

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

Zona de empate: o elo entre transformadas de *watershed* e conexidade nebulosa

Autor: Romaric Matthias Michel Audigier

Orientador: Prof. Dr. Roberto de Alencar Lotufo

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Engenharia de Computação**.

Banca Examinadora

Prof. Dr. Alexandre Xavier Falcão IC/Unicamp
Prof. Dr. Clésio Luis Tozzi DCA/FEEC/Unicamp
Prof. Dr. Fernando José Von Zuben DCA/FEEC/Unicamp
Prof. Dr. Jorge Stolfi IC/Unicamp
Prof. Dr. Sérgio Shigemi Furuie InCor/USP-São Paulo

Julho de 2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Au24z Audigier, Romaric Matthias Michel
 Zona de empate: o elo entre transformadas de watershed
 e conexidade nebulosa / Romaric Matthias Michel
 Audigier. –Campinas, SP: [s.n.], 2007.

 Orientador: Roberto de Alencar Lotufo.
 Tese (doutorado) - Universidade Estadual de Campinas,
 Faculdade de Engenharia Elétrica e de Computação.

 1. Processamento de imagens. 2. Morfologia
 matemática. 3. Teoria de grafos. 4. Árvores (Teoria dos
 grafos). I. Lotufo, Roberto de Alencar. II. Universidade
 Estadual de Campinas. Faculdade de Engenharia Elétrica e
 de Computação. III. Título.

Título em Inglês: Tie-zone: the bridge between watershed transforms and fuzzy
connectedness.

Palavras-chave em Inglês: Image segmentation, Mathematical morphology, Watershed,
Image-foresting transform (IFT), Shortest-path forest,
Minimum spanning forest, Graph theory, Fuzzy
connectedness.

Área de concentração: Engenharia de Computação

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Alexandre Xavier Falcão, Clésio Luis Tozzi, Fernando José Von
Zuben, Jorge Stolfi e Sérgio Shigemi Furuie.

Data de defesa: 31/07/2007

Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – TESE DE DOUTORADO

Candidato(a): Romaric Matthias Michel Audigier

Data da Defesa: 31 de Julho de 2007

Título da Tese: "Zona de Empate: O Elo entre Transformadas de Watershed e Conexidade Nebulosa"

Prof. Dr. Roberto de Alencar Lotufo (Matr. 59145): 

Dr. Sérgio Shiguemi Furuie:  Sérgio Furuie

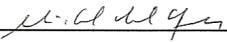
Prof. Dr. Jorge Stolfi: 

Prof. Dr. Alexandre Xavier Falcão:  Alexandre Falcão

Prof. Dr. Clésio Luiz Tozzi: 

Prof. Dr. Fernando José Von Zuben:  Fernando José Von Zuben

Secretário(a) Edson Vicente Sanches Filho:  Edson V. Sanches Filho

Coordenador de PG Prof. Dr. Michel Daoud Yacoub:  Michel Daoud Yacoub

Resumo

Esta tese introduz o novo conceito de transformada de zona de empate que unifica as múltiplas soluções de uma transformada de *watershed*, conservando apenas as partes comuns em todas estas, tal que as partes que diferem constituem a zona de empate. A zona de empate aplicada ao *watershed* via transformada imagem-floresta (TZ-IFT-WT) se revela um elo inédito entre transformadas de *watershed* baseadas em paradigmas muito diferentes: gota d'água, inundação, caminhos ótimos e floresta de peso mínimo. Para todos esses paradigmas e os algoritmos derivados, é um desafio se ter uma solução única, fina, e que seja consistente com uma definição. Por isso, propõe-se um afinamento da zona de empate, único e consistente. Além disso, demonstra-se que a TZ-IFT-WT também é o dual de métodos de segmentação baseados em conexidade nebulosa. Assim, a ponte criada entre as abordagens morfológica e nebulosa permite aproveitar avanços de ambas. Em consequência disso, o conceito de núcleo de robustez para as sementes é explorado no caso do *watershed*.

Palavras-chave: segmentação de imagens, morfologia matemática, *watershed*, transformada imagem floresta (IFT), floresta de caminhos mínimos, floresta de peso mínimo, teoria dos grafos, conexidade nebulosa.

Abstract

This thesis introduces the new concept of tie-zone transform that unifies the multiple solutions of a watershed transform, by conserving only the common parts among them such that the differing parts constitute the tie zone. The tie zone applied to the watershed via image-foresting transform (TZ-IFT-WT) proves to be a link between watershed transforms based on very different paradigms: drop of water, flooding, optimal paths and forest of minimum weight. For all these paradigms and the derived algorithms, it is a challenge to get a unique and thin solution which is consistent with a definition. That is why we propose a unique and consistent thinning of the tie zone. In addition, we demonstrate that the TZ-IFT-WT is also the dual of segmentation methods based on fuzzy connectedness. Thus, the bridge between the morphological and the fuzzy approaches allows to take benefit from the advance of both. As a consequence, the concept of cores of robustness for the seeds is exploited in the case of watersheds.

Keywords: image segmentation, mathematical morphology, watershed, image-foresting transform (IFT), shortest-path forest, minimum spanning forest, graph theory, fuzzy connectedness.

*Aos meus pais Jean-Claude e Anne-Marie,
Às minhas irmãs Laure e Emilie,
À minha avó Edmée*

Agradecimentos

Gostaria de agradecer a todos que, de longe ou de perto, me acompanharam durante esses anos e me ajudaram neste trabalho:

Ao meu orientador, Roberto Lotufo, pelo incentivo, pelas reuniões enriquecedoras, pela lucidez com a qual sempre soube enxergar este trabalho, pela confiança, pelo apoio e pelo exemplo de humildade que deu.

Aos Professores Clésio Luis Tozzi, Jorge Stolfi e Sérgio Shigemi Furuie por terem aceitado participar da banca de defesa. Aos Professores Alexandre Xavier Falcão e Fernando José Von Zuben por terem aceitado ser membros das bancas de defesa e qualificação.

À CAPES, pelo apoio financeiro.

Aos amigos e às amigas do LCA, Adler, André, Carmen, Fábio, Fernanda, Franklin, Gisele, Gustavo, Harlen, Nicola, Pasquini, Rodrigo, Virgínia e os outros, pelo bom humor no laboratório, pela simpatia e pelo acolhimento. Aos amigos e companheiros, André, Jane, Letícia, Mábia (a \LaTeX -woman que me ajudou) e Patrícia, pela companhia e pelas conversas regeneradoras.

À turma do triatlon do “arróxa”, Andrés, Fernanda, João, Mara, Mário, Sakai, Sandra, Serjão, Zé, pelos bons momentos passados nos treinos e pelas risadas! À Fumiko e aos amigos do projeto Raio de Sol, pelos sábados tão animados passados com a criançada. Às amigas helenas, Cristina e Josiane, pelas baladas. Ao amigo Danilo, pelas intermináveis conversas e bons momentos passados juntos. À turma dos chorões forrozeiros, Gabriel, Érica, Guilherme e Marina, pelas noites musicais. A todos vocês, amigos, que foram as veredas no grande sertão do doutorado.

À minha família, pelo apoio, pelo incentivo, pela atenção e pela paciência durante esta jornada, bem além do oceano, lá pelas linhas divisoras de águas...

Sumário

Sumário	xi
Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Siglas e Abreviaturas	xix
Trabalhos Publicados Pelo Autor	xxi
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	4
1.3 Organização do documento	4
1.4 Síntese	5
1.4.1 Zona de empate: Unificação de múltiplas soluções	5
1.4.2 Zona de empate: Elo entre definições de <i>watershed</i>	6
1.4.3 Zona de empate: Elo com as abordagens de conectividade nebulosa	9
1.4.4 Robustez da segmentação e conjunto mínimo de sementes	10
1.4.5 Contribuições originais	11
2 Unificando Soluções Múltiplas via Zona de Empate	13
2.1 Introduction	14
2.2 Definition	16
2.2.1 Overview of the Image Foresting Transform (IFT)	16
2.2.2 The Tie-Zone Watershed (TZWS) transform	17
2.3 Algorithm	17
2.4 Applications	19
2.4.1 Maximal extension of segmented objects	19
2.4.2 Progressive segmentation	19
2.5 Conclusions and Future Work	20
3 Afinamento da Zona de Empate com Solução Única	23
3.1 Introduction	25
3.2 Watershed Transform by Image Foresting Transform	29

3.2.1	Notation and Definitions	30
3.2.2	Algorithms for IFT	31
3.3	Optimal forest paradigm and tie-zone watershed	32
3.3.1	The tie-zone watershed (TZW) transform	33
3.3.2	Watermerging paradigm	36
3.3.3	Multipredecessor Optimal Graph	36
3.4	Thinning of the tie-zone based on label frequency	37
3.4.1	Computing label frequency	39
3.4.2	Computing the map of most frequent labels	42
3.4.3	Dijkstra-IFT-based algorithm for the label-frequency thinning of the tie-zone	45
3.4.4	Illustration	47
3.5	Conclusion	50
4	Transformadas de <i>Watershed</i>	53
4.1	Introduction	54
4.2	The Image Foresting Transform (IFT)	57
4.2.1	Watershed by Image Foresting Transform (IFT-WT)	57
4.2.2	Tie Zone	59
4.2.3	Multipredecessor Optimal Graph and Lower Complete Graph	60
4.3	Watershed based on a Local Condition	61
4.4	Watershed based on Topographic Distance	61
4.4.1	Relationship with Local-Condition Watershed	62
4.4.2	Relationship with Tie-Zone Watershed by IFT	63
4.5	Watershed based on a Minimum Spanning Forest	63
4.6	Conclusion and Future Works	64
5	Dualidade com a Conexidade Nebulosa	67
5.1	Introduction	69
5.2	The Image Foresting Transform	70
5.2.1	Notation and definitions	70
5.2.2	Algorithms	71
5.3	Watershed segmentation under the IFT framework	72
5.3.1	Watershed transforms by IFT	72
5.3.2	Tie-zone watershed	73
5.4	Fuzzy Connectedness Segmentation under the IFT Framework	76
5.4.1	Key ideas on fuzzy connectedness	76
5.4.2	Segmentation by fuzzy connectedness	78
5.4.3	Robustness of the segmented objects	81
5.4.4	Illustration	82
5.5	Conclusion	83

6	Robustez em Relação às Sementes e Conjuntos Mínimos de Sementes	85
6.1	Introduction	88
6.2	Seeded segmentation methods	88
6.2.1	Watershed by Image Foresting Transform	89
6.2.2	Watershed by Minimum Spanning Forest	90
6.2.3	Fuzzy connectedness approach	91
6.3	Robustness regions for seeds	91
6.3.1	Cores for the shortest-path forests and the iterative relative fuzzy object	91
6.3.2	Cores for the minimum spanning forests	92
6.4	Minimal sets of seeds	95
6.4.1	Definitions	96
6.4.2	Relationship with cores and tie-zone	97
6.4.3	Algorithm	98
6.5	Conclusion	101
7	Conclusão	103
	Referências bibliográficas	105
A	Zona de Empate, Gargalos e Análise da Robustez da Segmentação	113
A.1	Introduction	114
A.2	The Tie-Zone Watershed (TZWS)	115
A.2.1	Overview of the Image Foresting Transform (IFT)	115
A.2.2	The Tie-Zone Watershed (TZWS) transform	117
A.2.3	Algorithm	118
A.3	Bottlenecks	119
A.3.1	Watershed vs. watermerging	119
A.3.2	Multiplicity-based Optimal Forests (MOF)	120
A.3.3	Bottleneck characterization	122
A.4	On the robustness of a segmentation	123
A.4.1	Robustness based on tie-zone analysis	123
A.4.2	Robustness based on bottlenecks analysis	125
A.5	Conclusions and future work	131

Lista de Figuras

1.1	Segmentação por watershed ou por conectividade nebulosa: principais paradigmas e interligações entre métodos.	6
2.1	Original images and two possible labeled WS outputs (raster or anti-raster scan) of a line-algorithm or a region-algorithm.	15
2.2	Original grayscale image, TZWS using 4-adjacency, label merging, minimal and maximal extensions, map of number of tied labels, progressive segmentation, watersheds by Vincent and Soille’s algorithm (raster or anti-raster scan), and watersheds by Dijkstra-IFT varying scanning-order.	21
3.1	Data processing order introduces a 1-pixel bias: Input images, WTs according to the definition, and two possible labelled WT outputs (raster or anti-raster scan) of a line-algorithm and a region-algorithm.	27
3.2	Multiple solutions do not contradict the definition but are consequences of specific implementations, in the cases of the WT by IFT and the topological watershed.	27
3.3	Solution given by the strict definition of WT by immersion: watershed line is not separating. Possible watershed lines by Vincent and Soille’s algorithm: Multiple solutions are not predicted by the definition but are consequences of the bias introduced by implementation.	28
3.4	Lexicographic cost avoids lower completion of the image	32
3.5	TZW applied on image I_3 of Figure 3.2 using 4-adjacency, and the label merging result.	33
3.6	Watermerging paradigm. On the left (right), the arrows point the 2 minima (maxima) of the topography. The triple line and the grey zone constitute the tie-zone where waters merged.	36
3.7	Difference between frequency of forests (or predecessor arcs) and frequency of linking paths.	40
3.8	Drop of water is sliding down from a TZ pixel and broken up in equal parts whenever it meets an intersection of descending paths.	42
3.9	Computing the most frequent label: Possible predecessors, label frequencies computed according to the recursive formula, label and predecessor assigned when possible, isolated (ISO) pixel case and equal frequency (EF) case.	43
3.10	Input greyscale image, Multipredecessor Optimal Graph, TZW, frequency of label assignments, resulting label map L and one of its supports P . Observe the cases of equal frequency (EF) and isolated pixel (ISO).	44

