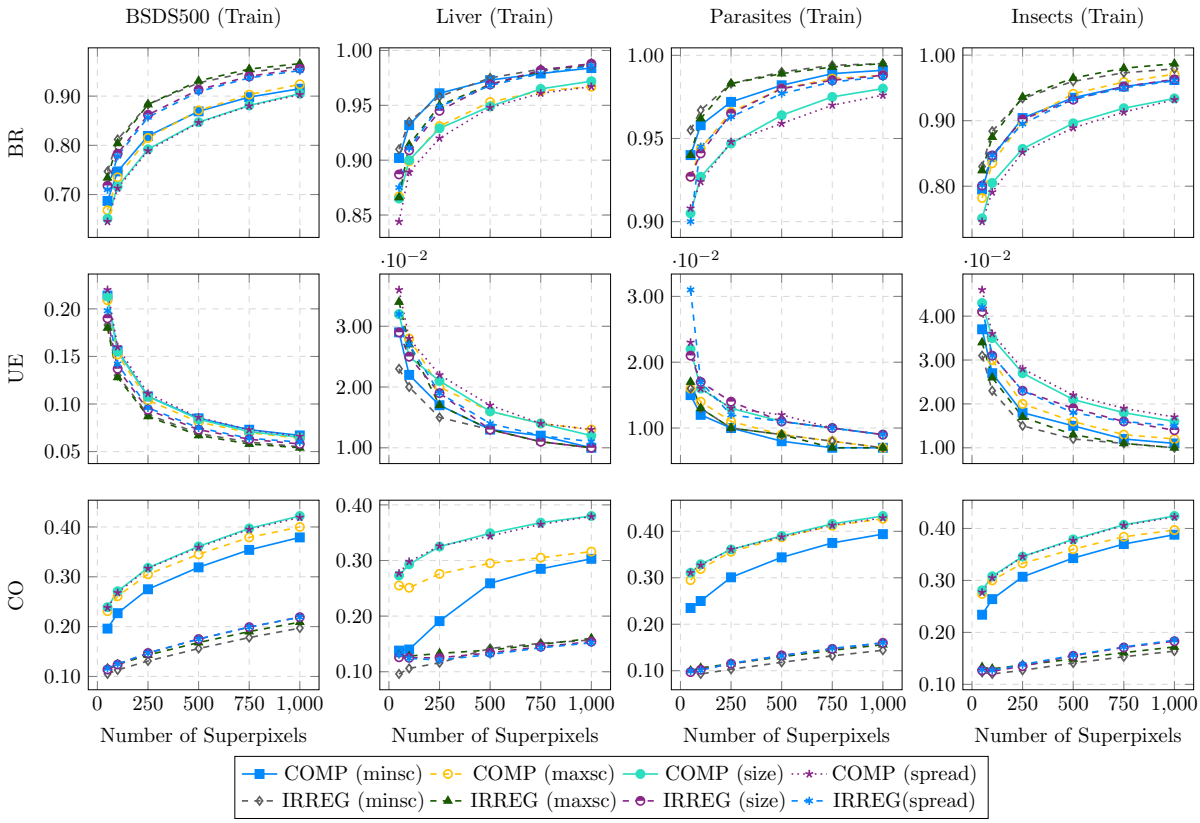


Figure 6.2 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying $\boldsymbol{V}_*$.

$\boldsymbol{V}_* \in \{\boldsymbol{V}_{\text{size}}, \boldsymbol{V}_{\text{minsc}}, \boldsymbol{V}_{\text{maxsc}}, \boldsymbol{V}_{\text{spread}}\}$ for SICLE-IRREG and SICLE-COMP. First, analyzing SICLE-IRREG, it is interesting to notice that size and spreading superpixels are fair options for delineation with irregular superpixels. Note that both tend to promote a slightly increased overall compactness. But, it is clear that contrast significantly assists in delineating the object. We argue that, since SICLE-IRREG maximizes delineation, it is expected that both $\boldsymbol{V}_{\text{minsc}}$ and $\boldsymbol{V}_{\text{maxsc}}$ present similar behavior, given that both select seeds surrounded by one or more dissimilar neighbors. However, in Liver, one may note that the latter shows a worse performance than the former. We claim that the contrast between the torso and background is sharp, leading to $\boldsymbol{V}_{\text{maxsc}}$ favor seeds near the boundaries of the torso, while competition internally (where the object is located) is significantly deteriorated by the lack of representatives. For such reason, we opt for $\boldsymbol{V}_{\text{minsc}}$ for SICLE-IRREG.

We observe a different behavior for SICLE-COMP. For $\boldsymbol{V}_{\text{size}}$ and $\boldsymbol{V}_{\text{spread}}$, SICLE-COMP presents top compactness but with fair delineation. Again, contrast is crucial for improving performance, but since SICLE-IRREG maximizes delineation, SICLE-COMP must balance between the latter and compactness. Thus, by analyzing $\boldsymbol{V}_{\text{minsc}}$, one can see that such criterion is well-suited for all datasets, especially Liver, but it impacts $\boldsymbol{CO}$ negatively. We argue that positioning seeds in heterogeneous regions avoids and prevents leaking, as seen for SICLE-IRREG. But, for SICLE-COMP, placing in such regions will generate irregular superpixels if ι is not sufficiently low. In contrast, $\boldsymbol{V}_{\text{maxsc}}$

not only achieves similar effective performance throughout datasets but minimally impacts compactness. We claim that, by positioning seeds near sharp borders, such contour is first delineated by color dissimilarity, and leakings are prevented and diminished by the shape regularity. Therefore, we chose $\boldsymbol{V}_{\text{maxsc}}$ for SICLE-COMP.

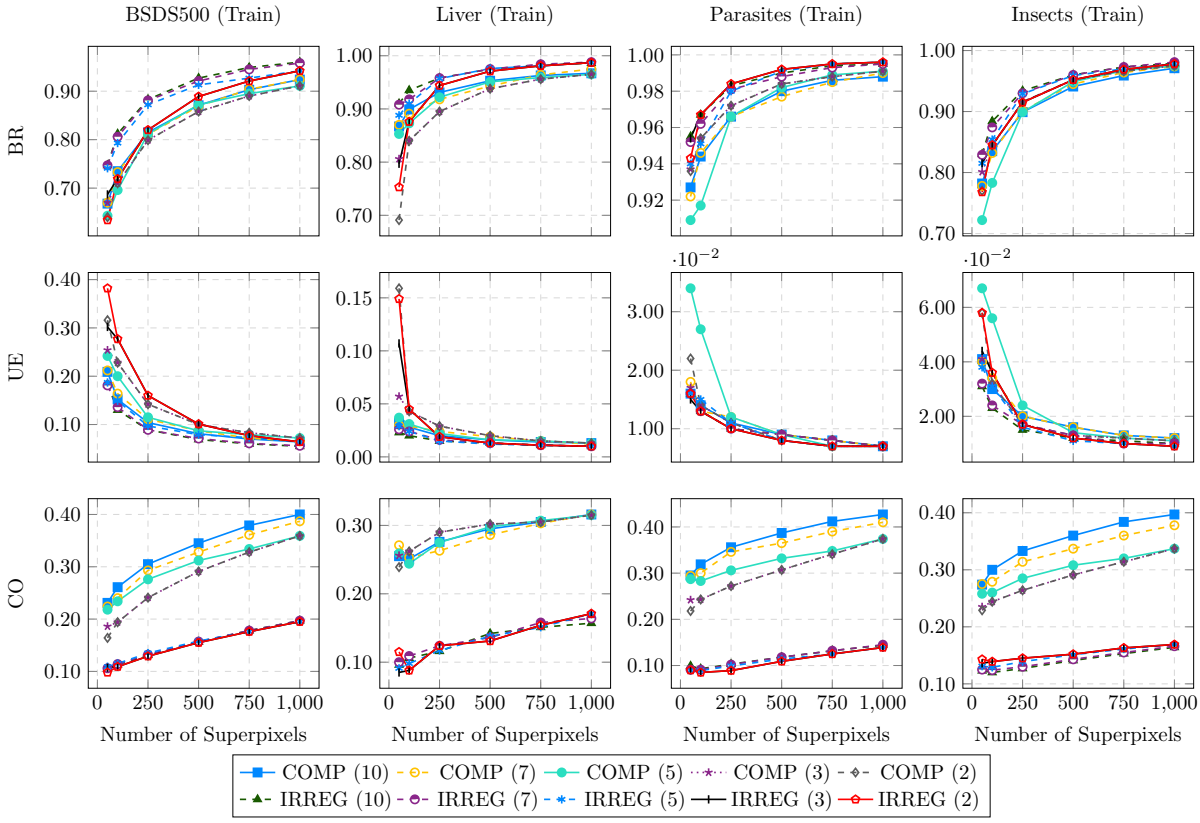


Figure 6.3 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying Ω.

If a seed relevance criterion is accurate, SICLE variants would demand fewer iterations to achieve top delineation performance. Thus, Figure 6.3 presents the experiment performed for optimizing $\Omega \in \{10, 7, 5, 3, 2\}$. Unsurprisingly, for SICLE-IRREG, requiring three iterations or less deteriorates segmentation significantly since $\boldsymbol{V}_{\text{minsc}}$ is effective but not ideal. On the other hand, due to its accuracy, it is clear that increasing Ω over five iterations does not lead to significant improvements. In fact, increasing it only leads to higher execution time. Consequently, we set SICLE-IRREG to perform, at most, $\Omega = 5$ iterations.

Likewise, we set $\Omega = 7$ for SICLE-COMP for similar reasons. For $\Omega \leq 3$, one can see that $\boldsymbol{CO}$ is decreased with a slight impact in $\boldsymbol{BR}$ and $\boldsymbol{UE}$. For $\Omega = 5$, SICLE-COMP improves its performance, but still is worst for $\Omega \geq 7$, especially in Parasites. Since $\Omega = 10$ offers little to no improvement over $\Omega = 7$, we chose the latter. We argue that such difference to SICLE-IRREG is that $\boldsymbol{V}_{\text{maxsc}}$ requires more iterations for accurate selection since SICLE-COMP imposes shape constraints, while $\boldsymbol{V}_{\text{minsc}}$ can perform the same task faster, maximizing only delineation (and irregularity).