3.11	Gradient image of an airplane, TZW, frequency-based thinning, and detail of the frequency-based thinning showing optimal trees in CBs, EF (white with dot) and ISO (white with black square) pixels.	49
3.12	WT corresponding to a random optimal forest (one of the solution of WT by IFT). . .	50
3.13	WT by Vincent-Soille's algorithm for four different image scanings.	51
4.1	Lower-complete input grayscale image, some WTs according to definitions from literature, lower complete graph and multipredecessor optimal graph.	56
4.2	A weighted graph with two markers (\circ) and its 3 possible SPF-max and 2 MSF (total weight = 3). SPF 3 is not a MSF (total weight = 4).	63
5.1	Images and object extractions.	82
6.1	Núcleos, bacias e zona de empate de um grafo ponderado e suas sementes: As sementes não alteram a segmentação quando ficam nos seus núcleos.	86
6.2	A weighted graph with two seeds (\circ) and its 3 possible SPF-max and 2 MSFs. . . .	91
6.3	Non-robustness of the MSF when the seed (\circ) is moved within a tree.	92
6.4	Non-robustness of the set of MSFs when the seed (\circ) is moved within a tree.	93
6.5	Extreme pass and mute arcs in cycle σ^{cd}	94
A.1	Original images and two possible labeled WS outputs (raster or anti-raster scan) of a line-algorithm or a region-algorithm.	114
A.2	Input grayscale image, bottlenecks, TZWS, label merging, MOF and non-MOF, map of multiplicity, watersheds by Dijkstra-IFT varying scanning-order, watersheds by Vincent and Soille's algorithm (raster or anti-raster scan).	116
A.3	Watermerging paradigm. On the left (right), the arrows point the 2 minima (maxima) of the topography. A triple line (of constant altitude) shows the bottlenecks locus and constitutes with the gray zone the tie-zone where waters merged. Only the front bottleneck in contact with the gray zone (its influence zone) is non-trivial.	121
A.4	<i>Airplane</i> image: Watershed lines (by Vincent-Soille's algorithm) of the gradient (a) and the filtered gradient (b)(c) (with $h = 10; 35$ resp.). (d)–(f) Respective TZ in white.	124
A.5	R_2 in function of h for <i>weavetile</i> , <i>tree-leaf</i> , <i>mirage</i> , <i>airplane</i> , <i>lena</i> , <i>MR heart</i> , <i>XR particles</i> and <i>leaf-grass</i> images. $R_2 < 1\%$ for $h \geq 0, 3, 3, 8, 8, 14, 15, 37$ respectively.	126
A.6	(a) Input <i>mirage</i> gradient image. (b)–(d) TZWS (white) after filtering the gradient minima with $h = 0; 5; 10$ (116, 13, 4 min.) resp.	127
A.7	(a) Original <i>leaf-grass</i> image. (b)–(d) TZWS (white) after filtering the gradient minima with $h = 10; 35; 50$ (2936, 411, 123 min.) resp.	128
A.8	Pseudo-cone. (a) Input image with 24 minima at 0-level (4-adjacency) and 101 bottlenecks (in bold). (b) TZWS (white). (c)–(d) WS by Dijkstra's and Vincent-Soille's algorithms (raster-scan order).	129
A.9	LM algorithm on <i>mirage</i> (detail): bottlenecks (white contour) and IZs (shaded). . . .	130
A.10	(a) Cumulated weighted histogram. (b) Quartiles for 3 segmented images varying h : w-ranges of the BNs that contribute for each quarter of the total TZ area.	132

Lista de Tabelas

3.1	Table of abbreviations	25
4.1	Characteristics of the main watershed transform (WT) definitions.	57
5.1	Duality in concepts and segmentation methods.	78
6.1	Table of abbreviations	87

Lista de Siglas e Abreviaturas

FC	- Fuzzy connectedness	Conexidade nebulosa
Flood	- Flooding	Inundação
FOE	- Fuzzy object extraction	Extração de objetos nebulosos
IFT	- Image foresting transform	Transformada imagem floresta
IRFOE	- Iterative relative fuzzy object extraction	Extração iterativa de objetos nebulosos relativos
LC	- Local condition	Condição local
LM	- Label merging	Fusão de rótulos
lex	- Use of lexicographic path-cost	Uso do custo de caminho lexicográfico
MOFS	- Multiple object fuzzy segmentation	Segmentação nebulosa de objetos múltiplos
MSF	- Minimum spanning forest	Floresta geradora mínima
MSS	- Minimal seed set	Conjuntos mínimos de sementes
NRRR	- Non-redundant receptive region	Regiões receptivas não redundantes
RFOE	- Relative fuzzy object extraction	Extração de objetos nebulosos relativos
SPF	- Shortest path forest	Floresta de caminhos mínimos
SPF-max	- Shortest path forest using the max-arc path cost	Floresta de caminhos mínimos utilizando o custo de caminho do arco máximo
TD	- Topographical distance	Distância topográfica
Topol	- Topological	Topológico
TZ	- Tie zone	Zona de empate
WT	- Watershed transform	Transformada das linhas divisoras de águas

Trabalhos Publicados Pelo Autor

1. R. Audigier, R. Lotufo, and M. Couprie. “The tie-zone watershed: Definition, algorithm and applications.” In *Proceedings of IEEE International Conference on Image Processing (ICIP’05)*, volume 2, pages 654–657, Genova, Italy, September 2005.
2. R. Audigier and R. Lotufo. “Tie-zone watershed, bottlenecks and segmentation robustness analysis.” In *XVIII Brazilian Symp. on Computer Graph. and Image Proc. (SIBGRAPI’05)*, pages 55–62, Natal, Brazil, October 2005. IEEE Press.
3. R. Audigier and R. Lotufo. “Duality between the watershed by image foresting transform and the fuzzy connectedness segmentation approaches.” In *XIX Brazilian Symp. on Computer Graph. and Image Proc. (SIBGRAPI’06)*, pages 53–60, Manaus, Brazil, October 2006. IEEE Press.
4. R. Audigier and R. Lotufo. “Uniquely-determined thinning of tie-zone watershed based on label frequency.” *Journal of Mathematical Imaging and Vision*, 27(2):157–173, February 2007.
5. R. Audigier and R. Lotufo. “Watershed by image foresting transform, tie-zone, and theoretical relationships with other watershed definitions.” In *8th International Symposium on Mathematical Morphology (ISMM’07)*, Rio de Janeiro, Brazil, October 2007. Accepted.
6. R. Audigier and R. Lotufo. “Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches.” In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI’07)*, Belo Horizonte, Brazil, October 2007. IEEE Press. Accepted.

Capítulo 1

Introdução

1.1 Motivação

O uso de imagens digitais tem se tornado cada vez mais freqüente em numerosos domínios de aplicação. Muitas vezes, elas não são utilizadas diretamente e requerem, portanto, algum processamento. A segmentação é certamente um dos problemas cruciais nas diferentes etapas do processamento digital de imagens. Ela permite particionar uma imagem em elementos, ou segmentos, que são susceptíveis de representar regiões com significados diferentes [GW93]: geralmente, objetos de interesse a serem separados ou destacados do fundo.

Assim, a comunidade de processamento digital de imagens tem dedicado muitos esforços para resolver o problema de segmentação. Muitas abordagens de segmentação foram propostas [FM81, HG85, PP93], entre as quais podemos citar por exemplo: a detecção de bordas (*edge detection*) [Can86], a minimização de energia como os contornos ativos (*active contour*) [KWT88] e os modelos de balões [Coh91], o corte de grafos (*graph cut*) [SM00], a limiarização (*thresholding*) [Ots79, SSWC88], as agregações (*clustering*) [Pap92], o crescimento, a divisão ou a fusão de regiões (*region growing, splitting* ou *merging*) [AB94], a linha divisora de águas (*watershed*) [BM93] e a conectividade nebulosa (*fuzzy connectedness*) [US96]. As duas últimas são estudadas nesta tese.

A transformação de *watershed* é uma ferramenta poderosa de segmentação que foi proposta no quadro da morfologia matemática no final dos anos 70 [BL79]. Desde então, vários paradigmas têm sido usados para calculá-la: da simulação de gotas d'água caindo sobre um relevo representando a imagem [BBM⁺97], à simulação da inundação desse relevo [VS91], passando pelo cálculo de floresta de peso mínimo a partir de um grafo modelando a imagem [Mey94a]. Existem, portanto, várias definições da transformação de *watershed* que geralmente resultam em soluções diferentes. Algumas delas também permitem soluções múltiplas, logo, segmentações diferentes (cf. Figuras 3.2 e 4.1(c)–(h)).

Além disso, muitos algoritmos de *watershed* são presentes na literatura, e nem sempre correspondem a uma definição. Neste caso, a segmentação retornada ainda pode depender da implementação escolhida (cf. Figura 3.3). Diante dessa variedade, às vezes confusa, de definições e algoritmos, escolher um método e sua implementação pode se revelar arbitrário e a segmentação obtida pode ser apenas o resultado desta escolha arbitrária. Uma revisão bibliográfica parcial desses métodos e algoritmos foi apresentada em [RM00] sem, entretanto, oferecer uma análise comparativa profunda que ajude o usuário a prever e entender as diferenças das segmentações retornadas por esses métodos e suas implementações. Assim, as diferenças entre as segmentações de uma imagem antes e depois de uma rotação poderiam deixá-lo perplexo, sabendo que a mesma implementação do mesmo algoritmo foi usada.

Outra particularidade desses métodos é que alguns retornam linhas de *watershed*, isto é, alguns pixels não pertencem a nenhum segmento, mas representam a fronteira entre segmentos. Para certas definições e certos algoritmos, essa fronteira pode ser relativamente espessa (vários pixels de espessura como mostram as Figuras 2.2(i)–(j)). Em aplicações de segmentação, linhas espessas são indesejáveis.

Resumindo, a transformada de *watershed*, apesar de ser uma boa ferramenta de segmentação, sofre de vários problemas:

- Possui várias definições não unificadas e sem relações entre si.
- Possui muitos algoritmos nem sempre consistentes com uma definição.
- Certas definições têm múltiplas soluções.
- Nenhum dos numerosos algoritmos, seja qual for sua implementação, garante a mesma solução quando rotações ou espelhamentos são aplicados na imagem, ou seja, todos os algoritmos podem retornar múltiplas soluções que dependem da ordem de processamento.
- Algumas definições e alguns algoritmos podem retornar fronteiras espessas entre os objetos de interesse e o fundo.

A transformação imagem-floresta (IFT) consiste em construir uma floresta de caminhos ótimos a partir de um grafo ponderado e sementes [FSL04]. Neste quadro genérico, dependendo da modelagem da imagem em um grafo ponderado, da função de custo de caminho e das sementes escolhidas, pode-se obter uma transformação de *watershed*: a transformação de *watershed* por IFT (IFT-WT). A IFT oferece, portanto, um quadro consistente para a definição de uma transformação de *watershed* e a implementação de algoritmos fiéis a sua definição. Entretanto, ela tampouco define uma única transformada.

A segunda família de métodos de segmentação estudados nesta tese baseia-se no conceito de conexidade nebulosa (*fuzzy connectedness*) [USL02, HC01]. A idéia da conexidade nebulosa é derivada da teoria dos conjuntos nebulosos em que funções de pertinência decidem do grau com o qual um elemento pertence a um objeto. A conexidade nebulosa representa a força de coesão entre os elementos de um objeto. Na prática, essas técnicas avaliam a similaridade entre pixels e agrupam os pixels mais fortemente ligados para retornar regiões homogêneas. Assim, cada segmento é representado por um conjunto nebuloso de elementos associados com um certo grau de pertinência. Em geral, esses métodos necessitam sementes representativas de cada objeto para retornar os conjuntos de pixels que mais se parecem ou se conectam com essas sementes.

Portanto, percebe-se que há uma semelhança entre os métodos baseados em *watershed* e os baseados em conexidade nebulosa. Todos utilizam sementes e tentam agrupar pixels em um mesmo segmento quando a dissimilaridade entre eles é mínima ou, respectivamente, a similaridade é máxima. Isto constitui uma motivação para analisar com detalhes quais são as diferenças entre esses métodos e até qual ponto eles são similares.

1.2 Objetivos

Os objetivos desta tese são:

- Analisar as definições de *watershed* e suas inter-relações, ver quais são as diferenças conceituais e práticas, e determinar onde elas ocorrem.
- Definir uma transformada de *watershed* com solução fina e única, e determinar um algoritmo que retorna uma solução consistente com esta definição. Para tanto, introduzimos o conceito de zona de empate para unificar as múltiplas soluções relativas a um método específico.
- Entender o que pode aproximar as abordagens de *watershed* daquelas baseadas em conectividade nebulosa.
- Investigar, enfim, como se comportam as segmentações definidas por esses métodos quando as sementes de entrada variam; ou, ao contrário, como estas devem ser definidas para se obter a mesma segmentação.

1.3 Organização do documento

O corpo desta tese é composto por cinco artigos em inglês publicados em congressos ou revista. Como eles podem ser lidos independentemente, alguns conceitos e definições são inevitavelmente repetidos. As referências bibliográficas, porém, aparecem agrupadas no final da tese. A numeração dos algoritmos propostos, das seções e figuras dos artigos foi unificada.

Um sexto artigo publicado busca entender a origem da zona de empate, ou seja, a zona de desacordo entre as múltiplas soluções de *watershed*. Ele mostra que a presença de zona de empate não depende da presença de platôs na imagem e aponta para os gargalos, pontos de onde se origina a zona de empate. Porém, esse artigo acabou não sendo tão importante quanto esperado para as linhas de raciocínio da tese e, por isso, foi colocado em anexo.

Os resultados de nossas pesquisas foram publicados à medida que eles foram sendo descobertos. Assim, alguns podem parecer espalhados em artigos diferentes apesar de fortemente conexos. Para agregar tais resultados segundo a linha diretriz fixada pelos objetivos acima, a seção seguinte apresenta uma síntese das contribuições desta tese. Nesta descrição simples da tese, detalhes são omitidos. No entanto, são feitas referências a figuras, algoritmos e equações presentes no corpo da tese. Além disso, para ajudar a leitura dos artigos, cada capítulo contém um resumo introdutório das contribuições originais do artigo apresentado em seguida.

1.4 Síntese

1.4.1 Zona de empate: Unificação de múltiplas soluções

Começamos por observar que a transformação de *watershed* via IFT (IFT-WT na Figura 1.1), bastante estudada [FSL04, FCL01] e utilizada em segmentação de imagens [LF00, ALF04, ALF06], tem várias soluções consistentes, isto é, várias florestas de caminhos mínimos. Dependendo do algoritmo utilizado e da implementação, é possível obter segmentações diferentes (cf. Figuras 2.2(k)–(n)). Por isso, introduzimos o novo conceito de zona de empate [ALC05]. Considerando-se o conjunto de soluções possíveis, a zona de empate é a região onde as soluções diferem, i.e., a região que pode ser atribuída a diferentes segmentos (cf. Seção 2.2.2). As regiões que sempre pertencem aos mesmos segmentos são chamadas de bacias de retenção (*catchment basin*, CB, cf. Figura 2.2(b) e Equação 4.3). Portanto, a transformada em zona de empate unifica as soluções possíveis: É uma solução única e consistente com sua definição. Uma variante da transformada em zona de empate consiste em especificar para cada pixel quais são os rótulos dos segmentos empatados. Ela contém, portanto, mais informação sobre a zona de empate (cf. Seção 3.3.2 e Figura). Ela é batizada “fusão de rótulos” (*label merging*, LM, cf. Figura 2.2(c)).

Nota-se que os caminhos mínimos até cada nó do grafo são determinados a partir da minimização do custo de caminho. No caso da IFT-WT, utiliza-se o arco de peso máximo (*max-arc*) do caminho para calcular seu custo associado (cf. Equação 4.1). Um custo lexicográfico de dois componentes pode também ser utilizado [LF00]. Além de utilizar o custo anterior (*max-arc*) como componente de peso mais significativo, um segundo componente desempata o primeiro: Para um caminho levando a um nó, ele corresponde ao número de nós neste caminho que têm o mesmo componente mais significativo que este nó (cf. Equação 4.2). Intuitivamente, diz-se que o custo lexicográfico desempata os “platôs” formados por nós de mesmo custo *max-arc*. Na prática, quando se usa uma relação de adjacência simétrica, este custo tem efeito de particionar eqüitativamente qualquer platô entre os diferentes segmentos candidatos a sua incorporação, caso apenas o primeiro componente do custo fosse usado. A notação “IFT-WT-lex” designa esse caso específico de transformação de *watershed* por IFT.

O uso do custo lexicográfico restringe o número de soluções da IFT-WT, embora não as reduza a uma única solução. Conseqüentemente, a zona de empate correspondente (denominada TZ-IFT-WT-lex na Figura 1.1) será relativamente menos espessa. No entanto, ela pode continuar espessa (cf. Figuras 2.2(b) e 3.11(b)). Quando o objetivo da segmentação é atribuir todos os pixels a algum segmento, não é desejável ter uma zona de empate espessa. Portanto, propomos um afinamento dessa zona de empate [AL07b] que conserva a unicidade da solução, mas que dá prioridade ao segmento que mais freqüentemente incorpora a zona em questão (Equações 3.5 e 3.6–3.9).

Apresentamos algoritmos eficientes para o cálculo direto da TZ-IFT-WT com custo lexicográfico [ALC05] (cf. Algoritmo 1, Seção 2.3) ou sem [AL06] (cf. Algoritmo 3, Seção 5.3.2) e também para o afinamento da TZ-IFT-WT-lex [AL07b] (Cf. Algoritmo 2, Seção 3.4.3). Nota-se que os pixels da imagem são processados apenas uma vez com o uso do custo lexicográfico, e duas vezes senão. O afinamento é feito simultaneamente.

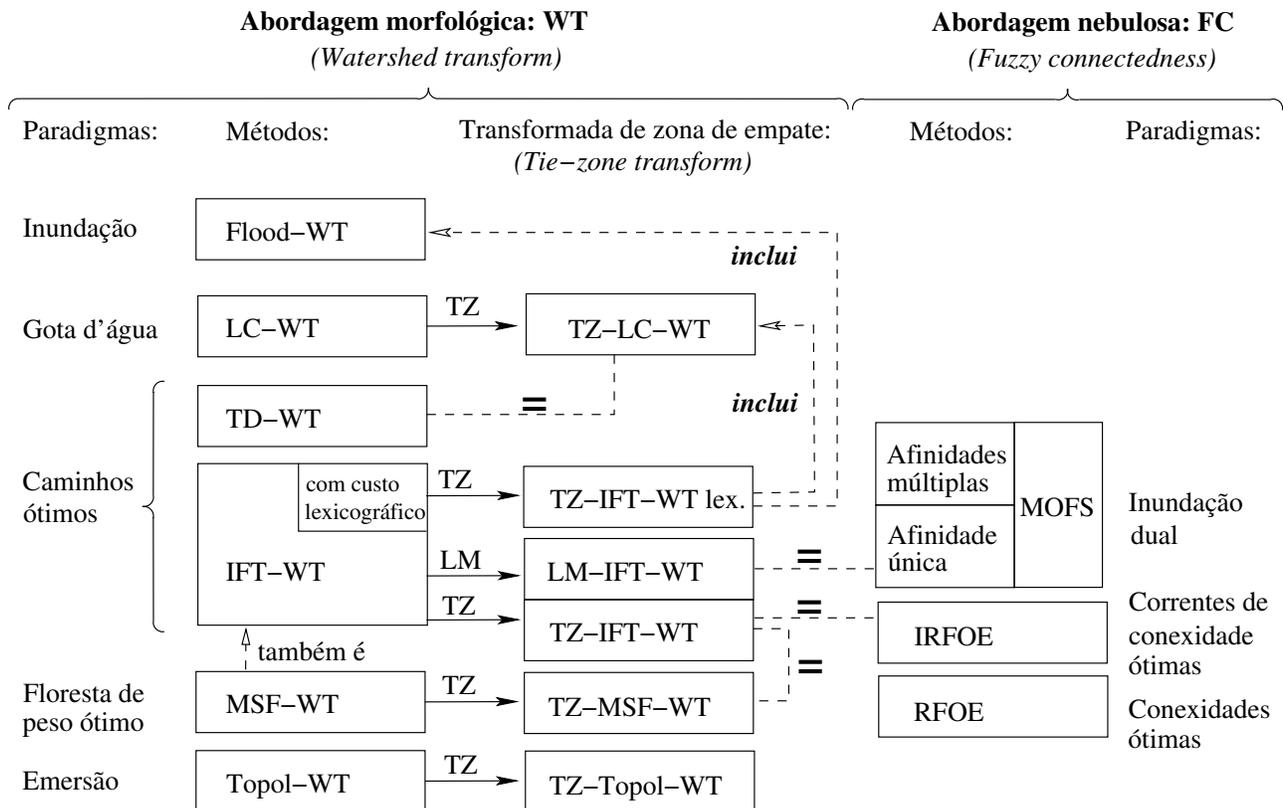


Fig. 1.1: Segmentação por watershed ou por conectividade nebulosa: principais paradigmas e interligações entre métodos.

1.4.2 Zona de empate: Elo entre definições de *watershed*

Estudamos em seguida as diferentes definições da transformada de *watershed* na literatura [AL07c]. Como mostra a Figura 1.1, podemos classificá-las em cinco paradigmas diferentes: rastreamento das *gotas d'água* que caem na superfície representando a imagem, *inundação* da superfície perfurada nos fundos dos vales, *emersão* da superfície previamente imersa, *caminhos ótimos* e *floresta de peso mínimo*.

Gota d'água

O paradigma da gota d'água é aquele que se aproxima mais do fenômeno físico conhecido em geociência e que inspirou a transformada de *watershed*. Quando uma gota d'água cai num ponto de uma superfície topográfica, ela desliza nesta superfície e desce até atingir um vale (chamado de mínimo regional da imagem). Considera-se a imagem, geralmente seu gradiente, como uma superfície topográfica onde os níveis de cinza correspondem à altitude. O lugar dos pontos onde essa gota deslizaria até o mesmo vale corresponde à bacia hidrográfica (ou bacia de retenção) deste vale. As linhas divisoras de água (*watershed*) delimitam as diferentes bacias.

Seguindo esse paradigma da gota d'água, a transformada de *watershed* baseada em condições locais (*local condition watershed transform*, LC-WT, cf. Definição 1) é obtida por indicar localmente (portanto, independentemente) em qual pixel vizinho a gota deslizaria e depois rastrear os possíveis caminhos mais íngremes das gotas até os vales para formar as bacias [RM00, LTHS06, RLLV07]. Esse método é particularmente propício a algoritmos paralelos [BBM⁺97, BM98, MR98]. Nota-se que pode oferecer soluções múltiplas, nas quais todos os pixels pertencem a bacias, as linhas divisoras ficando implícitamente entre os pixels (cf. Figuras 4.1(e)–(f)). Observa-se que o problema de se saber aonde a gota d'água escorre quando chega a um platô é resolvido pela operação de declivamento (*lower completion*) aplicada à imagem em pré-processamento. Ela cria um declívio nos platôs, de modo que todo pixel que não esteja no fundo de um vale (mínimo regional) tenha um vizinho de altitude mais baixa (cf. Figura 3.4).

Inundação

O paradigma da inundação procede diferentemente. Os mínimos regionais são perfurados, e a superfície é mergulhada gradativamente na água. A água invade os vales, de modo que a extensão das bacias aumenta progressivamente. Quando duas bacias estão prestes a se fundir, uma barragem é construída para evitar a mistura das bacias. A solução definida pelo método baseado em inundação (*flooding watershed transform*, Flood-WT) é única [VS91]. As linhas divisoras podem coincidir com pixels e/ou estar entre pixels (cf. Figuras 4.1(b) e 3.3(b)). Observa-se que o famoso algoritmo correspondente [VS91] não segue exatamente a definição proposta, mas retorna múltiplas soluções com linhas sempre localizadas em pixels e separadoras (cf. Figura 3.3(c)–(d)). Em outras palavras, duas bacias quaisquer não possuem nenhum contato direto.

Emersão

A transformada de *watershed* topológica (*topological watershed transform*, Topol-WT) é vista como um afinamento homotópico que escava os vales da superfície e erode parte dos montes, en-

quanto preserva a saliência (altura mínima da barreira montanhosa) entre os vales [CB97a]. Essa transformada pode ser calculada pelo paradigma de emersão [NCB05]. Ao contrário do paradigma de inundação, observam-se os lagos que se formam quando se retira da água a superfície previamente imersa. Também neste caso, existem várias soluções que satisfazem a definição desta transformada. As linhas divisoras são sempre explícitas, isto é, constituídas de pixels (cf. Figuras 4.1(c)–(d) e 3.2(d)–(f)).

Caminhos ótimos

A transformada de *watershed* também pode ser considerada como um problema de otimização de caminhos. A imagem é vista como um grafo onde os pixels constituem os nós. No método baseado em distância topográfica (*topographic-distance watershed transform*, TD-WT)[Mey94b], um pixel pertence a uma bacia se sua distância topográfica ao mínimo regional correspondente é estritamente menor às distâncias dos outros mínimos regionais (cf. Definição 2). Concretamente, essa distância é definida de tal modo que os caminhos mínimos correspondem aos caminhos descendentes mais íngremes. Quando vários caminhos mais íngremes levam de um pixel até pelo menos dois mínimos regionais diferentes, este constitui a linha divisora de água. Observa-se, no entanto, que a linha divisora não passa necessariamente (só) por pixels (cf. Figura 4.1(i)). A solução assim definida é única.