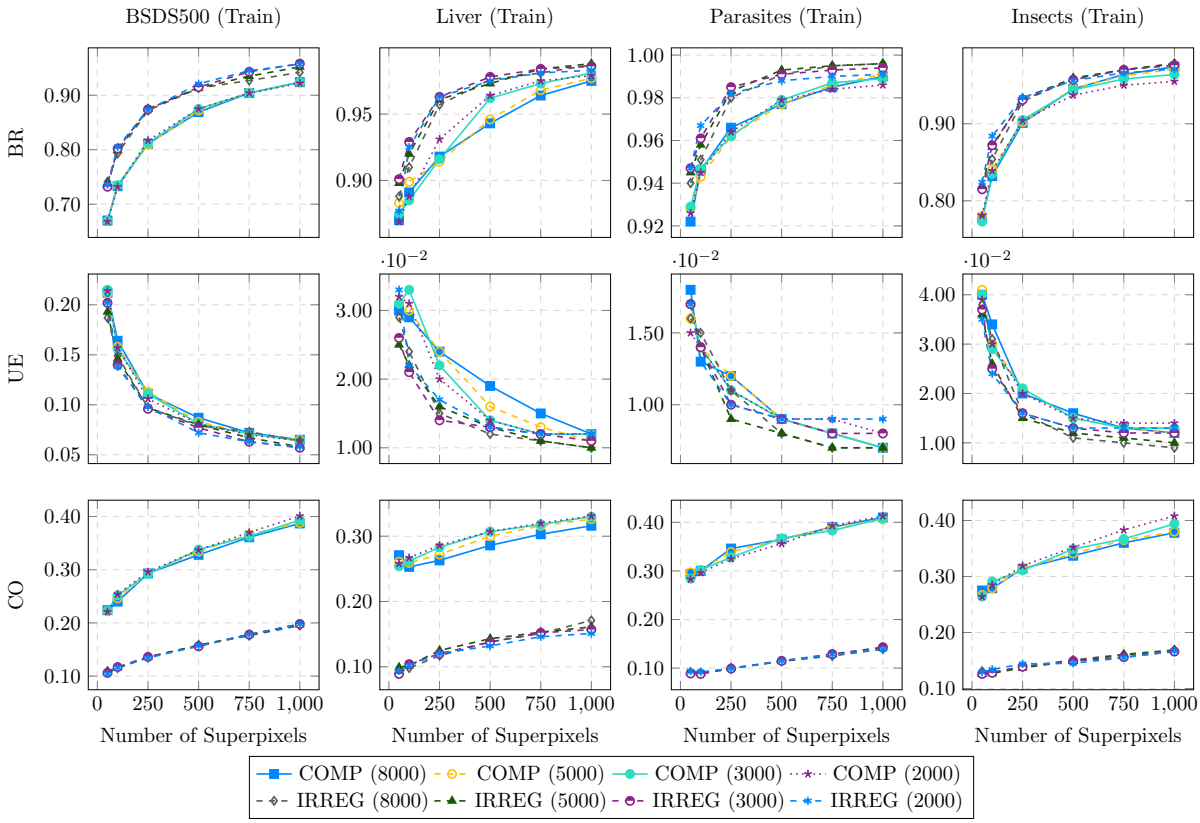


Figure 6.4 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying N_0.

In Figure 6.4, we evaluated two conditions. First, if oversampling does contain all relevant seeds, increasing $N_0 \in \{8000, 5000, 3000, 2000\}$ would not lead to significant improvements since it would only add either redundantly relevant seeds or irrelevant ones. Second, if $\boldsymbol{V}_{\text{minsc}}$ and $\boldsymbol{V}_{\text{maxsc}}$ are accurate, they would not require a big seed set S for selection. As one may note, both inferences seem true since $N_0 = 3000$ seeds are sufficient for effective delineation for both SICLE-IRREG and SICLE-COMP. Although one could argue in favor of selecting 2000 initial seeds in oversampling, we say that seeing the performance in Parasites, larger images tend to require more seeds to be sampled. Clearly, $N_0 \geq 5000$ either deteriorates delineation for SICLE-COMP, as seen in Liver, or does not provide any significant improvement, as seen for SICLE-IRREG in all datasets. Thus, setting $N_0 = 3000$ still reduces the amount of seeds to be sampled while maintaining its delineation performance.

6.4.2 Quantitative and Qualitative Results

The quantitative results of all methods are shown in Figure 6.5 against our optimized SICLE variants. It is possible to see that SICLE-IRREG achieves superior delineation in all datasets for ***BR***, except for Insects. We argue that since SICLE is a seed-based method, the positioning of seeds is obviously crucial. However, even though oversampling selects seeds in regions near or at the bugs' legs, sustaining such delineation is challeng-

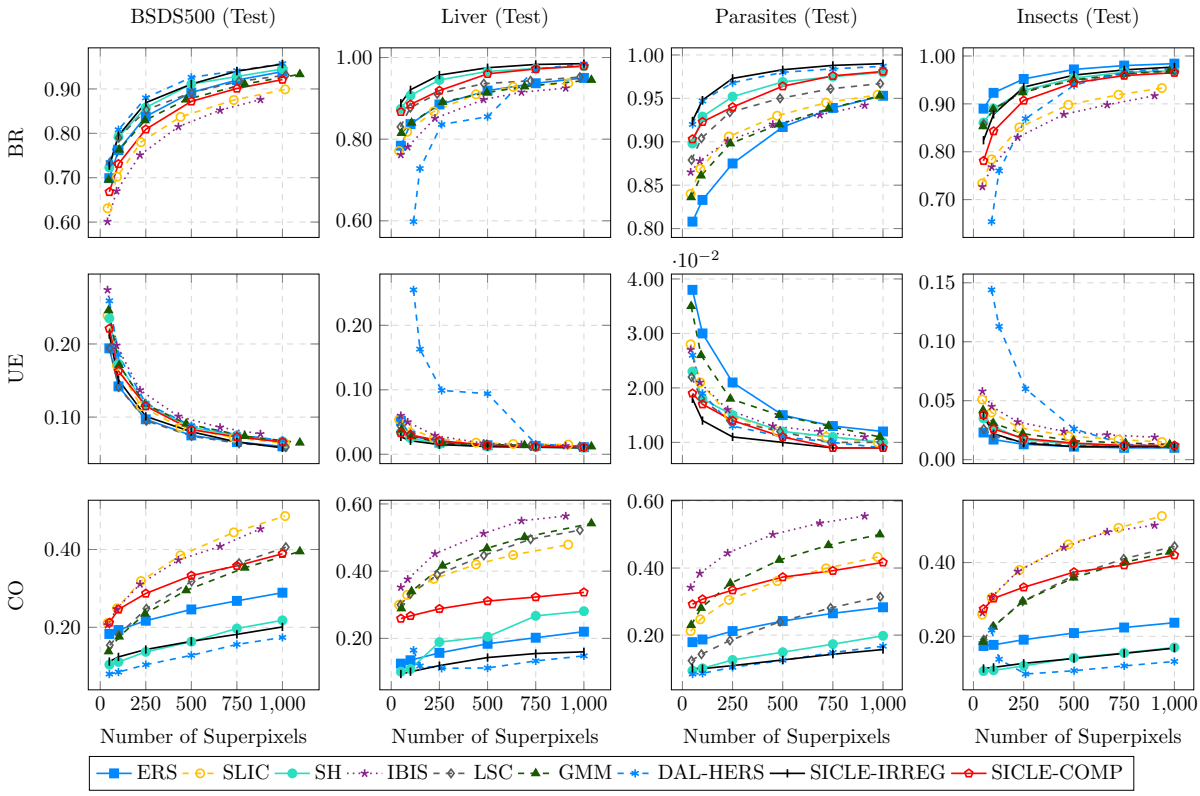


Figure 6.5 – Quantitative results on the test set of each dataset when no object information is provided.

ing, especially due to their thin size. As an example, ERS, which excels in Insects, is a seedless graph-based approach, but it is important to note that ERS is fair for both medical image datasets. Although SH falls in the same category, the grouping of regions presents the same drawback of sustaining a thin region throughout the algorithm. When verifying DAL-HERS, it is interesting to notice that, although it achieves on-par performance with our approach in BSDS500 and Parasites, it fails for Liver and Insects, which share similarities with BSDS500 by being also a natural image dataset. In fact, apart from SICLE-IRREG and SH, the remaining irregular methods do not present a consistent performance throughout the baselines. We argue that such behavior indicates that the latter approaches are best suited for specific conditions rather than a general agnostic solution in superpixel segmentation for any domain. Concerning ***UE***, we can again note that SICLE-IRREG is continuously on par with the best methods on each dataset, specially for Parasites.

We can also notice such effective performance and consistency in SICLE-COMP for all metrics considered in all datasets. For ***BR***, our approach achieves on-par performance with GMM and LSC, the most effective methods for generating compact superpixels, while surpassing both by a significant margin in Liver and Parasites. Likewise, SICLE-COMP presents values for ***UE*** on par with the best methods on each dataset. One may perceive stability in generating compact superpixels by verifying that SICLE-COMP consistently achieves similar absolute performance ***CO***, irrespective of the dataset. Amongst the best

baselines in compactness, only IBIS shows similar behavior. Other approaches, such as SLIC, GMM, and LSC, are unable to exhibit.

N_f	SLIC	SH	IBIS	LSC	SICLE-IRREG	SICLE-COMP
50	**0.610±0.027**	0.856±0.026	**0.762±0.027**	**0.774±0.026**	0.808±0.065	1.031±0.047
250	**0.612±0.027**	0.857±0.028	0.753±0.025	0.789±0.027	**0.678±0.030**	**0.746±0.035**
500	**0.615±0.027**	0.853±0.026	0.752±0.025	0.787±0.029	**0.567±0.024**	**0.718±0.030**
1000	**0.620±0.028**	0.857±0.026	0.748±0.025	0.803±0.030	**0.538±0.021**	**0.566±0.023**

Table 6.1 – Speed performance, in seconds, of the fastest methods on the BSDS500 dataset when no object information is provided. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f.

Regarding speed, Table 6.1 compares the performance of the fastest methods in the BSDS500 dataset. As one may note, SLIC is the fastest method for most values of N_f, but followed by SICLE variants. Although SLIC, IBIS, and LSC are $O(|V|)$ and SICLE variants are $O(|V| \log |V|)$, the strategy of avoiding recomputing unnecessary iterations proves to be worthy since SICLE's performance approximates to SLIC's and surpasses the remaining baselines as N_f increases. For all the previous three, they perform a strict number of 10 iterations irrespective of N_f. One would argue that SH is the fastest since it would not require several executions for generating a multiscale segmentation $\mathcal{M}$. In fact, only SICLE and SH produce a multiscale segmentation in $O(1)$ complexity, while the remaining methods would require $k \in \mathbb{N}_+$ iterations for generating k scales. However, SICLE generates the exact number of scales required by the user, whereas SH may unnecessarily exceed such quantity by computing the whole hierarchy.

In Figure 6.6, we can perceive the effective performance of the novel SICLE variants by their delineation of the parasite egg. Compared with irregular methods, SICLE-IRREG best approximates the object borders and presents minimal leaking errors. Although SH also manages to present accurate object delineation, it still presents leaking around the object, as can be seen for ERS. When comparing compact methods, we can see that LSC, IBIS, and GMM are unable to delineate the object properly, although generating highly compact superpixels. SICLE-COMP, on the other hand, best approximates the object boundaries while generating regular superpixels, as can be best seen in the background.

6.5 Object-based Analysis

From SICLE-IRREG and SICLE-COMP optimized in the last section: (i) $\boldsymbol{V}_{\text{minsc}}$ and $\boldsymbol{V}_{\text{maxsc}}$; (ii) $\iota = 0.12$; (iii) $\Omega = 5$ and $\Omega = 7$; and (iv) $N_0 = 3000$; we further optimize both for effective object delineation when an object saliency map (estimated by [64]) is given.