Os métodos IFT-WT e IFT-WT-lex apresentados acima também seguem o paradigma dos caminhos ótimos. Apenas a definição de distância muda, e é chamada de função de custo de caminho. A floresta de caminhos mínimos retornada é composta de árvores representando as bacias e tendo como sementes os mínimos regionais da imagem. Na verdade, essas árvores são constituídas de caminhos descendentes, não necessariamente dos mais íngremes.

Floresta de peso mínimo

O último paradigma que se encontra na literatura [Mey94a] para descrever a transformada de *watershed* é a floresta de peso mínimo (*minimum spanning forest*, MSF). Novamente, a imagem é modelada por um grafo, e as bacias por árvores. Mas desta vez, o peso total da floresta, ou seja, a soma de seus arcos é minimizada (cf. Figuras 4.2(a)–(b)).

Ligações entre transformadas de *watershed*

Embora os métodos de *watershed* apresentados sigam paradigmas diferentes e geralmente dêem segmentações diferentes da mesma imagem, conseguimos relacioná-las graças à transformada em zona de empate (vide a Figura 1.1). Assim, a transformada em zona de empate das múltiplas soluções

da LC-WT é equivalente à TD-WT (cf. Seção 4.4.1). Mostramos também que toda MSF também é uma IFT-WT, mas o contrário é falso (cf. Teorema 2). Entretanto, as respectivas transformadas em zona de empate são idênticas (cf. Teorema 3).

Por outro lado, a zona de empate da TZ-IFT-WT-lex inclui a zona de empate da TZ-LC-WT, e cada bacia da TZ-LC-WT inclui uma bacia de TZ-IFT-WT-lex (cf. Seção 4.4.2). As mesmas relações de inclusão também existem entre a TZ-IFT-WT-lex e a Flood-WT.

Para uma aplicação onde a confiança na segmentação obtida é crucial, diferenças devidas aos métodos escolhidos ou a suas implementações não podem ser aceitáveis. Portanto, a utilização da TZ-IFT-WT revela-se particularmente interessante. Com efeito, ela retorna com total segurança as extensões mínimas das bacias (segmentos) que todos esses métodos de *watershed* têm em comum, a zona de empate sendo partida e atribuída a um ou outro segmento por arbitrariedade do método escolhido [AL05].

1.4.3 Zona de empate: Elo com as abordagens de conectividade nebulosa

Conexidades ótimas e correntes de conectividade ótima

A zona de empate também permite relacionar a abordagem morfológica da transformada de *watershed* com a abordagem da conectividade nebulosa (*fuzzy connectedness*, FC) como mostra a Figura 1.1. Os objetos nebulosos extraídos são formados pelos pixels mais fortemente conexos a suas sementes respectivas. Para tanto, uma afinidade define o grau de semelhança, ou afinidade, entre dois pixels adjacentes.

No caso da extração de objetos nebulosos relativos (*relative fuzzy object extraction*, RFOE), um objeto é constituído pelos pixels com força de conectividade com sua semente estritamente maior do que qualquer conectividade com outras sementes (cf. Equação 5.8).

No caso da extração iterativa de objetos nebulosos relativos (*iterative relative fuzzy object extraction*, IRFOE), um objeto é constituído pelos pixels ligados a sua semente por correntes de força de conectividade estritamente maiores do que com outras sementes, sendo que essas correntes devem pertencer inteiramente ao mesmo objeto (cf. Figura 5.1). Demonstramos que esse método é exatamente o dual da TZ-IFT-WT, portanto, retorna a mesma segmentação [AL06]. Isto também significa que a IRFOE pode ser vista como a transformada em zona de empate das florestas de peso mínimo (TZ-MSF-WT) [AL07a].

Inundação dual

Existe outro paradigma da abordagem nebulosa adotado pelo método de segmentação nebulosa de objetos múltiplos (*multiple object fuzzy segmentation*, MOFS). Esta segmentação [HC01, CHK05]

pode atribuir um mesmo pixel a vários segmentos, à maneira da fusão de rótulos (*label merging*). O método procede por inundação dual gradativa (um tipo de emersão). Propagam-se os rótulos de um pixel processado no nível de inundação anterior em um pixel do nível corrente de inundação dual, sob condição desses dois terem maior força de conexidade (não estrita), em relação a outras possíveis propagações.

Quando a afinidade entre dois pixels particulares é fixa, o método de MOFS é dual de LM-IFT-WT, portanto, as segmentações correspondentes são idênticas.

Entretanto, a MOFS também permite a definição de várias afinidades entre dois pixels particulares: uma por objeto. Neste caso, o cálculo da força de conexidade depende do objeto cujo rótulo se quer propagar. A MOFS com múltiplas afinidades não tem mais critério de otimalidade global, mas apenas otimiza as forças de conexidade a cada nível de inundação. A dualidade com a LM-IFT-WT não é mais válida.

1.4.4 Robustez da segmentação e conjunto mínimo de sementes

Uma vez que os resultados da IFT, da MSF, dos métodos derivados, e dos métodos por conexidade nebulosa dependem das sementes de entrada, pode-se perguntar o quanto eles dependem delas. Definimos o núcleo de robustez (*core*) associado a uma semente como a região onde a semente pode se deslocar sem alterar a segmentação resultante (cf. Figura 6.1). Determinamos esses núcleos no caso da TZ-IFT-WT (cf. Equações 6.4–6.5) e da TZ-MSF-WT (cf. Seção 6.3.2) por transposição dual do objeto nebuloso relativo (*relative fuzzy object*, RFO, cf. Equação 6.3).

A determinação dos núcleos oferece informações valiosas. Núcleos com área grande podem avisar o usuário que as sementes manualmente escolhidas irão gerar uma segmentação muito estável, mesmo que a escolha de sementes sofra variações intra- e inter-usuários. Núcleos com pouca extensão significam que um leve deslocamento das sementes irão modificar a segmentação. Além disso, as características dos núcleos (área, distribuição) podem servir de realimentação num processo de detecção automática de sementes.

Tendo em mão definições consistentes, num quadro unificado, de entidades que ajudam a entender diversos métodos de segmentação, podem-se enxergar melhor as relações com o problema dos conjuntos mínimos de sementes. Este problema inversa da segmentação [LS02] visa determinar os conjuntos mínimos de sementes a serem escolhidas para se obter uma segmentação dada. Foram definidas as “regiões receptivas não redundantes”, onde pelo menos uma semente deve ser escolhida [LS02] para recobrir a segmentação de entrada. Mais sementes podem ser escolhidas em tais regiões ou até mesmo em “regiões receptivas redundantes” e em “zonas mortas”, desde que elas tenham o rótulo adequado (cf. Seção 6.4.1). Mostramos na Seção 6.4.2 que quando exatamente uma semente é escolhida por região receptiva não redundante, essa região corresponde ao núcleo de robustez definido

acima.

Enfim, propomos na Seção 6.4.3 o Algoritmo 4 para o cálculo dos conjuntos mínimos de sementes aplicável ao nível do pixel e não somente das bacias primitivas (isto é, as bacias de uma transformada de *watershed* prévia), como era o caso do algoritmo proposto em [Sil01].

1.4.5 Contribuições originais

Esta tese tem várias contribuições originais de natureza teórica principalmente. A pesquisa fundamental que foi desenvolvida sobre a transformada de *watershed* ajuda na compreensão da diversidade até agora muito confusa dos métodos e algoritmos presentes na literatura, e mostra suas diferenças sutis. Embora teóricos, os resultados desta tese abrem novas portas para futuras aplicações práticas. Resumindo, as contribuições desta tese são as seguintes:

1. Definimos um novo paradigma no Capítulo 2: a zona de empate.
2. Introduzimos novos métodos e seus algoritmos nos Capítulos 2, 3 e 5:
 - TZ-IFT-WT-lex. (Algoritmo 1),
 - seu afinamento baseado na frequência dos rótulos (Algoritmo 2),
 - TZ-IFT-WT (Algoritmo 3).
3. Provamos nos Capítulos 4–6 as ligações entre os métodos:
 - TD-WT e TZ-LC-WT (equivalência),
 - TZ-IFT-WT-lex e TZ-LC-WT (inclusão),
 - TZ-IFT-WT-lex e TZ-flood-WT (inclusão),
 - TZ-IFT-WT e TZ-MSF-WT (equivalência),
 - TZ-IFT-WT e IRFOE (equivalência),
 - LM-IFT-WT e MOFS com afinidade única (equivalência).
4. Transpomos o conceito de núcleos (*cores*), regiões de robustez para as sementes, nos métodos:
 - TZ-IFT-WT (Capítulo 5) e
 - TZ-MSF-WT (Capítulo 6).
5. Demonstramos a ligação entre os núcleos de robustez e as regiões receptivas não redundantes no problema dos conjuntos mínimos de sementes (Capítulo 6).

6. Propomos um algoritmo para o cálculo dos conjuntos mínimos de sementes (Algoritmo 4, Capítulo 6).

Capítulo 2

Unificando Soluções Múltiplas via Zona de Empate

Este capítulo contém o artigo intitulado *The Tie-Zone Watershed: Definition, Algorithm and Applications* [ALC05]. Ele apresenta a transformada em zona de empate (*tie zone*, TZ) no caso específico do *watershed* via transformada imagem-floresta (IFT-WT). Esta nova transformada deriva do conjunto de soluções da IFT-WT, que são florestas de caminhos mínimos. Mais exatamente, é o consenso entre elas: as partes segmentadas da mesma maneira por todas essas florestas constituem as bacias de retenção, enquanto as partes onde há litígio entre as florestas não são atribuídas a nenhum segmento e constituem a zona de empate.

Essa transformada é dita de região, pois não há linha divisora explícita (i.e., formada por pixels) entre as bacias de retenção: as bacias podem simplesmente ser adjacentes, como ser separadas pela zona de empate. Além de obedecer a um critério de otimalidade por ser baseada em IFT, a solução retornada é única. Portanto, não depende da implementação escolhida.

Propomos um algoritmo eficiente baseado no algoritmo dos caminhos mínimos de Dijkstra, onde cada pixel da imagem é processado apenas uma vez. Este utiliza um custo lexicográfico para desempatar parcialmente os caminhos de mesmo custo, em função da distância ao longo da qual o custo de caminho foi constante.

Introduz-se uma variante da transformada em zona de empate: a fusão de rótulos (*label merging*, LM). Em vez de atribuir um rótulo especial mas único para cada pixel da zona de empate, guardam-se os rótulos que cada solução da IFT-WT atribuiria a este pixel. Assim a rotulação final atribui para cada pixel um conjunto de rótulos. Duas aplicações utilizando a fusão de rótulos são propostas.

Primeiro, podem ser extraídos limites confiáveis da extensão de cada segmentos. As bacias de retenção (regiões onde é atribuído apenas um rótulo por pixel) constituem as extensões mínimas dos segmentos. A extensão máxima de um segmento referente a um rótulo específico é obtida

acrescentando-se à extensão mínima todos os pixels da zona de empate cujo conjunto de rótulos fusionados contém este rótulo. Extensões mínimas e máximas permitem ter intervalos confiáveis para as medidas feitas em cada segmento. Extensões mínimas também podem guiar o usuário num processo de segmentação interativa incentivando-o a refinar a segmentação caso estas não corresponderem a segmentações aceitáveis.

Segundo, como a zona de empate pode ser espessa às vezes, propomos um afinamento progressivo. Após uma primeira TZ-IFT-WT, a zona de empate é substituída por uma imagem, cujos níveis de cinza correspondem ao número de rótulos fusionados em cada pixel. Aplica-se novamente a TZ-IFT-WT nesta imagem com sementes nas bacias de retenção da segmentação anterior. Repete-se o processo de substituição e segmentação até estabilização da segmentação.

Abstract

In this work, a new type of watershed transform is introduced: the Tie-Zone WaterShed (TZWS). This region-based watershed transform does not depend on arbitrary implementation and provides a unique and optimal solution. Indeed, many solutions are sometimes possible when segmenting an image with a watershed algorithm. In this case, the TZWS assigns each pixel to a catchment basin (CB) if in all solutions it belongs to this CB. Otherwise, the pixel is said to belong to a tie-zone (TZ). We propose an efficient algorithm based on Image Foresting Transform (IFT) which computes the TZWS transform as a shortest-path forest. Finally, two applications of this TZWS are presented: bounding intervals for segmented objects' extensions and a progressive segmentation procedure.

2.1 Introduction

The watershed (WS) transform is a well-known and powerful segmentation tool for morphological image processing. It was first introduced by Beucher and Lantuéjoul [BL79] for contour detection and applied in image segmentation by Beucher and Meyer [MB90]. Nowadays, there are many definitions and algorithms of watershed transforms in literature. Roerdink and Meijster [RM00] give a comparison of some of them. The algorithm of Vincent and Soille [VS91] is based on immersion simulation: the image is represented by a topography inundated by water that springs from regional minima. The watershed lines are dams constructed for separating the growing catchment basins (CB) corresponding to minima. The algorithm of Meyer [Mey94b] computes the WS transform by solving a shortest paths problem with respect to a topographical distance function.

In the numerous WS algorithms, variations may first occur in the input: all regional minima; only imposed minima to avoid oversegmentation (WS from markers [MB90]); or grayscale mark-

ers [LFZ02] to specify the depth (handicap) of some imposed minima. Then, the output may be of different types: ‘line-algorithms’ return separating WS lines that are sometimes valued (as in the topological watershed [CB97b, NC03a] which conserves the saliency between minima) whereas ‘region-algorithms’ return labeled regions (the CBs) that form a partition of the image.

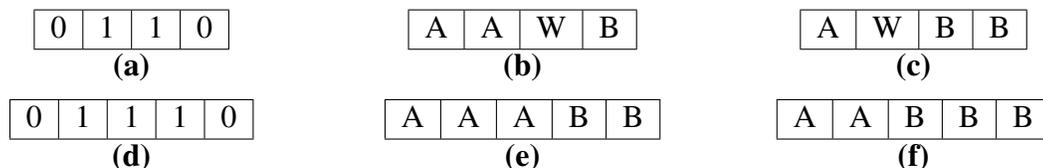


Fig. 2.1: Original images (a)(d) and two possible labeled WS outputs (raster or anti-raster scan) of a line-algorithm (b)(c) or a region-algorithm (e)(f). W represents the WS line.

In lots of (line- or region-) algorithms, the result varies with implementations (scanning order and other arbitrary processing order) or may be inconsistent with the WS definition as observed in [RM00]. This variation due to implementation can be insignificant in some cases (1 pixel bias for the line/region position, see Fig. 2.1) but in other cases, it becomes considerable: in some images an entire region is reached passing by a ‘bottleneck’ pixel and consequently included to the first (or last) CB that invades the bottleneck (like in Fig. 2.2(k)-(n)). Thus, the problem does not occur only on plateaux. Furthermore, it is of theoretical interest having a unique solution for the WS transform. These arguments encouraged the investigation of a WS definition that would result in a unique and consistent solution.

The Image Foresting Transform (IFT), introduced by Falcão, Lotufo and Stolfi [FSL04] and based on Dijkstra’s algorithm [Dij59], provides a sound framework for the efficient implementation of many image processing operators [FCL01]. For instance, the WS transform is computed as a problem of trees of minimal paths.

This paper is organized as follows. In section 2.2, an overview of the IFT framework is given to define in this context the Tie-Zone Watershed (TZWS) that results in a unique solution, regardless of implementation. An efficient algorithm is presented in section 2.3 and finally two applications are described in section 2.4: a bounding interval for each segmented object’s extension and a progressive segmentation procedure.

2.2 Definition

2.2.1 Overview of the Image Foresting Transform (IFT)

Under the IFT framework, an image is seen as a weighted graph $G = (V, A, I)$ where each pixel (or voxel in 3D) is represented by a node or vertex $v \in V$ with intensity $I(v)$ (I is a map from V to \mathbb{Z} for digital image). An arc $\langle u, v \rangle \in A$ exists between vertices u and v when the corresponding pixels are adjacent according to the defined adjacency (usually 4- or 8-adjacency in 2D and 6- or 26-adjacency in 3D). A path from a node u to a node v in a graph (V, A, I) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said simple if all its nodes are different from each other. Let $S \subseteq V$ be a set of particular nodes s_i called seeds or markers.

For a given weighted graph $G = (V, A, I)$ and a set $S = \{s_i\}$ of seeds, an IFT is a *directed forest* F of G , i.e. a directed acyclic subgraph¹ F of G , such that (i) there exists for each node $v \in V$ a *unique and directed simple path* $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path has a *minimum cost* for linking v to any seed of S , according a specified path cost function f_C .

Assume that the arcs $\langle u, v \rangle$ are weighted with the gray-level $I[v]$ of the pixel corresponding to v . Assume that the seed nodes correspond to the regional minima of the image (or to imposed minima, i.e. markers). If the path cost function is defined as the ‘maximal arc’ function f_{max} ,

$$f_{max}(\langle v_1, v_2, \dots, v_n \rangle) = \max \{h(v_1), I(v_2), \dots, I(v_n)\}$$

where h is a fixed but arbitrary *handicap cost* [LFZ02], the IFT computes a region-WS transform where each tree of the forest² corresponds to a CB. Note that all vertices (pixels) are covered by this forest. The IFT can result in many optimal forests because many paths of minimum cost are sometimes possible. The set of all optimal forests is denoted by Φ .

The optimality of the WS by IFT was proved in [LF00] where a two-component lexicographic cost function f_{LC} was proposed to mimic the flooding process and handle with plateaux too: $f_{LC} = (f_{max}, f_d)$. The first component, of highest priority, is the max-arc function and represents the flooding process. The second one makes different waters propagate on plateau at a same speed rate:

$$\begin{aligned} f_d(\langle v_1, v_2, \dots, v_n \rangle) &= \max_{k \in [0, n-1]} \{k, C[v_n] = C[v_{n-k}]\} \\ C[v_n] &= f_{max}(\langle v_1, v_2, \dots, v_n \rangle) \end{aligned}$$

¹The graph $G' = (V', A')$ is a subgraph of G if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$.

²A tree of the forest F is a connected component of F .

This lexicographic path cost, inspired from Meyer’s topographical distance strategy [Mey94b], is very simple to compute, avoids a prior lower completion on image with plateaux, and provides partitions that seem to be more equitable (on plateaux) than when only the maximum cost is used.

2.2.2 The Tie-Zone Watershed (TZWS) transform

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image-graph. We propose then a new definition of watershed transform in the IFT context which results in a unique partition.

A node is included in a specific catchment basin CB_i when it is linked by a path to a same seed s_i in all the optimal forests, otherwise it is included in the Tie-Zone T :

$$CB_i = \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\}$$

$$T = V \setminus \bigcup_i CB_i$$

If a node is in the tie-zone, it means that it could be included in different CBs without affecting the forest optimality. CBs are only the common part of all optimal solutions whereas differing parts are considered TZ. Therefore, the tie-zone existence prevents from making any arbitrary choice between optimal solutions. Consequently, the TZWS solution is defined without ambiguity.

Note that this definition does not produce watershed lines but only regions: catchment basins and tie zone. They form together a *unique* optimal partition of the image. If all pixels are assigned in a catchment basin, the tie zone will be empty. This situation can occur when the lexicographic path-cost function unties growing CBs on plateaux. So, the watershed transform possibly does not contain any tie zone.

Unlike in the WS by IFT, each CB corresponds to a tree *or* part of it, while the TZ is composed of many terminal parts of trees as in the example of Fig. 2.2(b).

2.3 Algorithm

In this section, we present an efficient algorithm that labels the image in order to obtain a TZWS. It is based on Dijkstra’s shortest path algorithm [Dij59] and utilizes an ordered queue Q where each bucket has a FIFO policy. Note that the second component C_2 of the lexicographic cost is *not* intrinsically computed by the FIFO policy and must be explicit in the TZWS by IFT in order to prevent 1-pixel bias.

The algorithm input is: the image (or scene) as a weighted graph $G = (V, A, I)$, the seed node

set S with associated labeling function λ and handicap function h . The priority queue Q is initially empty: `DequeueMin` removes from Q and returns the node of minimum cost; `Enqueue(p, c)` inserts node p in Q at priority (cost) c bucket. We denote the neighborhood of a node $p \in V$ by: $N_G(p) = \{q \in V, \langle p, q \rangle \in A\}$. Label map L corresponds to the TZWS result, map P gives each node's predecessor in the tree and maps C_1, C_2 give the lexicographic cost of an optimal path from a seed to each node.

The beginning of the algorithm (lines 1 to 11) is identical with the IFT algorithm in [FSL04]. Lines 12-16 are TZWS-specific. In line 12, the second component of lexicographic cost is incremented, as water propagates on plateau. Lines 13 to 16 detect the nodes where paths from (at least) two seeds ($L[p] \neq L[v]$) tie, i.e. have same costs (C_1 and C_2).

Algorithm 1: TZWS by IFT with lexicographic path cost.

1. $\forall p \in V, C_2[p] \leftarrow 0; \quad done(p) \leftarrow \text{FALSE};$
2. $\forall p \notin S, C_1[p] \leftarrow \infty; \quad L[p] \leftarrow \text{NIL}; \quad P[p] \leftarrow \text{NIL};$
3. $\forall p \in S, C_1[p] \leftarrow h(p); L[p] \leftarrow \lambda(p); \quad P[p] \leftarrow p; \quad \text{Enqueue}(p, h(p));$
4. **while** `QueueNotEmpty`,
5. $v \leftarrow \text{DequeueMin}; done(v) \leftarrow \text{TRUE};$
6. $\forall p \in N_G(v) \text{ and } done(p) = \text{FALSE},$
7. $c \leftarrow \max\{C_1[v], I[p]\};$
8. **if** $c < C_1[p],$
9. **if** p in $Q, \text{Dequeue}(p);$
10. $C_1[p] \leftarrow c; L[p] \leftarrow L[v]; P[p] \leftarrow v;$
11. `Enqueue($p, C_1[p]$)`;
12. **if** $c = C_1[v], C_2[p] \leftarrow C_2[v] + 1;$
13. **else, if** $c = C_1[p] \text{ and } L[p] \neq L[v],$
14. **if** $c = C_1[v],$
15. **if** $C_2[p] = C_2[v] + 1, L[p] \leftarrow \text{TZ};$
16. **else** $L[p] \leftarrow \text{TZ};$

This algorithm is fast and has the same speed performance as the IFT-WS [FSL04]. The solution of TZWS is optimal because it is based on IFT, it keeps therefore the optimality of the shortest-path forest solution as demonstrated in [LF00, FSL04]. Besides, the other algorithms generally depend on arbitrary decisions in processing order (which pixel is removed first from a priority bucket of the queue?) that are not in the strict definition of WS and that introduce bias (see the different solutions of Fig. 2.2(i)-(n)). Note that the case of buttonhole shown in Fig. 2.2 is similar to configurations found in real images, as referred by [NC03a]. Observe that the bias problem does not occur only on plateaux and may be unacceptable for some applications (e.g. precise measures on segmented objects).

2.4 Applications

The Label Merging Algorithm (LMA) is a useful variation of the previous algorithm. When waters from different minima are merging, it assigns a new blended label to the region invaded by these waters. Substitute in algorithm 1 TZ label (lines 15-16) by `MergeLabels(L[p], L[v])`. So, the final labeled image allows a traceability on tie zones: it informs exactly which and how many trees of different labels tied together at each node. There is no more one TZ label but so many as the distinct label mergings (see Fig. 2.2(c)). This merging information can be used in several applications.

2.4.1 Maximal extension of segmented objects

As we saw, the proposed TZWS gives sets of nodes that are unequivocally associated with specific seeds, the CBs: they represent the minimum extensions E_{min}^i of segmented objects O^i and form, with the tie zone T , a partition of the image (V) (see Fig. 2.2(b)): $\forall i \neq j, E_{min}^i \cap E_{min}^j = \emptyset, E_{min}^i \cap T = \emptyset$ and $V = \bigcup_i E_{min}^i \cup T$.

The LMA allows to obtain the maximal extension E_{max}^i of each segmented object, composed by the corresponding E_{min}^i and all nodes assigned with merged labels where label i tied. Therefore, one can be sure that with a defined set of markers/minima, all WS algorithms (with same adjacency and cost definitions) will result in segmented objects O^i in these limits: $E_{min}^i \subseteq O^i \subseteq E_{max}^i$ (see Fig. 2.2(d)-(f)). Consequently, measures on segmented objects can be bounded. Moreover, in interactive segmentation procedures, these boundaries can help the user to correct or determine new markers for further resegmentations.

2.4.2 Progressive segmentation

As the final goal of segmentation is to partition images into objects, it is desirable not to have large undefined regions. A thinning procedure can therefore be used. We propose now a progressive segmentation in order to reduce the thick tie-zones without making implementation-dependent choices.

The number of labels that tied together at each node can be seen as the degree of liability of a single-label assignment. More labels tied at a node, more unliable would become a single-label assignment. The number of tied labels NTL is computed during the LMA at each call of the `MergeLabels` function. The map of NTL can be used to resegment the result of a first TZWS. This step is repeated until stability of the segmentation result. In most of cases, the thick tie zones will be progressively thinned or disappear. It is another possible option for the segmentation task.

In this new procedure, all regions segmented with certitude (minimum extensions) are definitively

labeled whereas the tie zones have to be conquered in the next step by these labeled regions. The tie zones' original topography is then modified to allow a new segmentation by TZWS: it is modified into plateaux whose level is proportional to the number of candidate labels (Fig. 2.2(g)). Consequently, the proximity criterion progressively unties the tie-zones (Fig. 2.2(h)).

2.5 Conclusions and Future Work

In this work, we introduced a new definition of watershed transform under the IFT framework: the Tie-Zone Watershed. It provides an optimal, unique and, thereby, unbiased solution: it is not implementation-dependent. We presented an efficient algorithm based on Dijkstra's one and a simple lexicographic cost, and proposed finally two possible applications of the TZWS. Firstly, it provides limits for the segmented objects' extensions which could guide the user in interactive segmentation process or bound measures on objects. Secondly, a procedure of progressive segmentation based on TZWS is described: the number of merging CBs is the key to this tie-zone thinning proposal.

As the TZWS is the consensus of all IFT-based solutions, the resulting partition is unique and unbiased. In future work, we plan to characterize the robustness of segmentation with tie-zone features. For example, the tie-zone size can be a warning of potential bias or unreliability in segmentations. We also intend to compare the TZWS with hierarchical WS approach and to investigate the relation of the bottleneck phenomenon (in bold in the Fig. 2.2(a)) with pass-value and saliency concepts, as well as similarities with the topological watershed.