6.5.1 Ablation Study

Figure 6.7 shows the impact of $\alpha \in \{0.0, 0.5, 1.0, 2.0, 3.0\}$ in object delineation given a saliency map generated by the U^2-Net [64]. For SICLE-IRREG, one can see that ***BR***, and especially ***UE***, is significantly improved as α grows. Clearly, $\alpha = 0.0$ is an unsuitable option for good-quality saliency estimations. However, as one may note, $\alpha \geq 2.0$ does not

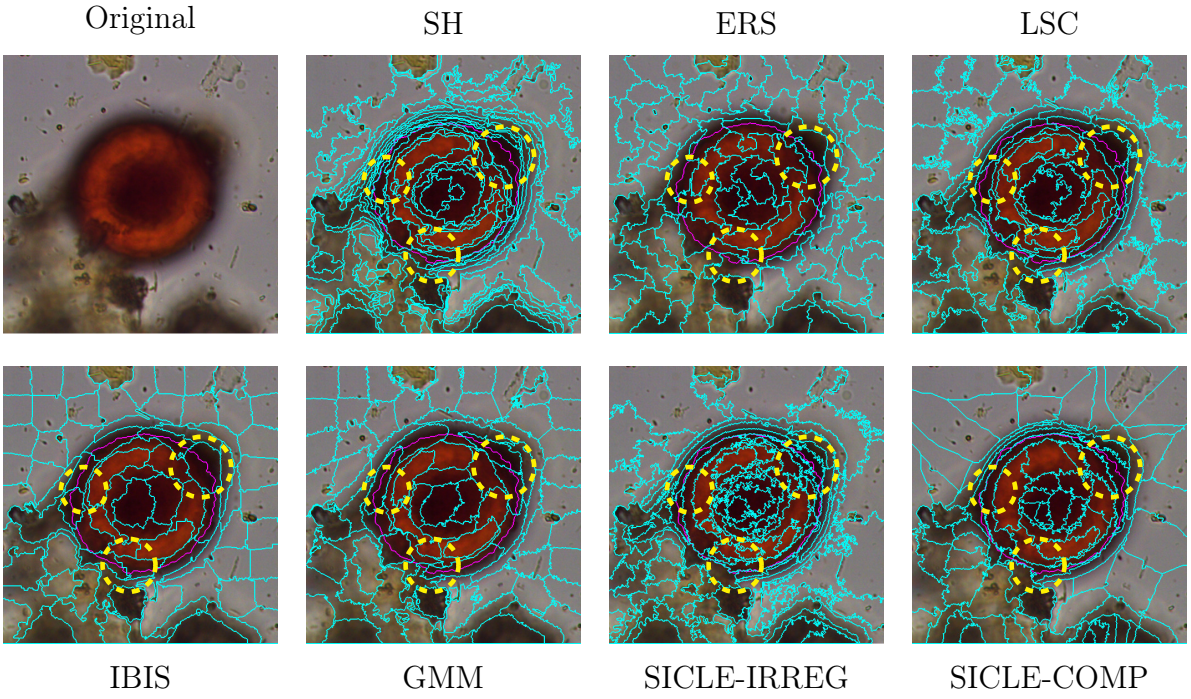


Figure 6.6 – Qualitative results with an image from Parasites requiring 100 superpixels. The object boundaries and the superpixel borders are depicted in magenta and cyan, respectively. The images were cropped for visualization purposes. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines.

show significant improvements. In fact, we argue that significantly high values of α may severely penalize saliency border crossings, leading to majorly enforced saliency borders, which is not desirable whenever the saliency given is not ideal. Therefore, we opted for $\alpha = 2.0$ for SICLE-IRREG.

For SICLE-COMP, it is interesting to see that varying α did not alter ***CO*** substantially. We infer from such observation that the saliency information is appropriately considered for avoiding leaks and improving delineation whilst promoting compacity whenever saliency borders are not crossed. Similarly to SICLE-IRREG, setting $\alpha > 2.0$ may solely reinforce saliency borders instead of improving the delineation of the image. Thus, to avoid "overfitting", we opt again for $\alpha = 2.0$ for SICLE-COMP.

Still, we further analyzed the upper and, somewhat, lower bound for our SICLE variants (Figure 6.8). For the former, we simply considered the ground truth (GT) itself as an ideal saliency map, whereas, for the latter, we computed the object's minimum bounding box (BB) as a saliency map with poor-quality borders. BB and, most importantly, GT in superpixel segmentation are considered solely for this experiment. Clearly, we did not apply any seed relevance penalization for proper delineation analysis, and for BB, we found that $\alpha = 0.0$ offers the best delineation performance. For GT maps and U^2-Net maps, we maintained the value of $\alpha = 2.0$, as optimized in the previous experiment. Analyzing SICLE-IRREG, one can see that improving saliency up to golden standards can positively impact our approaches, especially in ***BR*** and ***UE***. In other terms, although U^2-Net is an accurate saliency estimator, it is not ideal.

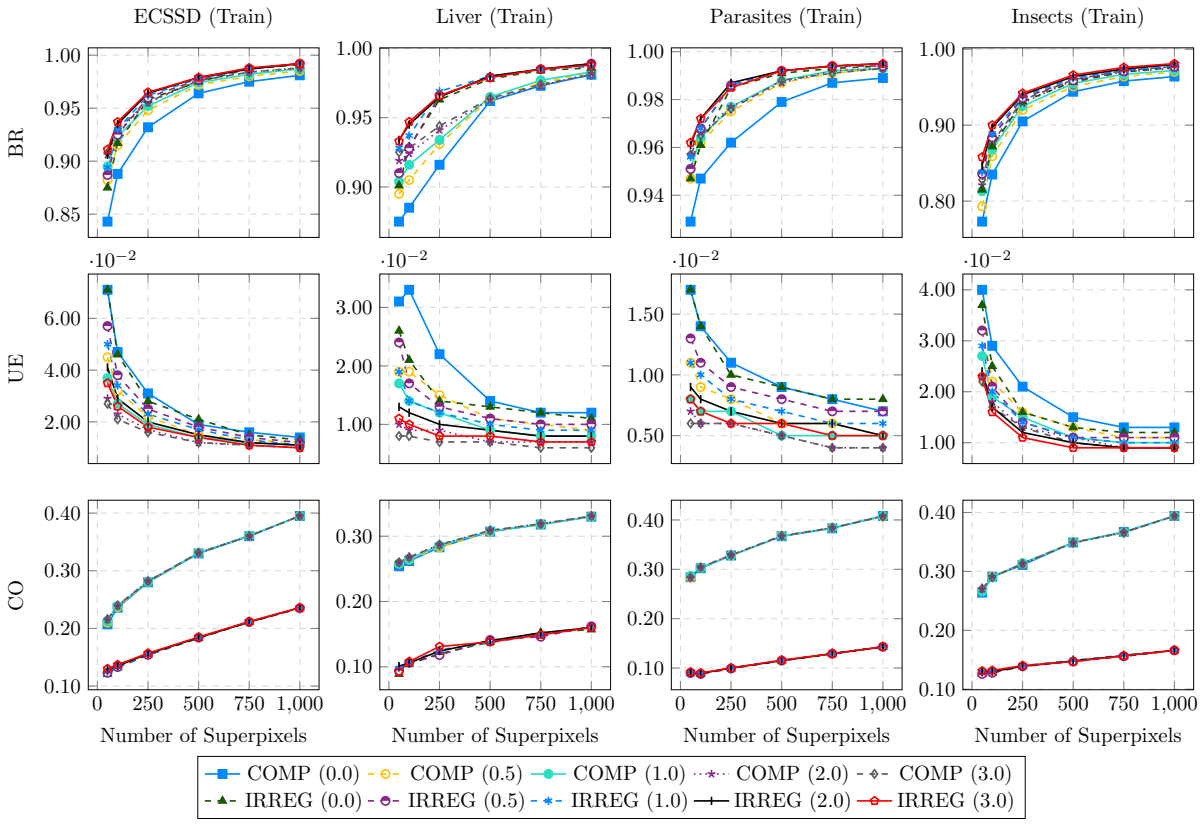


Figure 6.7 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying α and using the U^2-Net map [64].

One interesting characteristic can be seen for SICLE-COMP with GT. If an ideal saliency map is provided, SICLE-COMP may obtain maximum performance in object delineation with no impact on compactness. We argue that this behavior is equivalent to the one detailed for the previous α experiment (Figure 6.7). In fact, as shown in Figure 6.9, SICLE-COMP presents an ability previously contradictory in several state-of-the-art algorithms. That is, our method, whenever a saliency map of sufficient quality is provided, may maximize both delineation and compacity. To the best of our knowledge, no method in literature possesses a similar ability.

Similarly to the case of verifying the most suitable $\boldsymbol{V}_*$, we, in Figure 6.10, present the experiment for determining the best penalization $\boldsymbol{B}_*$ for SICLE-COMP and SICLE-IRREG. It is clear that functions that prioritize object borders, such as $\boldsymbol{B}_{\text{obj}}$ and $\boldsymbol{B}_{\text{border}}$, significantly increase $\boldsymbol{BR}$ and reduce $\boldsymbol{UE}$, which are favorable for SICLE-IRREG. On the other hand, spreading seeds, even those in the background, deteriorates delineation. We claim that background seeds also play an important role in delineation since they, against internal (object) seeds, provide extreme competition for the object boundaries, which significantly improves its segmentation. Therefore, we chose $\boldsymbol{B}_{\text{border}}$ for SICLE-IRREG due to its superior performance over all functions considered.

When analyzing SICLE-COMP, one may see that the same occurs. That is, $\boldsymbol{B}_{\text{border}}$ and $\boldsymbol{B}_{\text{obj}}$ assist in improving delineation, but they tend to deteriorate compactness, being unsuitable for our purposes. For $\boldsymbol{B}_{\text{osb}}$ and $\boldsymbol{B}_{\text{bobs}}$, one can see that both did not significantly