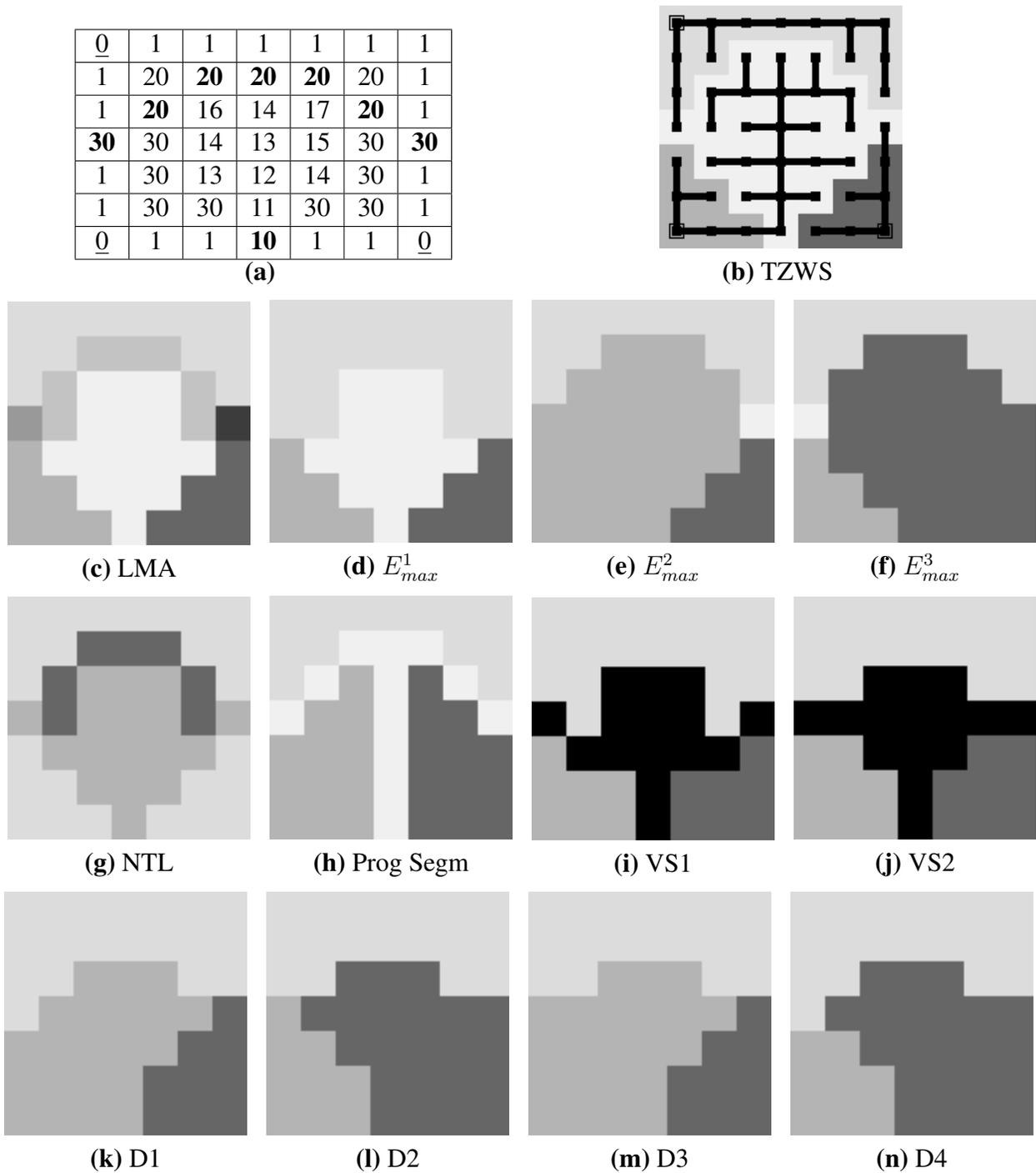


Fig. 2.2: (a): Original grayscale image (3 minima, 8 bottlenecks). (b): TZWS using 4-adjacency: 3 CBs (gray), TZ (white), forest (black). (c): Result of the Label Merging Algorithm. (d)-(f): Each image extends a specific CB_i to its E_{max}^i while remaining CB_j have only minimum extensions. Remaining TZ in white. (g): Map of Number of Tied Labels. (h): Result of the progressive segmentation: TZ (in white) is thinned. (i)-(j): Watersheds (black) by Vincent and Soille's algorithm (raster or anti-raster scan). (k)-(n): Watersheds by Dijkstra-IFT varying scanning-order.

Capítulo 3

Afinamento da Zona de Empate com Solução Única

Este capítulo contém o artigo intitulado *Uniquely-determined thinning of the tie-zone watershed based on label frequency* [AL07b]. Primeiro, ele ressalta que a transformada em zona de empate do *watershed* por IFT tem solução única independente de implementação e possui um algoritmo consistente com sua definição ao contrário dos outros *watersheds* encontrados na literatura. Com efeito, os algoritmos de *watershed* presentes na literatura se dividem em algoritmos “de linha” ou “de região”, conforme existem ou não linhas divisoras explícitas, isto é, materializadas por pixels. Esta dicotomia é incompatível com a maioria das definições de *watershed* no espaço discreto [BM93, Mey94b, VS91]. E quando é compatível, a transformada de *watershed* tem várias soluções [BBM⁺97, FSL04, CNB05a]. A transformada em zona de empate do *watershed* por IFT possui unicidade e consistência. Não tem linhas divisoras, mas a zona de empate, quando não for vazia, fica entre bacias de retenção, sem necessariamente separá-las totalmente: Bacias podem ser adjacentes em certos pontos. Isto acontece, por exemplo, nos pontos em que o custo lexicográfico desempata as bacias na atribuição dos platôs da imagem a diferentes segmentos. Também é frisado que o custo lexicográfico tem o mesmo papel que a operação de declivamento (*lower completion*) geralmente aplicada à imagem em pré-processamento quando outros tipos de *watershed* são utilizados.

Apesar de única e consistente, a transformada em zona de empate ainda pode gerar uma zona de empate espessa, embora menos espessa que se o custo lexicográfico não fosse utilizado. Por isso, propõe-se um afinamento da zona de empate baseado na frequência dos rótulos atribuídos, e com solução única. Considerando-se todas as rotulações decorrentes das soluções da IFT-WT-lex, calcula-se, para cada pixel, a frequência em que aparece cada rótulo. No final, o rótulo mais freqüente é escolhido.

Demonstra-se que a determinação das frequências de rótulo pode ser feita pelo paradigma da

fragmentação da gota d'água. Quando uma gota d'água cai em cima de um pixel e pode seguir vários caminhos descendentes, ela fragmenta-se em partes iguais e cada um dos fragmentos segue um dos caminhos até um mínimo regional. Em cada pixel encontrado, essa fragmentação é feita, mas fragmentos da gota também podem se juntar quando os caminhos se cruzam. A quantidade de água recolhida por cada mínimo regional corresponde à frequência com que seu rótulo respectivo é atribuído ao pixel. O grafo ótimo dos predecessores múltiplos (MOG) descreve todos os caminhos descendentes utilizáveis por uma gota d'água.

As quantidades de água recolhidas também podem ser calculadas recursivamente pelo processo inverso de inundação, onde os fluxos ascendentes inundando cada pixel são normalizados antes de se fundir. Isso permite calcular simultaneamente a zona de empate, a frequência dos rótulos e o afinamento, sem explicitar todas as soluções possíveis da IFT-WT. Por consequência, o algoritmo proposto, derivado da estrutura geral do algoritmo de Dijkstra, utiliza uma fila ordenada que processa cada pixel apenas uma vez. Observa-se que a atribuição do rótulo mais frequente é condicionada a regras que mantêm conexo cada segmento gerado. Portanto, existem dois fatores que podem limitar o afinamento total: a criação de regiões desconexas para um mesmo segmento e/ou a existência de pelo menos dois rótulos mais frequentes.

Abstract

There are many watershed transform algorithms in literature but most of them do not exactly correspond to their respective definition. The solution given by such algorithms depends on their implementation. Others fit with their definition which allows multiple solutions. The solution chosen by such algorithms depends on their implementation too. It is the case of the watershed by image foresting transform that consists in building a forest of trees with minimum path-costs. The recently introduced tie-zone watershed (TZW) has the advantage of not depending on arbitrary implementation choices: it provides a unique and, thereby, unbiased solution. Indeed, the TZW considers all possible solutions of the watershed transform and keeps only the common parts of them as catchment basins whereas parts that differ form a tie zone disputed by many solutions. Although the TZW insures the uniqueness of the solution, it does not prevent from possible large tie zones, sometimes unwanted in segmentation applications. This paper presents a special thinning of the tie zone that leads to a unique solution. Observing all the possible solutions of the watershed by image foresting transform, one can deduce the frequency of the labels associated with each pixel. The thinning consists in assigning the most frequent label while preserving the segmented region connectivity. We demonstrate that the label frequency can be computed both from an immersion-like recursive formula and the proposed fragmented drop paradigm. Finally, we propose an algorithm under the IFT framework that computes

the TZW, the label frequency and the thinning simultaneously and without explicit calculation of all the watershed solutions.

CB	catchment basin
EF	“equal frequency” label
FIFO	first-in-first-out
IFT	image foresting transform
ISO	“isolated” label
MOG	multipredecessor optimal graph
SKIZ	skeleton by influence zones
TZ	tie-zone
TZW	tie-zone watershed
WT	watershed transform

Tab. 3.1: Table of abbreviations

3.1 Introduction

The watershed transform (WT)¹ is a famous and powerful segmentation tool in morphological image processing. First introduced by Beucher and Lantuéjoul [BL79] for contour detection and applied in digital image segmentation by Beucher and Meyer [MB90], it is inspired from a physical principle well-known in geography: if a drop of water falls on a topographic surface, it will follow the greatest slope until reaching a valley. The set of points which lead to the same valley is called a catchment basin (CB). The watershed lines separate different catchment basins. In the WT, an image is seen as a topographic surface where grey-level corresponds to altitude. Generally, to segment an image by WT, a gradient of the image is used as topographic surface. In this case, it is expected that a region with low gradient, a valley, corresponds to a rather homogeneous region and possibly to the same object. Ideally, catchment basins correspond to segmented objects separated by watershed lines.

Many definitions for WT exist in literature. Definitions in the continuous space have been proposed [BL79, Mey94b, NS94, Prê93] and consider the watershed as a skeleton by influence zones (SKIZ) generalised to greyscale images. Each proposal gives a unique solution for the watershed. In the digital space (of interest in this paper), there are many definitions using different paradigms. We recall some of them.

The recursive definitions of digital WT by “immersion” [VS91] or “flooding” [BM93] and the definition of digital WT by “topographic distance” [Mey94b] can be seen as digital versions of the generalised SKIZ paradigm. Each definition, if strictly applied, gives a unique solution. But the

¹Table 3.1 contains all the abbreviations used in the text.

watershed is not necessarily thin [VS91]. Moreover, not all CBs are separated by watershed lines. Some can have direct contacts (e.g., see the case of an even plateau in Figure 3.1(a)-(b)). This is due to the discretization of a paradigm designed in the continuous space.

The topological watershed proposed by Couprie and Bertrand [CB97a] cannot be viewed as a generalised SKIZ but as the ultimate topological thinning that transforms the image while preserving some topological properties [Ber05, NCB05]. Furthermore, the separating watershed lines are valued such as the saliency between CBs is preserved [NC03b]. Multiple solutions are allowed by the topological watershed definition and each CB is necessarily separated from the others by a watershed line (cf. Figures 3.2(a),(d)-(f)).

The image foresting transform (IFT), a graph-based framework introduced by Falcão, Lotufo and Stolfi [FSL04], also defines a digital WT. Yet, this WT is made of CBs only (no separating watershed line): it is a “region WT”. In this paradigm, the image is seen as a graph where pixels are nodes, and the WT seen as a problem of trees of minimal paths [LF00]. Indeed, the WT by IFT consists in creating an optimal forest from the image-graph, i.e. a set of trees that have minimum path-costs. These trees correspond to the CBs. Multiple solutions (optimal forests) are allowed by the definition (cf. Figures 3.2(a)-(c)).

Next to the many definitions, there are lots of algorithms that compute a digital WT. The WT algorithms can be classified in two types: (i) *line-algorithms* that return watershed lines separating each CB from the others [CNB05a, CNB05b, VS91]; (ii) *region-algorithms* that return labelled regions (the CBs) which constitute a partition of the image without any watershed pixel [BM93, BBM⁺97, BM98, BM00, FSL04, MR98, Mey94b, RM00].

Yet, many algorithms do not correspond to their respective digital definition. For example, the Vincent-Soille’s algorithm [VS91] imposes lines between CBs so that it does not fit with any definition and many solutions are possible depending on the implementation (cf. Figures 3.1(c)-(d) and 3.3). On the contrary, the algorithms presented by Meyer in ref. [Mey94b] “do not produce watershed labels and, therefore, are not exact implementations of the definition. All pixels are merged with some basin, so that, dependent on the order in which pixels are treated, different results may be produced” [RM00]. It is clear (see Figure 3.1) that, by adapting the definitions to adequate with the output type constraint (lines or regions), these algorithms introduce a bias in the output according to the processing order (implementation choice). There is a lack of robustness of the result due to the implementation.

Other algorithms, like the topological watershed (line) and the watershed by IFT (region), fit with their respective definitions but give only one solution among the multiple solutions allowed, which depends on the (arbitrary) implementation. Therefore, they can give different solutions depending on the processing order (cf. Figure 3.2). Again, there is a lack of segmentation robustness due to the

implementation but it does not conflict with the definitions.

In summary, the dichotomy between line and region algorithms is incompatible with most of the proposed digital definitions. And when it is compatible, the WT is not uniquely defined.

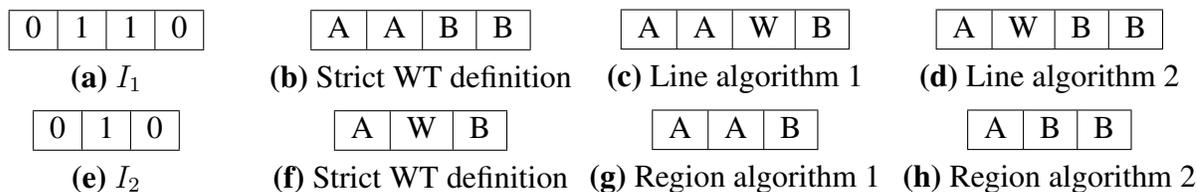


Fig. 3.1: Data processing order introduces a 1-pixel bias. **(a)-(d)**: Input image, WT according to the definition, and two possible labelled WT outputs (raster or anti-raster scan) of a line-algorithm. W represents the watershed line. **(e)-(h)**: Input image, WT according to the definition, and two possible labelled WT outputs (raster or anti-raster scan) of a region-algorithm.

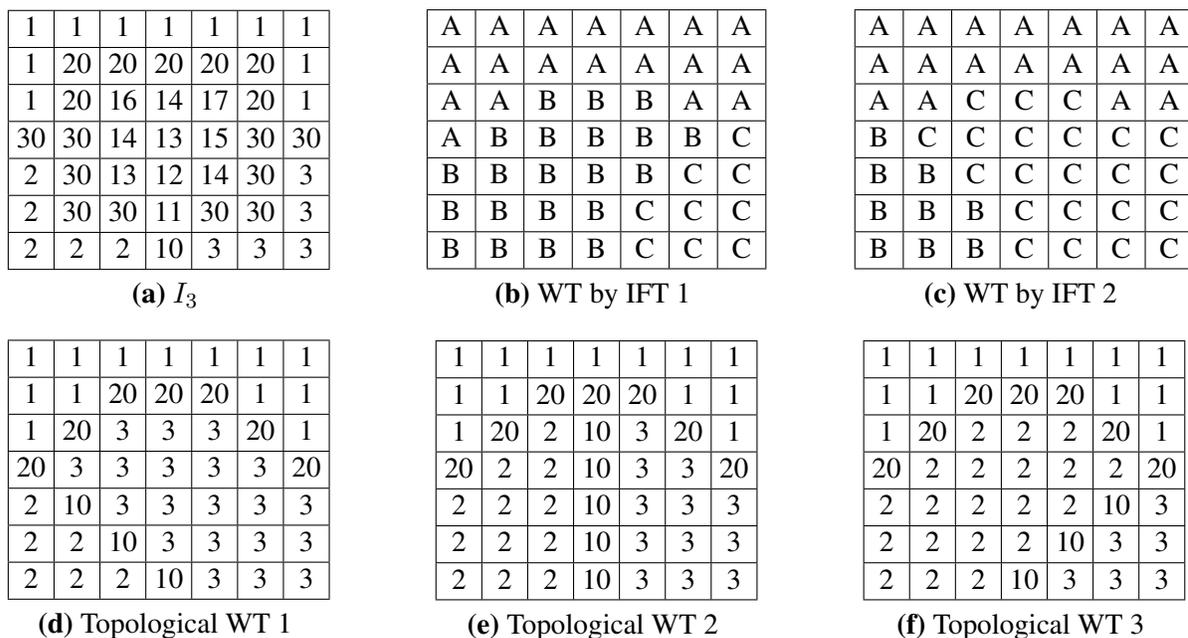


Fig. 3.2: Multiple solutions do not contradict the definition but are consequences of specific implementations. **(a)**: Input greyscale image with 3 minima. **(b)-(c)**: Possible label maps of the WT by IFT using 4-adjacency (raster or anti-raster scan respectively). **(d)-(f)**: Possible topological watersheds for different data processing orders. Minima's grey-levels are extended in CBs and watershed lines are valued.

In this paper, we focus on the WT by image foresting transform (IFT). By returning only one of the multiple solutions, the algorithm makes necessarily an arbitrary choice. The returned solution is, therefore, biased by the algorithm implementation. This variation due to implementation can be insignificant in some cases (1-pixel bias for the line or region position, see Figure 3.1(g)-(h)) but

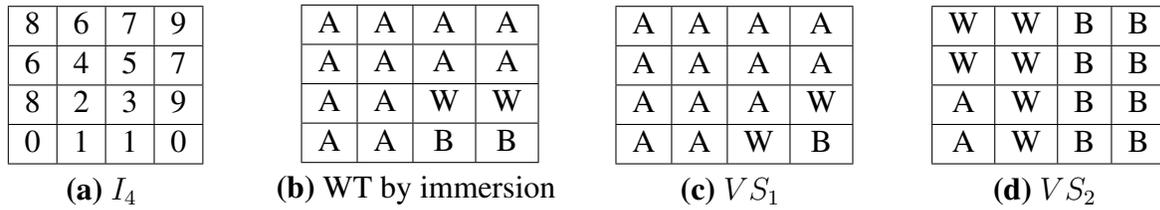


Fig. 3.3: **(a)**: Input greyscale image with 2 minima. **(b)**: Solution given by the strict definition of WT by immersion: watershed line is not separating. **(c)**-**(d)**: Possible watershed lines (W) by Vincent and Soille's algorithm (raster or anti-raster scan respectively) using 4-adjacency. Multiple solutions are not predicted by the definition but are consequences of the bias introduced by implementation.

in other cases, it becomes considerable and even may be unacceptable for some applications (e.g., reliable measures on segmented objects). In some images an entire region is reached passing by a bottleneck pixel and consequently included to the first (or other arbitrary) CB that invades the bottleneck (like in Figures 3.2(b)-(c)). Notice that the problem does not occur only on plateaux but also in the buttonhole configurations: regions invaded by passing through a bottleneck whose merging with a CB or a watershed line is affected by the 1-pixel bias depicted in Figure 3.1. It is important to remark that buttonhole cases “correspond to special pixel configurations which are not so rare in practice” as reported by refs. [NC03b, VS91].

How to provide a unique solution for the WT by IFT independent of the implementation? This question motivated the recent introduction of the tie-zone watershed (TZW). Roughly speaking, the definition of TZW [ALC05] takes into account all the possible solutions (optimal forests) derived from the strict WT definition to generate a unique solution: when the multiple solutions disagree with each other on the segmentation result of a region (i.e., the label to be assigned), the region is included to the tie-zone (TZ) and a specific tie label is assigned. The TZ is therefore a litigious zone. The TZW avoids to make arbitrary decisions.

The size and distribution of the TZ can be used as a tool that gives a measure of the non-robustness or unreliability of a segmentation in relation to the results given by other implementations of IFT-WS. In other words, the size of the TZ indicates how different could be the segmentation if another implementation were used [AL05].

For a segmentation purpose, the possibly large TZ may be unwanted. As a large TZ indicates a poor reliability of the segmentation, the user may add some extra markers (see refs. [BM93, MB90] for the concept of marker) to modify the segmentation and try to untie the TZ, and then increase the reliability of the segmentation. Observe that a reliable or robust solution is not necessarily a semantically correct solution. It is reliable in relation to the segmentation paradigm that was defined: in this case, the optimality of the forest.

An alternative to extra markers is to apply an automatic thinning on the TZ that assigns the TZ

to already segmented regions (CBs). The TZ thinning must be unique and independent of the implementation, otherwise it is not worth having firstly applied the TZW. However, there is still a trade-off between uniqueness of the solution and thinness of the TZs, so that perhaps only part of the TZ can be thinned to preserve the uniqueness of the solution.

A first proposal of TZ thinning was introduced in ref. [ALC05]. It consists in applying iteratively the TZW on a topography whose altitude is the number of different labels that tied together. Consequently, a distance criterion indirectly unties the TZ. This thinning drastically simplifies the topography of the original TZ.

In addition, an accidental leak of water in watershed can create a non-representative TZ. For example, if almost all the WT solutions agree in assigning a region to a same CB but only one WT solution assigns it to another one, the region becomes a TZ, equally disputed by both CBs. Thus, one solution has as much weight as all the others. On the contrary, it could be desirable that all the solutions have the same importance.

We propose in this paper a thinning of the TZ keeping the property of uniqueness and based on the label frequency: observing all the possible watershed solutions by IFT, the TZ can be untied when a region is most frequently assigned to a specific CB. Considering the drop of water paradigm, the TZ corresponds to regions where a drop of water could fall and follow several ways and slide down to different valleys. Imagine now that every time the drop of water can follow several ways, it is split in smaller equal amounts of water which follow the possible ways. The fractions of water the valleys receive are evaluated. At the end, the greatest fraction determines which CB the drop belongs to, as long as the CB connectivity is preserved, i.e., if there exists a path linking the locus the drop fell to the valley, entirely included in the CB. We demonstrate that the label frequency used by the thinning of TZW can be computed both from the fragmented drop paradigm and an immersion-based formula.

This paper is organised as follows. The notation and definitions on graphs necessary to understand the IFT framework are introduced in Section 3.2. Then, Section 3.3 gives an overview of TZW and recalls the watermerging paradigm that sustains the idea of tie-zones. The multipredecessor optimal graph, a special graph describing the water flows and mergings, is defined too. Finally, Section 3.4 deals with the several ways of determining the label frequency, defines the thinning of the TZ based on label frequency, and presents an algorithm to compute at the same time the TZW and its frequency-based thinning.

3.2 Watershed Transform by Image Foresting Transform

In this section, the notation and definitions for the watershed by image foresting transform (IFT) are recalled. The IFT is a general framework based on graph theory in which an image is interpreted

as a graph and pixels as its nodes. The key idea of this transform is to obtain, according to a path-cost function, a *shortest path forest* from an input image-graph. Depending on the path-cost function and some other input parameters (adjacency, arc weights), the IFT can compute different image processing operations [FCL01, FSL04]: distance transforms, connected filters, interactive object delineation (“live-wire”), segmentation by fuzzy connectedness and segmentation by watershed.

3.2.1 Notation and Definitions

Under the IFT framework, an image is seen as a weighted graph $G = (V, A, I)$ where each pixel (or voxel in 3D) is represented by a node or vertex $v \in V$ with intensity $I(v)$. For digital image, I is a map from V to \mathbb{Z} . An arc $\langle u, v \rangle \in A$ exists between vertices u and v when the corresponding pixels are adjacent according to the defined adjacency (usually 4- or 8-adjacency in 2D and 6- or 26-adjacency in 3D). A path $\pi(u, v)$ from a node u to a node v in a graph (V, A, I) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said simple if all its nodes are different from each other. Let $S \subseteq V$ be a set of particular nodes s_i called seeds. The graph $G' = (V', A')$ is a subgraph of G if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$. A directed forest F of G is a directed acyclic subgraph F of G . A tree of the forest F is a connected component of F .

For a given weighted graph $G = (V, A, I)$ and a set $S = \{s_i\}$ of seeds, the *image foresting transform* (IFT) returns an *optimal forest*, i.e. a directed forest F of G such that (i) there exists for each node $v \in V$ a unique and directed simple path $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path has a minimum (or “optimum”) cost for linking v to any seed of S , according to a specified path-cost function f_C .

Assume that the arcs $\langle u, v \rangle$ are weighted with the grey-level $I[v]$ of the pixel corresponding to v . Assume that the seed nodes correspond to the regional minima of the image (or to imposed minima, i.e. markers [BM93]). If the path-cost function f_C is defined as the ‘maximum arc’ function f_{\max} ,

$$f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) = \max \{h(v_1), I(v_2), \dots, I(v_n)\}$$

where h is a fixed but arbitrary handicap cost [LFZ02], the IFT returns a region-WT where the trees of the forest correspond to the CBs. Note that all vertices (pixels) are covered by this forest. The handicap cost is generally set to $I(v_1)$ or 0 in case of minima imposition [BM93].

The IFT can result in many optimal forests because many paths of (same) minimum cost are sometimes possible for some nodes. The set of all the optimal forests F is denoted by Φ . Observe that a forest can be simply represented by a predecessor map P where $P[v]$ denotes the predecessor of node v in the minimum path. In addition, a cost map C can indicate for each node v the cost $C[v]$ of the

minimum path from the tree root to v . When $f_C = f_{\max}$, it corresponds exactly to the morphological superior-reconstruction [FSL04] of the input image I from the specified seeds (natural or imposed minima). For segmentation purpose (WT by IFT), a label map L is generally associated with the optimal forest, so that, for each node v , $L[v]$ represents the label of the corresponding minimum-path root. Notice that the final cost map C is unique (values of the minimum paths are unique) while the predecessor map P and then the labelling L may be multiple.