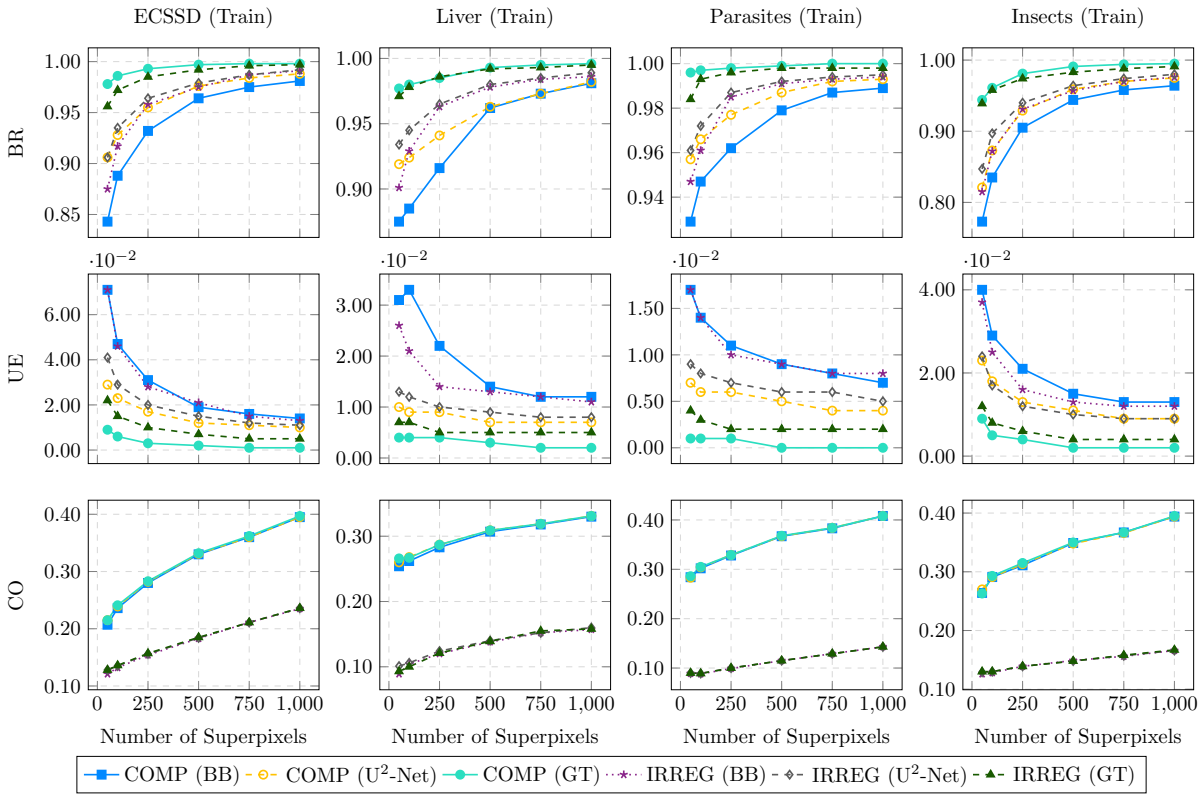
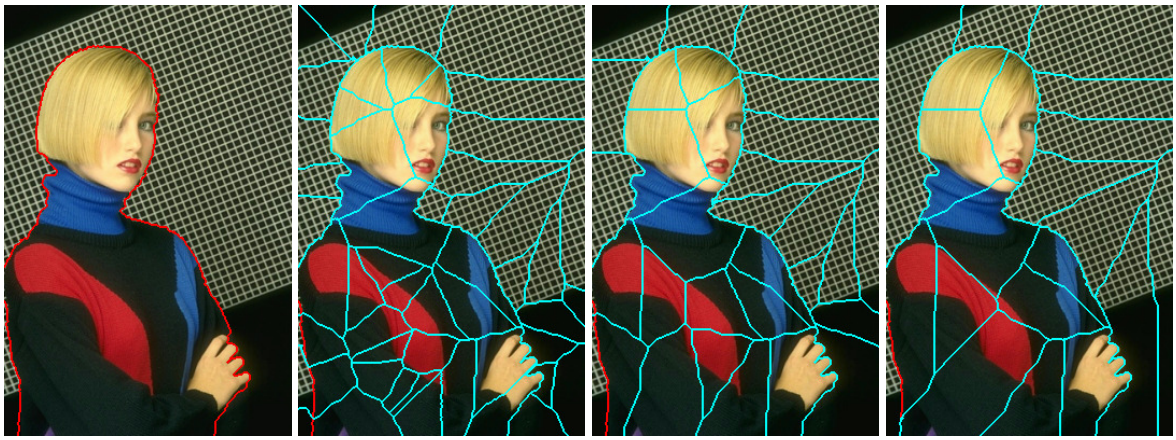


Figure 6.8 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying saliency quality: bounding box (BB), U^2-Net map [64], and ground-truth (GT).



(a) (b) (c) (d)

Figure 6.9 – Segmentation results with SICLE-COMP considering the ground-truth as saliency map. (a) original image whose object contours are depicted in red; (b-d) Results with $N_f \in \{50, 30, 20\}$. For all segmentations, $\alpha = 2.0$.

impact delineation. Yet, since our goal is to select regularly-sized superpixels, spreading and guaranteeing delineation is crucial. For this reason, we opted for $\boldsymbol{B}_{\text{bobs}}$ for SICLE-COMP.

Finally, although we already optimized SICLE-IRREG and SICLE-COMP to reduce

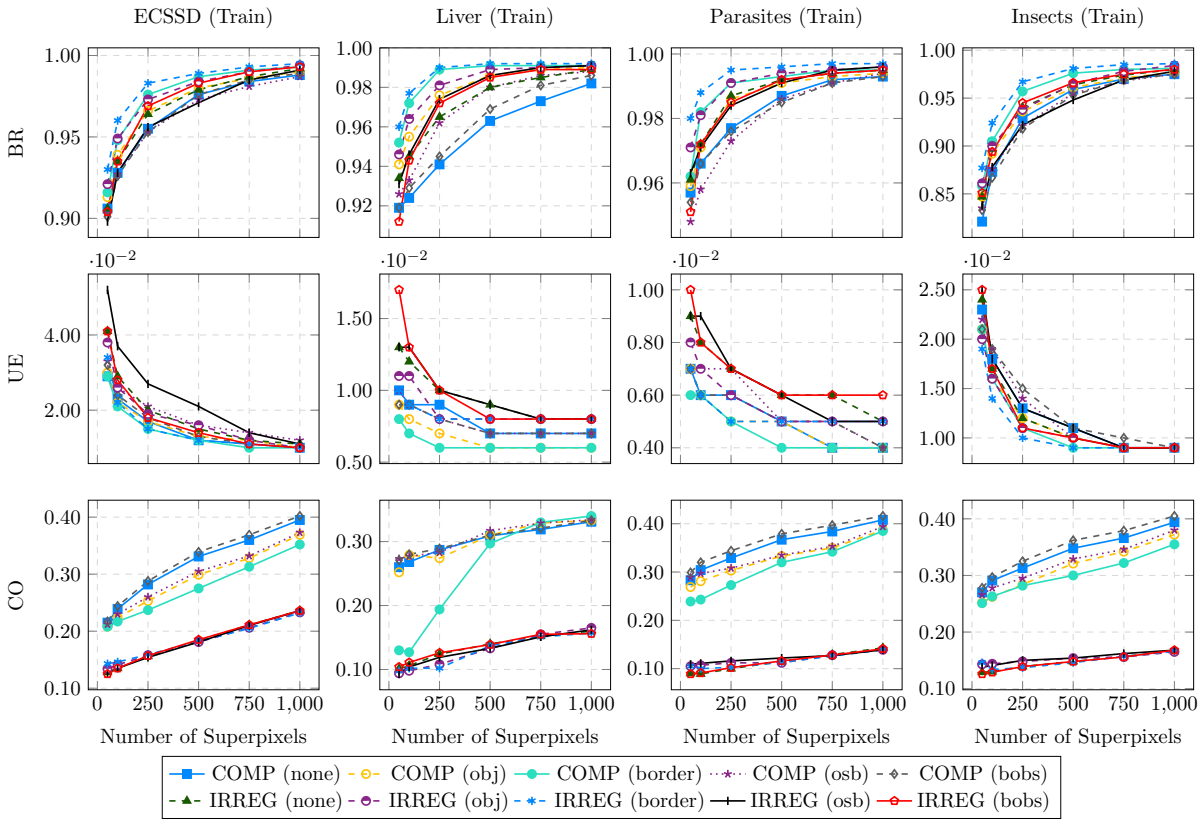


Figure 6.10 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying $\boldsymbol{B}_*$ and using the U^2-Net map [64].

the number of iterations, we performed a novel experiment to evaluate whether the saliency information allows further speed improvement by reducing unnecessary iterations. From Figure 6.11, we may see that this is the case. Simply put, only $\Omega = 2$ iterations suffice for segmentation whenever a good-quality saliency is provided. We claim that, since the set R of relevant seeds contains the necessary seeds for accurate segmentation, one can see that whenever a single object is desired, such set is significantly reduced and, thus, more irrelevant seeds will be removed. In other terms, by demanding fewer relevant seeds, more irrelevant ones may be removed in between iterations, leading to fewer iterations. Because of such fact, we set $\Omega = 2$ for both SICLE-IRREG and SICLE-COMP.

6.5.2 Quantitative and Qualitative Results

Figure 6.12 shows the quantitative results of all methods against our optimized SICLE variants. We emphasize that, in these following experiments, only saliencies estimated by [64] were considered. It is possible to see that, contrasting with Figure 6.5, SICLE variants improved their performance significantly, up to SICLE-IRREG surpassing ERS in Insects for $\boldsymbol{BR}$. Clearly, saliency plays an essential role in such improvement, especially when we consider such an accurate estimator as [64]. However, we argue that our proposals manage to incorporate such information with more perspicacity than other approaches. For instance, when seeing the performance of OISF, which considers the same saliency as

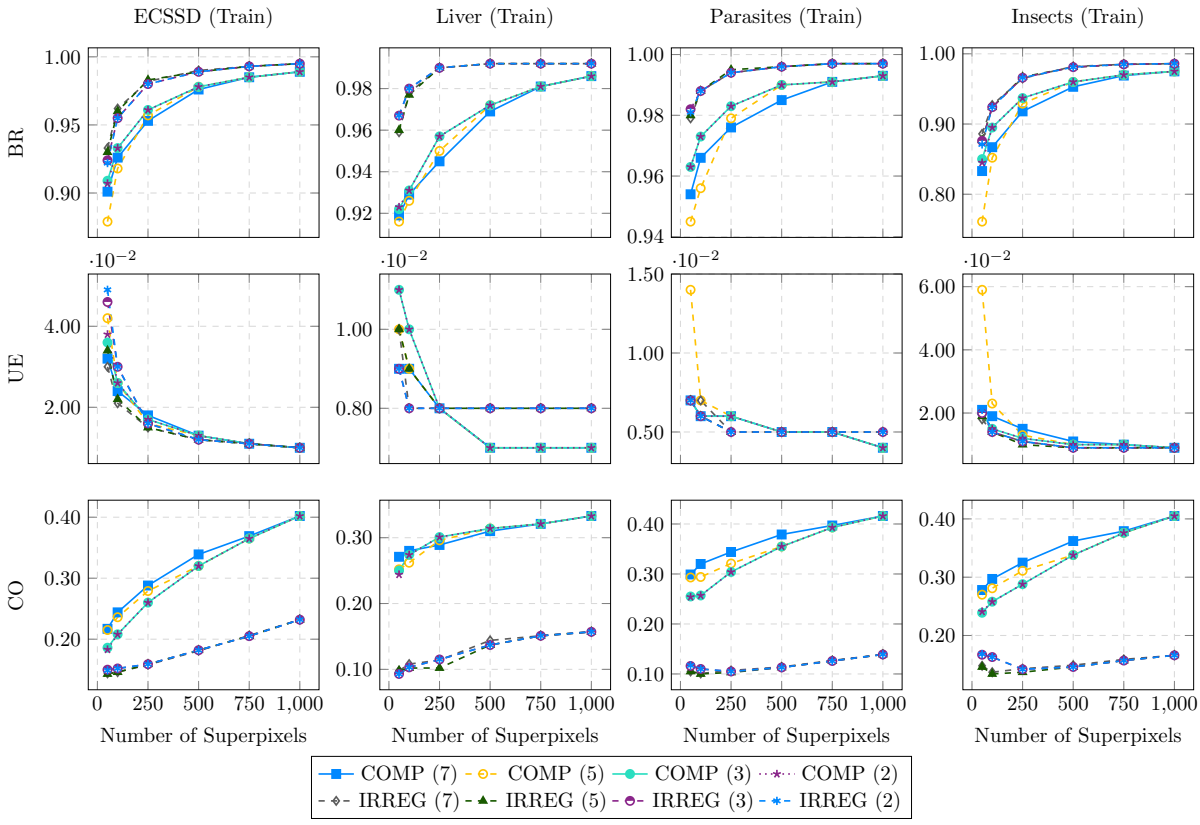


Figure 6.11 – Quantitative results on the train set of each dataset for SICLE-IRREG and SICLE-COMP for varying Ω and using the U^2-Net map [64].

SICLE variants, it is possible to note that OISF is unable to achieve similar performance, especially in Insects. As previously mentioned (in Chapter 4.2), OISF variants tend to fastly converge their performance, indicating an improvement for non-object-based approaches, but showing difficulties for further improvement as N_f increases, as expected in superpixel segmentation. This is mostly associated with the edge-cost function $\boldsymbol{w}_{\text{obj}}$ which relies significantly on the saliency quality. As consequence, OISF achieve effective delineation for few superpixels, but struggles to improve (and surpass) its counterparts as N_0 increases. Concerning ***UE***, SICLE variants managed to achieve even lower values, as commonly seen for object-based approaches [19].

Regarding SICLE-COMP, it is interesting to notice that, in terms of ***BR*** and ***UE***, it surpassed several methods, both irregular and compact approaches, such as SH. When contrasting with OISF, which generates irregular superpixels and uses saliency, SICLE-COMP achieved equivalent performance whilst requiring compact superpixels. Again, we argue that $\boldsymbol{w}_{\text{obj}}$ offers an unbalanced penalization, and deteriorates significantly the segmentation performance. It is interesting to note that, when comparing with the non-object-based scenario, this upgrade did not deteriorate SICLE-COMP's performance in ***CO***, as previously stated. When comparing with DAL-HERS, which incorporates object information through learning, in terms of producing either irregular or compact superpixels, we claim that our object-based approaches offer a more suitable methodology for object incorporation, especially when compared to the performance instability of DAL-

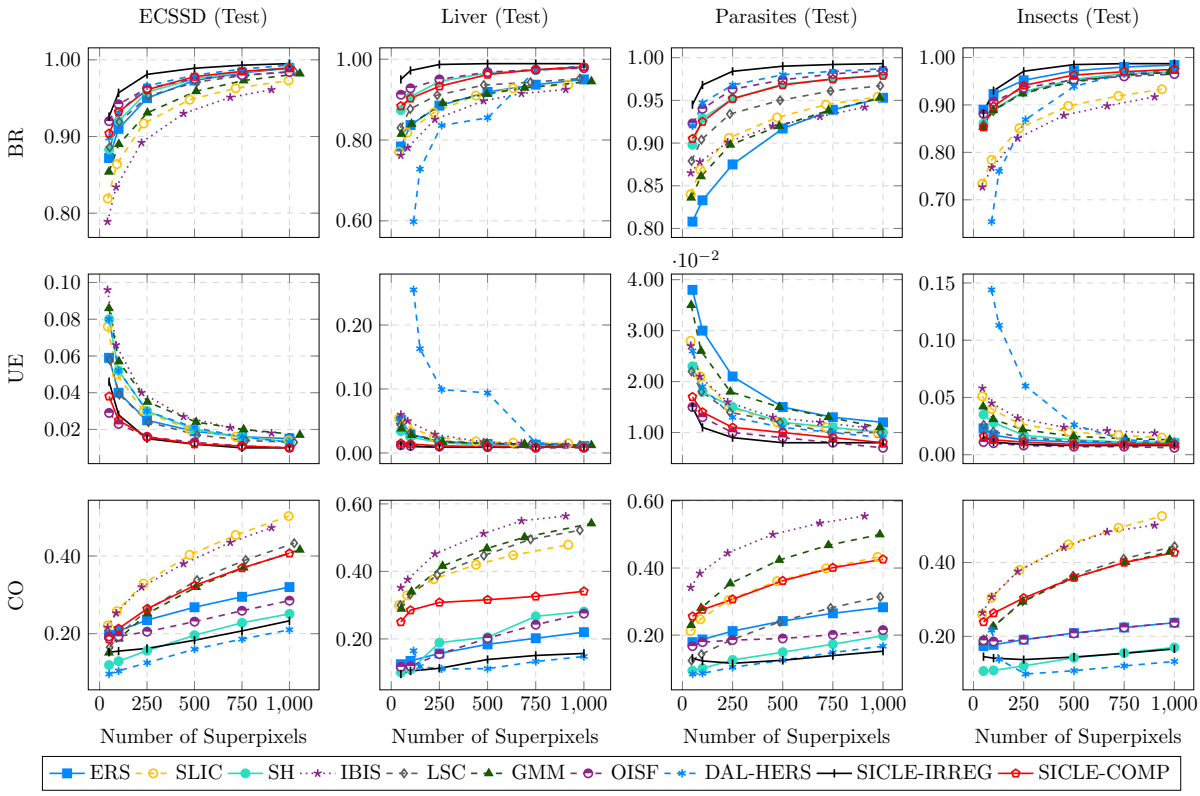


Figure 6.12 – Quantitative results on the test set of each dataset using the U^2-Net map [64].

HERS in both ***BR*** and ***UE***.

N_f	SLIC	SH	IBIS	LSC	SICLE-IRREG	SICLE-COMP
50	**0.576±0.029**	0.813±0.028	0.710±0.027	0.702±0.036	**0.501±0.055**	**0.514±0.046**
250	**0.581±0.027**	0.816±0.028	0.700±0.028	0.713±0.036	**0.503±0.040**	**0.511±0.043**
500	**0.583±0.028**	0.813±0.028	0.694±0.026	0.714±0.039	**0.507±0.039**	**0.517±0.042**
1000	**0.583±0.028**	0.812±0.027	0.694±0.028	0.718±0.036	**0.420±0.090**	**0.482±0.060**

Table 6.2 – Speed performance, in seconds, of the fastest methods on ECSSD dataset using the U^2-Net map [64]. The values depicted in bold, blue, and red indicate, respectively, the first, second, and third faster performances for each N_f.

The speed performance of the fastest methods in ECSSD is presented in Table 6.2. By comparing with Table 6.1, one can see that SICLE variants significantly improved their performance, mostly from the accurate seed removal procedure, which allowed further reducing the number of iterations for segmentation. As one may note, from $N_f = 1000$ to $N_f = 500$, both SICLE-IRREG and SICLE-COMP required slightly more time to compute the segmentation, even though only $\Omega = 2$ iterations were performed. We argue that such behavior is due to the DIFT algorithm. That is, the higher the number of seeds to be removed, the higher the number of "orphan" vertices to be conquered in the next iteration, requiring more time to compute the segmentation.

The other reason one may associate with previous researches using SICLE, which achieved faster performances [12]. We argue that the speedup achieved from exploiting DIFTs, is limited when a significantly high quantity of seeds is removed between iterations,

which is a direct result when $\Omega = 2$. Intuitively, removing superpixels can generate inconsistencies during the IFT execution, which are absent in the sequential version (as in [12]) and must be handled in the differential one. Thus, the higher the number to be removed, the higher the number of inconsistencies to be solved, impacting the overall performance. We may note such behavior on the performance discrepancy for $N_f = 1000$ and $N_f = 500$.

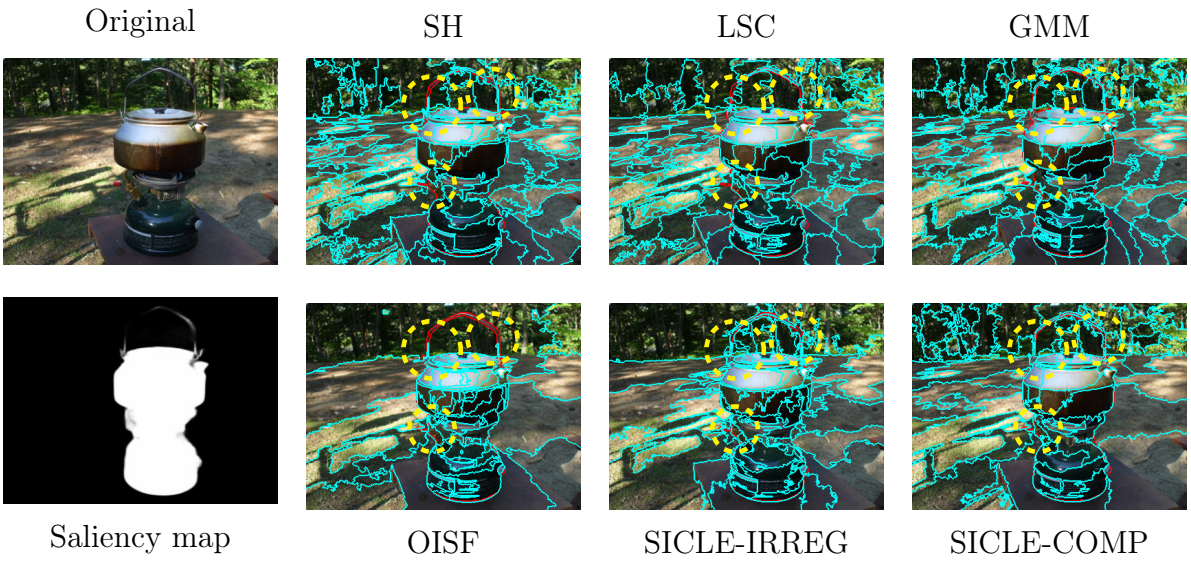


Figure 6.13 – Qualitative results with an image from ECSSD requiring 100 superpixels and using the U^2-Net estimator [64]. The object boundaries and the superpixel borders are depicted in red and cyan, respectively. Yellow circles indicate regions where SICLE best approximates the object borders in contrast with the baselines.

As an example of how SICLE variants profit from our proposals and use object information appropriately, we may note, in Figure 6.13, that OISF replicates the errors of the saliency map in its segmentation. It is possible to see that the strategy used by SICLE-IRREG and SICLE-COMP assisted in avoiding severe leaks, as seen for all methods, while achieving superior delineation amongst all of them. It is interesting to note that SICLE-COMP generated compact superpixels rather distributed along the image, resulting in less object superpixels when compared to SICLE-IRREG, but significantly more than GMM and LSC.

6.6 Hierarchiness Analysis

Figure 6.14 presents the average results of each method in Liver and Parasites datasets. Note that SH, a hierarchical method, achieves the expected behavior for all measures: maximum for nestedness and minimum for inflation ratio and refinement error. Likewise, Voronoi, due to its unstable partitioning strategy, undermines nesting regions, which explains its almost zero nestedness and quasi-maximum inflation ratio. In terms of refinement error, we argue that the Voronoi results are high due to the significant impact that random sampling imposes on the closest points to a seed. That is, different samplings in

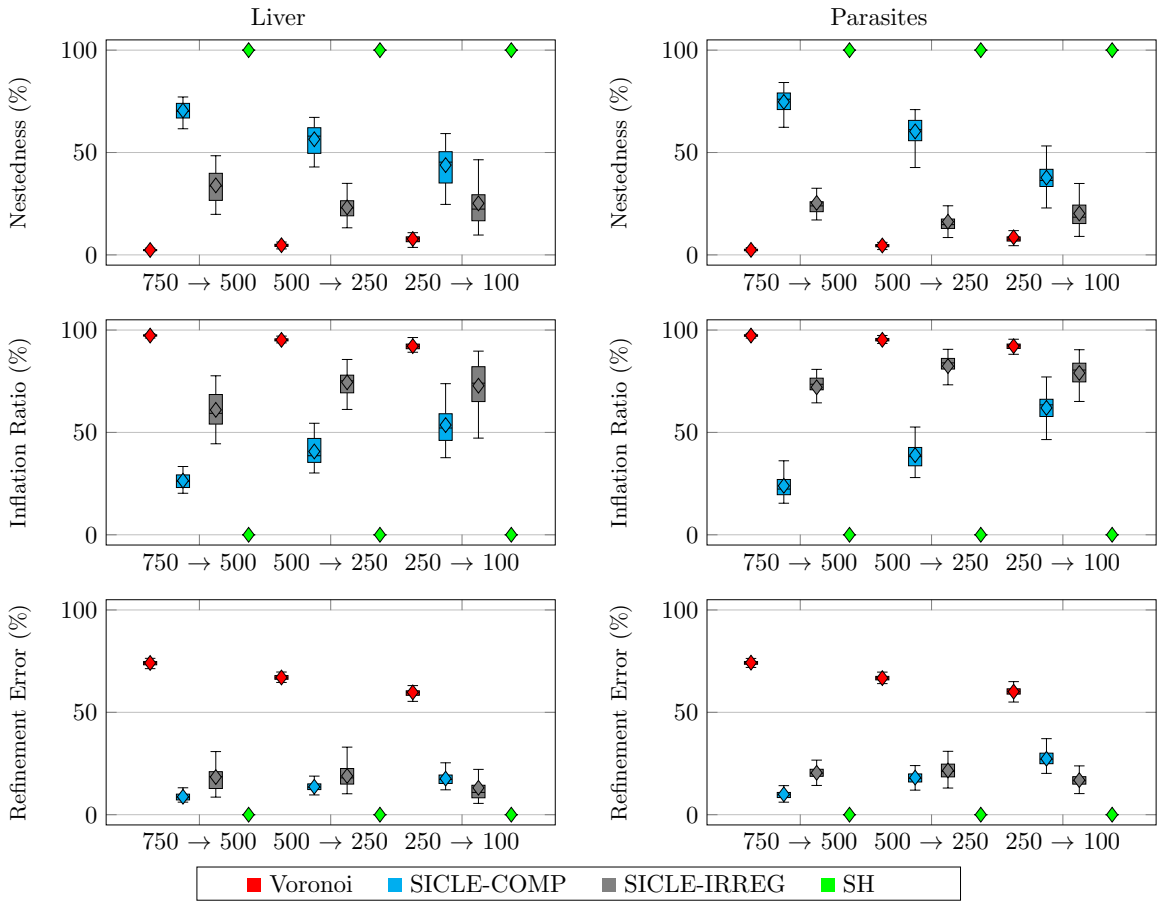


Figure 6.14 – Experimental results of the proposed hierarchiness measures on Liver and Parasites datasets (resp. left and right columns), whose average is depicted as a diamond.

subsequent scales lead to significantly different partitions, which, consequently, demands more effort to transform the regions in the preceding scale for nesting in the subsequent one. For instance, when considering $750 \rightarrow 500$, the competition for conquering pixels is near the seed, implying that slight changes in the seed coordinates severely affect the region, being necessary to substantially be reshaped. However, as the number of regions decreases, the competition is farther from the seed, limiting such impact mostly on the region's borders, which demands fewer modifications.

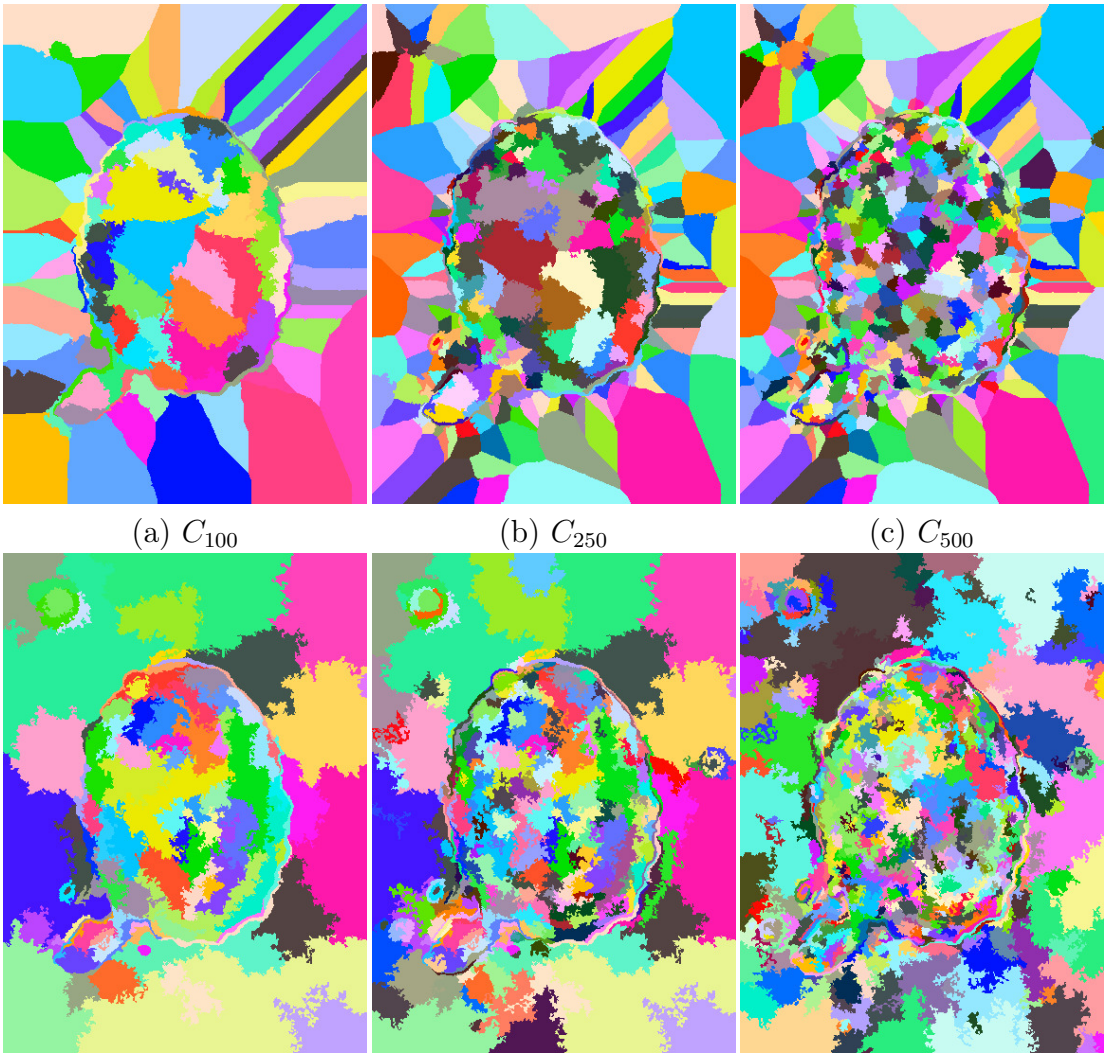
When analyzing the results from SICLE variants, one may note that the behavior is similar between datasets. For SICLE-IRREG, considering $750 \rightarrow 500$ and $500 \rightarrow 250$, the nestedness decreases while the inflation ratio increases. We argue that such behavior is expected since, by removing superpixels and using $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}^{\alpha}_{\text{root}}$, slight variations in competition may significantly impact the final segmentation result. Thus, not only sustaining nested regions tends to be challenging, but merging is also hard to perform since competition is pixel-based rather than superpixel-based. Interestingly, for $500 \rightarrow 250$ and $250 \rightarrow 100$, the nestedness performance of SICLE-IRREG does not suffer significant modifications. We claim that, since fewer superpixels are removed, the competition (and thus, reshaping) is limited to a small set of seeds. For instance, if a small to medium-sized superpixel is to be removed, its pixels will be conquered either by a superpixel whose size

(in pixels) is many magnitudes greater than the former or partially conquered by a small superpixel, whose impact is minimal with respect to the universe $|V|$ of pixels. On the other hand, when removing big superpixels, which may contain different parts of distinct objects, it is more difficult to replicate the borders of its predecessing (removed) superpixel for nesting, and thus favoring inflation.

For SICLE-COMP, it is interesting to notice that, for $750 \rightarrow 500$ and $500 \rightarrow 250$, it presents a nesting behavior significantly better than SICLE-IRREG, which also implies a smaller inflation ratio. We argue that, due to the shape regularization in $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}^{\alpha}_{\text{sum}}$, the conquering of "orphan" pixels is somewhat evenly distributed amongst the neighbors of removed superpixels. Thus, the remaining non-adjacent ones are not impacted by such modification, a feature that is not guaranteed for $\boldsymbol{w}^{\alpha}_{\text{root}}$ and $\boldsymbol{f}_{\text{max}}$ in SICLE-IRREG. For the same reason, such even conquering leads to an even impact on the inflation ratio for the adjacent superpixels, leading to either nested (equal) superpixels or slightly inflated ones. Unsurprisingly, for $500 \rightarrow 250$ and $250 \rightarrow 100$, the performance of both measures deteriorates, but, as we claim, the motivation is the same. As N_f decreases, the growth with shape regularity constraints (thus, conquering influenced by proximity) allows the merging of smaller regions with steady inflation. In contrast, SICLE-IRREG does not favor merging since its unbalanced rapid growth does not necessarily conquer all pixels from a removed superpixel.

Finally, it is worth noticing that both SICLE-COMP and SICLE-IRREG present low values of refinement error. One may conclude that, aside from the previous characteristics of both approaches, a multiscale generated by SICLE requires a small effort to resemble a hierarchy. For instance, seeing SICLE-IRREG results, a high inflation but small refinement error may indicate that the method misses in leaving a superpixel to conquer most, but not all, pixels from a removed superpixel. Conversely, for SICLE-COMP, one may expect that its high nestedness and fair inflation ratio demonstrate the ease of a superpixel to conquer a portion of "orphan" pixels but presents a high difficulty in conquering all of them, especially for a few superpixels. As one may note, for a nesting behavior, SICLE-IRREG, as we claim, would require adding the remaining pixels for merging, whereas SICLE-COMP, refine and removing the "surplus" pixels from inflation.

As one may note in Figs. 6.15(c-a), several regions in SICLE-COMP remained unaltered from one scale to the subsequent, achieving $\boldsymbol{NE}(C_{500}, C_{250}) = 71\%$, $\boldsymbol{IR}(C_{500}, C_{250}) = 30\%$, and $\boldsymbol{RE}(C_{500}, C_{250}) = 14\%$, when considering C_{500} and C_{250}. However, as the number of regions decreases, regions tend to be closer to one another, and thus, removing one or more nearby regions may destabilize the ones left out. As an example, for C_{250} and C_{100}, we have that $\boldsymbol{NE}(C_{500}, C_{250}) = 36\%$, $\boldsymbol{IR}(C_{500}, C_{250}) = 64\%$, and $\boldsymbol{RE}(C_{500}, C_{250}) = 27\%$. On the other hand, seeing the finer segmentation from SICLE-IRREG in Figure 6.15(f-d), even when removed regions are far from those left out, it may affect local tie-zones which, for a function that maximizes irregularity, can significantly alter the regions' shape in between partitions. Consequently, for I_{500} and I_{250}, we have that $\boldsymbol{NE}(I_{500}, I_{250}) = 16\%$, $\boldsymbol{IR}(I_{500}, I_{250}) = 83\%$, and $\boldsymbol{RE}(I_{500}, I_{250}) = 34\%$. In contrast, when considering coarser scales, the amount of tie-zones is significantly diminished, leading to more nested regions (*i.e.*, unaltered) and less inflation and locality errors, specifically, for I_{250} and I_{100}, $\boldsymbol{NE}(I_{250}, I_{100}) = 33\%$, $\boldsymbol{IR}(I_{250}, I_{100}) = 67\%$, and $\boldsymbol{RE}(I_{250}, I_{100}) = 18\%$.



(a) C_{100} (b) C_{250} (c) C_{500}

(d) I_{100} (e) I_{250} (f) I_{500}

Figure 6.15 – Pseudocolored segmentations using SICLE-COMP and SICLE-IRREG, respectively, identified by C_n and I_n, for $n \in \{100, 250, 500\}$ regions.

Chapter 7

Conclusion and Future Work

In this thesis, we propose *Superpixels through Iterative CLEarcutting* (SICLE), an agnostic superpixel segmentation framework, resultant from several years of study and research. Although it follows a similar pipeline to state-of-the-art approaches, SICLE variants maximize their performance by exploiting the particularities of each step. In order to avoid missing any important part of the object, a significantly higher number of initial points is then selected for segmentation. Subsequently, for allowing multiple and varied forms of partitioning an image, the use of the *Image Foresting Transform* (IFT) [35] is crucial. The last step favors the regions mostly associated with what is relevant in the application while diminishing those seen as irrelevant. From such an iterative process, SICLE variants achieve top performance in delineation, compacity, and speed when compared to different state-of-the-art approaches in superpixel segmentation.

Based on what we proposed and achieved with SICLE, we are able to answer three of our research questions stated in the first section (Section 1):

Question 1: *Is it possible to build a superpixel segmentation method that assimilates any bias?*
Yes, it is possible if one conceives such a mathematical formulation of each bias. Again, note that we assume that some biases may be mathematically defined and optimized, but there is no unique function that satisfies all biases. In SICLE, one can achieve such property, mainly through the seed relevance criterion. If a saliency is provided, it is possible to refine ambiguities defined by the criterion. For instance, if one desires to segment one object totally red, it is straightforward to define the red criterion for seed relevance estimation. However, if more red objects exist in the image, and the desired one may also be defined by shape, two possibilities arise: either modify the criterion for taking into account color and shape or, as we would suggest, recur to a saliency estimator based on shape and filter those whose size does not fit the pattern. While the former may increase the relevance of objects whose size matches but not the color, the latter evaluates those that match the color, and the penalization acts as a false-positive filtering. Clearly, for different criteria and penalizations, one may adapt the segmentation and produce a viable solution with SICLE with ease.

Question 2: *Is it possible to build a superpixel segmentation method that promotes any*

morphology?
Yes, similarly, it is possible if one conceives such a mathematical formulation of each morphology. Likewise, we say that any morphology may be mathematically defined, but no function (satisfactory) generates all possible morphologies with simple parameter tuning. In this thesis, $\boldsymbol{f}_{\text{sum}}$ and $\boldsymbol{w}_{\text{sum}}$ promote superpixels with circular shapes. However, for other shapes, such as striped superpixels, one may compute the distance between vertices $p, q \in V$ by $\|p_x - q_x\|_2$ or, analogously, by $\|p_y - q_y\|_2$. Note that since IFT minimizes cost, the former will favor "vertical" paths, whereas the latter will favor "horizontal paths". Simply put, SICLE can promote any morphology if one provides a "vertex-wise" regularization factor to be applied for optimum-paths during their growth in the IFT execution.

Question 3: *Is it possible to build a superpixel segmentation method that achieves both the latter with efficacy and efficiency for any object?*
Yes, especially considering our previous answers. From our results, SICLE is the most effective and efficient superpixel method in all datasets compared to several state-of-the-art algorithms. Thus, our contribution manages to achieve top delineation for clearly distinct domains, each one with its own bias. Obviously, the variants evaluated in SICLE are not the best ones possible for each dataset; rather, they are viable off-the-shelf solutions for all evaluated datasets. To achieve the ideal SICLE variant for each dataset, one would, as mentioned, conceive different seed criteria and, possibly, different seed penalizations for filtering false positives. Likewise, compactness is only one of the possible morphologies, but it is the most demanded one. Thus, alongside fine-tailored criterion and penalization, one would need a highly specialized regularization factor to generate the expected morphology. If one only aims for maximum delineation, the task is simpler since no regularization is required (and simply recurs to $\boldsymbol{f}_{\max}$ and $\boldsymbol{w}_{\text{root}}$). Finally, as previously stated, a highly-adapted SICLE variant may only require two iterations for accurate segmentation if the functions are accurate in their proposals.

Furthermore, in this thesis, we propose several measures for identifying whether a multiscale segmentation resembles a hierarchy (*i.e.*, its *hierarchiness*). For that, we identify eight possible cases when comparing regions in-between subsequent segmentations, and from their intersection, we are able to define the *nestedness*, the *inflation ratio*, and the *refinement error* measures. Simply put, nestedness evaluates the nesting behavior between segmentations, while the inflation ratio computes the portion resultant from inflating the regions in the preceding scale. Finally, the refinement error computes the "minimum effort" to alter the regions so that the preceding scale is now hierarchically nested in the subsequent one. Results show that our measures provide substantial insight into the violations in the multiscale segmentation, if there are any.

From our hierarchiness measures, we may answer our remaining research question:

Question 4: *Is it possible to conceive a measure which estimates the resemblance of a multiscale segmentation to a hierarchical one?*
Yes, as our contributions have shown. Through ***NE***, ***IR***, and ***RE***, one can estimate the resemblance of a multiscale segmentation to a hierarchical one by comparing two

subsequent partitions in a sequence. Since the nesting behavior is present throughout the scales, such pairwise comparison also guarantees a proper evaluation of hierarchiness. Similarly, considering the contours of the preceding scale as ground truth, it is possible to measure whether it is more practical to merge regions or to remove them so that nesting is now present. Finally, whenever merging regions is insufficient, our proposals can also measure the rate of inflation to transform a partition X into its subsequent Y. In the end, from our measures, one can perceive several characteristics of a multiscale segmentation and make a more informed decision about how to transform it into a hierarchical one.

For future work, we intend to further evaluate SICLE for multidimensional applications, such as n-dimensional media and, also, for clustering non-media data as in [5]. In terms of IFT, we intend to study new connectivity functions in which borders are guaranteed independently of inserting or, more importantly, removing seeds. Our motivation lies in providing a connectivity function that generates a hierarchical segmentation through several executions of the IFT, differential or not. On the other hand, we may study new heuristics to further improve SICLE's performance during the IFT execution.

With respect to the hierarchiness measures, we aim, for future endeavors, to provide measures with respect to other cases, especially those related to shrinking regions, such as full deflation and deflation and split. Also, we intend to provide more robust measures that take into account the possibility of altering the multiscale sequence, a property that our contributions yet to possess, with operations such as including, removing or altering the sequence of partitions. Finally, we also aim to provide "relaxed" hierarchiness measures with similar properties to be used in optimization tasks and machine learning techniques.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[2] R. Achanta and S. Süsstrunk. Superpixels and polygons using simple non-iterative clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4895–4904, 2017.

[3] C. Allène, J. Audibert, M. Couprie, J. Cousty, and R. Keriven. Some links between min-cuts, optimal spanning forests and watersheds. In *Proceedings of the 8th International Symposium on Mathematical Morphology*, volume 1, pages 256–264, 2007.

[4] J. Angulo and J. Dominique. Stochastic watershed segmentation. In *Proceedings of the 8th International Symposium on Mathematical Morphology*, volume 1, pages 265–276, 2007.

[5] D. Aparco-Cardenas, P. Rezende, and A. Falcão. An iterative optimum-path forest framework for clustering. In *Optimum-Path Forest*, pages 175–216. Academic Press, 2022.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[7] M. Awaisu, L. Li, J. Peng, and J. Zhang. Fast superpixel segmentation with deep features. In *Advances in Computer Graphics*, volume 11542, pages 410–416, 2019.

[8] Z. Ban, J. Liu, and L. Cao. Superpixel segmentation using gaussian mixture model. *IEEE Transactions on Image Processing*, 27(8):4105–4117, 2018.

[9] I. Barcelos, F. Belém, L. João, A. Falcão, and S. Guimarães. Improving color homogeneity measure in superpixel segmentation assessment. In *35th Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 79–84, 2022.

[10] I. Barcelos, F. Belém, L. João, Z. Patrocínio, A. Falcão, and S. Guimarães. A comprehensive review and new taxonomy on superpixel segmentation. *ACM Computing Surveys*, 56(8):1–39, 2024.

[11] I. Barcelos, F. Belém, P. Miranda, A. Falcão, Z. Patrocínio, and S. Guimarães. Towards interactive image segmentation by dynamic and iterative spanning forest. In *Discrete Geometry and Mathematical Morphology*, pages 351–364, 2021.

[12] F. Belém, I. Borlido, L. João, B. Perret, J. Cousty, S. Guimarães, and A. Falcão. Fast and effective superpixel segmentation using accurate saliency estimation. In *Discrete Geometry and Mathematical Morphology*, pages 261–273, 2022.

[13] F. Belém, I. Borlido, L. João, B. Perret, J. Cousty, S. Guimarães, and A. Falcão. Novel arc-cost functions and seed relevance estimations for compact and accurate superpixels. *Journal of Mathematical Imaging and Vision*, 65(5):770–786, 2023.

[14] F. Belém, J. Cousty, B. Perret, S. Guimarães, and A. Falcão. Towards a simple and efficient object-based superpixel delineation framework. In *34th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 346–353, 2021.

[15] F. Belém, F. Faria, A. Falcão, Z. Patrocínio, S. Guimarães, B. Perret, and J. Cousty. Measuring hierarchiness of image segmentations, 2024. To appear.

[16] F. Belém, S. Guimarães, and A. Falcão. Superpixel segmentation by object-based iterative spanning forest. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 11401, pages 334–341, 2019.

[17] F. Belém, S. Guimarães, and A. Falcão. Superpixel generation by the iterative spanning forest using object information. In *33rd Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 22–28, 2020. Workshop of Thesis and Dissertations.

[18] F. Belém, S. Guimarães, and A. Falcão. Superpixel segmentation using dynamic and iterative spanning forest. *Signal Processing Letters*, 27:1440–1444, 2020.

[19] F. Belém, L. Melo, S. Guimarães, and A. Falcão. The importance of object-based seed sampling for superpixel segmentation. In *32nd Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 108–115, 2019.

[20] F. Belém, B. Perret, J. Cousty, S. Guimarães, and A. Falcão. Efficient multiscale object-based superpixel framework, 2022.

[21] F. Belém, B. Perret, J. Cousty, S. Guimarães, and A. Falcão. Efficient multiscale object-based superpixel framework, 2024. Submitted to the Journal of the Brazilian Computer Society.

[22] M. Bergh, X. Boix, G. Roig, and L. Gool. SEEDS: Superpixels extracted via energy-driven sampling. *International Journal of Computer Vision*, 111(3):298–314, 2015.

[23] S. Beucher. *Watershed, Hierarchical Segmentation and Waterfall Algorithm*, pages 69–76. 1994.

[24] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*, volume 132, 1979.

[25] S. Bobbia, R. Macwan, Y. Benezeth, K. Nakamura, R. Gomez, and J. Dubois. Iterative boundaries implicit identification for superpixels segmentation: A real-time approach. *IEEE Access*, 9:77250–77263, 2021.

[26] J. Bragantini, S. Martins, C. Castelo-Fernandez, and A. Falcão. Graph-based image segmentation using dynamic trees. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 470–478, 2018.

[27] P. Buyssens, I. Gardin, S. Ruan, and A. Elmoataz. Eikonal-based region growing for efficient clustering. *Image and Vision Computing*, 32(12):1045–1054, 2014.

[28] C. Castelo-Fernandez and A. Falcão. Learning visual dictionaries from class-specific superpixel segmentation. In *Computer Analysis of Images and Patterns*, pages 171–182, 2019.

[29] K. Ciesielski, A. Falcão, and P. Miranda. Path-value functions for which dijkstra's algorithm returns optimal mapping. *Journal of Mathematical Imaging and Vision*, 60(7):1025–1036, 2018.

[30] M. Condori, F. Cappabianco, A. Falcão, and P. Miranda. An extension of the differential image foresting transform and its application to superpixel generation. *Journal of Visual Communication and Image Representation*, 71:102748, 2020.

[31] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1384–1399, 2011.

[32] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, 2009.

[33] S. Dhore and D. Abin. Chest x-ray segmentation using watershed and super pixel segmentation technique. In *4th International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–4, 2021.

[34] A. Falcão and F. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Transactions on Medical Imaging*, 23(9):1100–1108, 2004.

[35] A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[36] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[37] F. Galvão, A. Falcão, and A. Chowdhury. RISF: recursive iterative spanning forest for superpixel segmentation. In *31st Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 408–415, 2018.

[38] U. Gaur and B. S. Manjunath. Superpixel embedding network. *IEEE Transactions on Image Processing*, 29:3199–3212, 2020.

[39] R. Giraud, V. Ta, and N. Papadakis. SCALP: Superpixels with contour adherence using linear path. In *23rd International Conference on Pattern Recognition (ICPR)*, pages 2374–2379, 2016.

[40] H. Gonçalves, G. Vasconcelos, P. Rangel, M. Carvalho, N. Archilha, and T. Spina. cudaift: 180x faster image foresting transform for waterpixel estimation using cuda. In *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, volume 4, pages 395–404, 2019.

[41] Y. Gong and Y. Zhou. Differential evolutionary superpixel segmentation. *IEEE Transactions on Image Processing*, 27(3):1390–1404, 2017.

[42] L. Guigues, J. Cocquerez, and H. Le Men. Scale-sets image analysis. *International Journal of Computer Vision*, 68(3):289–317, 2006.

[43] S. Guimarães, Y. Kenmochi, J. Cousty, Z. Patrocínio, and L. Najman. Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the felzenszwalb-huttenlocher method. *Mathematical Morphology - Theory and Applications*, 2(1):55–75, 2017.

[44] A. Ilesanmi, O. Idowu, and S. Makhanov. Multiscale superpixel method for segmentation of breast ultrasound. *Computers in Biology and Medicine*, 125:103879, 2020.

[45] V. Jampani, D. Sun, M. Liu, M. Yang, and J. Kautz. Superpixel sampling networks. In *European Conference on Computer Vision (ECCV)*, volume 11211, pages 363–380, 2018.

[46] C. Jerônimo, F. Belém, S. Carneiro, Z. Patrocínio, L. Najman, A. Falcão, and S. Guimarães. Graph-based supervoxel computation from iterative spanning forest. In *Discrete Geometry and Mathematical Morphology*, pages 404–415, 2021.

[47] R. Kalboussi, M. Abdellaoui, and A. Douik. Video saliency using supervoxels. In *Intelligent Interactive Multimedia Systems and Services (IIMSS)*, pages 544–553, 2018.

[48] X. Kang, L. Zhu, and A. Ming. Dynamic random walk for superpixel segmentation. *IEEE Transactions on Image Processing*, 29:3871–3884, 2020.

[49] L. Lacerda, F. Belém, A. Falcão, Z. Patrocínio, and S. Guimarães. Seed-based superpixel re-segmentation for improving object delineation, 2024. Submitted to the Iberoamerican Conference on Pattern Recognition.

[50] Q. Leng, H. Yang, J. Jiang, and Q. Tian. Adaptive multiscale segmentations for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(8):5847–5860, 2020.

[51] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. TurboPixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–2297, 2009.

[52] Z. Li and J. Chen. Superpixel segmentation using linear spectral clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1356–1363, 2015.

[53] C. Liu, R. Zhao, and M. Pang. A fully automatic segmentation algorithm for ct lung images based on random forest. *Medical Physics*, 47(2):518–529, 2019.

[54] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2097–2104, 2011.

[55] Y. Liu, M. Yu, B. Li, and Y. He. Intrinsic manifold SLIC: A simple and efficient method for computing content-sensitive superpixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):653–666, 2018.

[56] Y. Lv, N. Han, and H. Du. Estimation of bamboo forest aboveground carbon using the rglm model based on object-based multiscale segmentation of SPOT-6 imagery. *Remote Sensing*, 15(10), 2023.

[57] V. Machairas, M. Faessel, D. Cárdenas-Peña, T. Chabardes, T. Walter, and E. Decenciere. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015.

[58] L. Mansilla and P. Miranda. Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. In *29th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 289–296, 2016.

[59] S. Martins, G. Ruppert, F. Reis, C. Yasuda, and A. Falcão. A supervoxel-based approach for unsupervised abnormal asymmetry detection in MR images of the brain. In *IEEE 16th International Symposium on Biomedical Imaging (ISBI)*, pages 882–885, 2019.

[60] P. Neubert and P. Protzel. Superpixel benchmark and comparison. In *Forum Bildverarbeitung*, volume 6, pages 1–12, 2012.

[61] N. Passat, B. Naegel, F. Rousseau, M. Koob, and J. Dietemann. Interactive segmentation based on component-trees. *Pattern Recognition*, 44(10):2539–2554, 2011.

[62] H. Peng, A. Aviles-Rivero, and C. Schonlieb. Hers superpixels: Deep affinity learning for hierarchical entropy rate segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 72–81, 2022.

[63] J. Pont-Tuset and F. Marqués. Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1–1, 2015.

[64] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. Zaiane, and M. Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.

[65] C. Ronse. Orders on partial partitions and maximal partitioning of sets. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 49–60, 2011.

[66] A. Schick, M. Fischer, and R. Stiefelhagen. An evaluation of the compactness of superpixels. *Pattern Recognition Letters*, 43:71–80, 2014.

[67] J. Serra. Grain building ordering. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 37–48, 2011.

[68] J. Shi, Q. Yan, L. Xu, and J. Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):717–729, 2015.

[69] I. Silva, A. Sousa, A. Falcão, and J. Bragantini. Differential dynamic trees for interactive image segmentation. In *26th International Conference on Pattern Recognition (ICPR)*, pages 4328–4334, 2022.

[70] A. Sousa, S. Martins, A. Falcão, F. Reis, E. Bagatin, and K. Irion. Altis: A fast and automatic lung and trachea ct-image segmentation method. *Medical Physics*, 46(11):4970–4982, 2019.

[71] D. Stutz, A. Hermans, and B. Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.

[72] T. Suzuki. Superpixel segmentation via convolutional neural networks with regularized information maximization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2573–2577, 2020.

[73] S. Todorovic and M.C. Nechyba. Dynamic trees for unsupervised segmentation and matching of image regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1762–1777, 2005.

[74] W. Tu, M. Liu, V. Jampani, D. Sun, S. Chien, M. Yang, and J. Kautz. Learning superpixels with segmentation-aware affinity loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 568–576, 2018.

[75] J. Vargas-Muñoz, A. Chowdhury, E. Alexandre, F. Galvão, P. Miranda, and A. Falcão. An iterative spanning forest framework for superpixel segmentation. *IEEE Transactions on Image Processing*, 28(7):3477–3489, 2019.

[76] D. Vieira, I. Borlido, , F. Belém, Z. Patrocínio, A. Falcão, and S. Guimarães. Streaming graph-based supervoxel computation based on dynamic iterative spanning forest. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 90–104, 2024.

[77] L. Wan, X. Xu, Q. Zhao, and W. Feng. Spherical superpixels: Benchmark and evaluation. In *Asian Conference on Computer Vision (ACCV)*, volume 11366, pages 703–717, 2019.

[78] M. Wang, X. Liu, Y. Gao, X. Ma, and N. Soomro. Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28–39, 2017.

[79] Y. Wang, Y. Wei, X. Qian, L. Zhu, and Y. Yang. AINet: Association implantation for superpixel segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7058–7067, 2021.

[80] X. Wei, Q. Yang, Y. Gong, N. Ahuja, and M. Yang. Superpixel hierarchy. *IEEE Transactions on Image Processing*, 27(10):4838–4849, 2018.

[81] J. Wu, C. Liu, and B. Li. Texture-aware and structure-preserving superpixel segmentation. *Computers and Graphics*, 94:152–163, 2021.

[82] Z. Wu, L. He, Z. Hu, Y. Zhang, and G. Wu. Hierarchical segmentation evaluation of region-based image hierarchy. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):2718–2727, 2019.

[83] X. Xiao, Y. Zhou, and Y. Gong. Content-adaptive superpixel segmentation. *IEEE Transactions on Image Processing*, 27(6):2883–2896, 2018.

[84] L. Xu, L. Zeng, and Z. Wang. Saliency-based superpixels. *Signal, Image and Video Processing*, 8(1):181–190, 2014.

[85] F. Yang, Q. Sun, H. Jin, and Z. Zhou. Superpixel segmentation with fully convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13961–13970, 2020.

[86] J. Yao, M. Boben, S. Fidler, and R. Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2947–2955, 2015.

[87] Y. Yu, Y. Makihara, and Y. Yagi. Pedestrian segmentation based on a spatio-temporally consistent graph-cut with optimal transport. *Transactions on Computer Vision and Applications*, 11(1):10, 2019.

[88] Y. Yu, Y. Yang, and K. Liu. Edge-aware superpixel segmentation with unsupervised convolutional neural networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 1504–1508, 2021.

[89] Q. Yuan, S. Lu, Y. Huang, and W. Sha. SIN: Superpixel interpolation network. In *Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, pages 293–307, 2021.

[90] S. Zhang, H. Wang, W. Huang, and Z. You. Plant diseased leaf segmentation and recognition by fusion of superpixel, K-means and PHOG. *Optik*, 157:866–872, 2018.

[91] W. Zhang, D. Xiang, and Y. Su. Fast multiscale superpixel segmentation for sar imagery. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[92] J. Zhao, R. Bo, Q. Hou, M. Cheng, and P. Rosin. FLIC: Fast linear iterative clustering with active search. *Computational Visual Media*, 4(4):333–348, 2018.

[93] J. Zhou, J. Ruan, C. Wu, G. Ye, Z. Zhu, J. Yue, and Y. Zhang. Superpixel segmentation of breast cancer pathology images based on features extracted from the autoencoder. In *IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, pages 366–370, 2019.

[94] L. Zhu, Q. She, B. Zhang, Y. Lu, Z. Lu, D. Li, and J. Hu. Learning the superpixel in a non-iterative and lifelong manner. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1225–1234, 2021.