In ref. [LF00], a two-component lexicographic cost function f_{LC} was proposed to mimic the flooding process and handle with plateaux too: $f_{LC} = (f_{\max}, f_d)$. The first component, of highest priority, is the max-arc function and represents the flooding process. The second one corresponds to the geodesic distance to the lower boundary of the plateau and makes different waters propagate on plateau at a same speed rate:

$$\begin{aligned} f_d(\langle v_1, v_2, \dots, v_n \rangle) &= \max_{k \in [0, n-1]} \{k, C[v_n] = C[v_{n-k}]\} \\ C[v_n] &= f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) \end{aligned}$$

The use of the lexicographic cost provides partitions that seem to be more equitable (on plateaux) than those obtained with the maximum cost component only. In fact, the lexicographic cost avoids a prior lower completion on image with plateaux but has strictly the same role.

In Figure 3.4(a), a greyscale image with two minima and many plateaux is showed. Figures 3.4(a) and (b) show the two cost maps corresponding to the two lexicographic cost components returned by the IFT. The first component (f_{\max}) always corresponds to the reconstruction of the input image from the specified seeds (natural or imposed minima). In this case, all the regional minima are considered seeds, so the reconstruction is the input image itself. The second component (f_d) corresponds exactly to the geodesic distance (minus one) to the lower boundary of the plateau. With these geodesic distances on plateaux, it is possible to transform (see definition 3.4 of ref. [RM00]) the original input image to a lower complete image (see Figure 3.4(c)), as required in many algorithms (e.g., hill climbing, topographic distance).

3.2.2 Algorithms for IFT

The algorithm for IFT computes three attributes for each vertex $v \in V$: its predecessor $P[v]$ in the minimum path, the cost $C[v]$ ($C[v] = (C_1[v], C_2[v])$ when the lexicographic cost is used) of that path, and the corresponding root label $L[v]$.

The efficient ordered queue-based algorithm for IFT proposed in refs. [FSL04, LF00] is essentially Dijkstra's algorithm [Dij59], extended for multiple sources and a more general path-cost

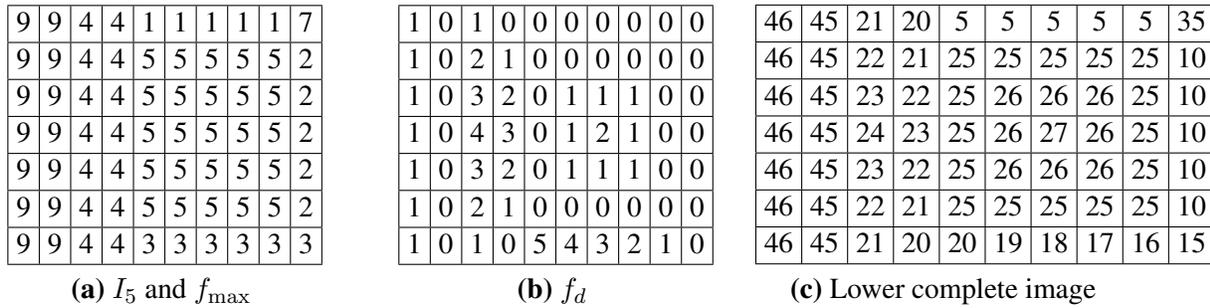


Fig. 3.4: **(a)**: Input greyscale image I_5 with 2 minima is equal to the first component (f_{\max}) of the lexicographic cost. **(b)**: The second component (f_d) of the lexicographic cost corresponds to the geodesic distance to the lower border. **(c)**: The lower complete image. Lower completion increases the image grey-level range.

function. It is denoted Dijkstra-IFT in this paper. Note that, for the 4- or 8-adjacency in 2D or 6- or 26-adjacency in 3D, the ordered queue can be implemented such that the IFT algorithm will run in time proportional to the number of vertices [FSL04].

The lexicographic path-cost is very simple to compute: only the first component (maximum arc) is explicitly computed. The second component is implicitly computed by using a priority first-in-first-out (FIFO) queue [LF00]. As we said, the use of this lexicographic path-cost substitutes the lower completion step. Observe that the explicit lower completion as presented in Figure 3.4(c) increases the range of the image grey-levels. So, in addition to the computation cost of the lower completion step, there may be, in some images, an extra storage cost.

Note also that other algorithms are able to compute the IFT. For example, the ordered queue is not necessary. One can process the image data in raster-scan and anti raster-scan order alternatively until stability of the result (algorithm not presented here and based on Berge's one [Ber58], cited in Section 4.3 of ref. [Mey94b]).

3.3 Optimal forest paradigm and tie-zone watershed

We recall in this section the main definitions and concepts on tie-zone watershed introduced in ref. [ALC05].

3.3.1 The tie-zone watershed (TZW) transform

Definition

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image. We propose then a new definition of WT in the IFT context which results in a unique partition, i.e. a unique label map.

A node is included in a specific catchment basin CB_i when it is linked by a path to a same seed s_i in all the optimal forests, otherwise it is included in the Tie-Zone T :

$$CB_i = \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\}$$

$$T = V \setminus \bigcup_i CB_i$$

If a node is in the tie-zone, it means that it could be included in different CBs without affecting the forest optimality. CBs are only the common part of all optimal solutions whereas differing parts are considered TZ. Therefore, the tie-zone existence prevents from making any arbitrary choice between optimal solutions. Consequently, the TZW solution is defined without ambiguity.

Note that this definition does not produce watershed lines but only regions: catchment basins and tie-zone. They form together a unique optimal partition of the image. If all pixels are assigned to catchment basins, the tie-zone will be empty. This situation can occur when the lexicographic path-cost function unties growing CBs on plateaux (e.g., Figure 3.1(b) also corresponds to the TZW of image I_1 but has no tie-zone). So, this watershed transform possibly does not contain any tie-zone. In some cases, the TZ can be quite large (see Figure 3.5).

Unlike in the WT by IFT where each CB corresponds to entire trees with same label, in the TZW by IFT, CBs may correspond to rooted parts of trees while the TZ is composed of many terminal parts of trees as in the example of Figure 3.5(a).



Fig. 3.5: **(a)**: TZW applied on image I_3 of Figure 3.2 using 4-adjacency: 3 CBs (grey), TZ (white), a possible forest (black). **(b)**: Result of the Label Merging algorithm.

Dijkstra-based Algorithm

Now, we transcribe an efficient algorithm that labels the image in order to obtain a TZW [ALC05]. It is based on Dijkstra's shortest path algorithm [Dij59] and utilises an ordered queue Q where each bucket has a FIFO policy.

The algorithm input is: the image as a weighted graph $G = (V, A, I)$, the seed set S with associated labelling function λ and handicap function h . We denote the neighbourhood of a node $p \in V$ by: $N_G(p) = \{q \in V, \langle p, q \rangle \in A\}$.

In the output, we have the label map L corresponding to the TZW result, map P giving each node's predecessor in the tree and maps (C_1, C_2) giving the lexicographic cost of an optimal path from a seed to each node. Note that, unlike the algorithm for IFT [FSL04], the second component C_2 of the lexicographic cost is not intrinsically computed by the FIFO policy and must be explicit in the TZW by IFT in order to prevent 1-pixel bias.

The priority queue Q is initially empty. `DequeueMin` removes from Q the node of minimum cost and returns it; `Enqueue(p, c)` inserts node p in Q at priority (cost) c bucket. `QueueNotEmpty` indicates that the queue is not empty. The state flag $done(p)$ is TRUE when the node has already been processed, i.e. it has its definitive attributes.

Algorithm 1: Dijkstra-based TZW with lexicographic path-cost.

Inputs: image (V, A, I) , neighbourhood N_G (derived from A), seeds S , handicap h and labelling λ functions.

Outputs: label L , predecessor P and cost (C_1, C_2) maps.

Auxiliary Data: empty ordered queue Q , state flag $done$, cost variable c .

1. $\forall p \in V, C_2[p] \leftarrow 0; \quad done(p) \leftarrow FALSE;$
2. $\forall p \notin S, C_1[p] \leftarrow \infty; \quad L[p] \leftarrow NIL; \quad P[p] \leftarrow NIL;$
3. $\forall p \in S, C_1[p] \leftarrow h(p); \quad L[p] \leftarrow \lambda(p); \quad P[p] \leftarrow p; \quad Enqueue(p, h(p));$
4. **while** `QueueNotEmpty`,
5. $v \leftarrow DequeueMin; \quad done(v) \leftarrow TRUE;$
6. $\forall p \in N_G(v)$ **and** $done(p) = FALSE,$
7. $c \leftarrow \max\{C_1[v], I[p]\};$
8. **if** $c < C_1[p],$
9. **if** p in $Q, Dequeue(p);$
10. $C_1[p] \leftarrow c; \quad L[p] \leftarrow L[v]; \quad P[p] \leftarrow v;$
11. $Enqueue(p, C_1[p]);$
12. **if** $c = C_1[v], C_2[p] \leftarrow C_2[v] + 1;$
13. **else, if** $c = C_1[p]$ **and** $L[p] \neq L[v],$

- 14.** **if** $c = C_1[v]$,
15. **if** $C_2[p] = C_2[v] + 1$, $L[p] \leftarrow \text{TZ}$;
16. **else** $L[p] \leftarrow \text{TZ}$;

The beginning of the algorithm (lines 1–11) is identical to the IFT algorithm in ref. [FSL04] based on Dijkstra’s. Lines 12–16 are TZW-specific (line numbers are bold-faced). After cost, label and predecessor initialisations (*l.* 1–3), a loop for emptying the priority queue Q starts (*l.* 4–16). This ordered queue has firstly been filled with seed nodes, properly labelled and with their respective handicap cost (*l.* 3). The node v of highest priority, i.e. lowest cost, is removed from Q (*l.* 5) with its definitive attributes (cost $C[v]$, label $L[v]$, and predecessor $P[v]$). This indicates that the minimum path $\pi(s_i, v)$ from some seed $s_i \in S$ to the node v has already be found. For each node p neighbour of v , such that p has not been definitively processed, the cost c of a candidate path with terminus p passing by v is evaluated (*l.* 6–7). If c is lower than the already assigned cost $C_1[p]$ (*l.* 8), then the path to p passing by v is considered better (cheaper) than the current path that reaches p and the three attributes of p are updated (*l.* 10). If p has never been visited, i.e. p is not in Q , it is inserted in Q with cost c (*l.* 11). Otherwise, only its position in Q is updated (*l.* 9,11).

In line 12, the second component of lexicographic cost is incremented, as water propagates on plateau. Lines 13–16 detect the nodes p where paths from (at least) two seeds with different labels ($L[p] \neq L[v]$) tie together, i.e. have same costs (C_1, C_2). The special label TZ is assigned to such nodes.

Note that the algorithm is efficient because it has the same complexity as the algorithm of ref. [FSL04] that computes a simple WT by IFT, and it is not necessary to compute explicitly all the WT solutions to obtain the TZW. The solution of TZW, based on IFT, is optimal because it keeps therefore the optimality of the shortest-path forest solution as demonstrated in refs. [FSL04, LF00]. In addition, label map L cannot be biased by arbitrary processing order in queue remotion nor neighbour selection.

The TZW can also be obtained without using an ordered queue by processing the image data in an unordered way, for example by sequential forward and backward scannings alternatively, until stability of the result (algorithm not presented here and based on Berge’s one [Ber58, Mey94b]).

The area of the TZs, their distribution and number and distribution of their sources, the so-called bottlenecks, can be correlated with the robustness of a segmentation, i.e. with the degree of confidence a particular segmentation by WT has [AL05].

The definition of TZW can be extended to other WT definitions: similarly, all solutions have to be taken into account and the regions where labelling differences occur constitute the tie-zone.

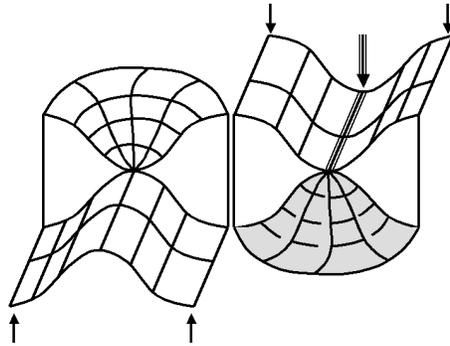


Fig. 3.6: Watermerging paradigm. On the left (right), the arrows point the 2 minima (maxima) of the topography. The triple line and the grey zone constitute the tie-zone where waters merged.

3.3.2 Watermerging paradigm

The WT is frequently compared to the flooding of a topography where dams are built to prevent distinct coloured waters from merging, supposing a colour is assigned to each minimum or marker [BM93, VS91]. The watermerging paradigm does not build such dams. For an intuitive comprehension of this paradigm, flip up-to-down the topography representing the image like in Figure 3.6. Imagine that each marker (former minimum that is now regional maximum) is a source of coloured water. When coloured waters meet together, no dam is built but the coloured waters naturally merge into a water of blended colour. Holes are punched in the regional minima (former maxima) for draining the water. Supposing waters propagate along all negative slopes (not only the steepest as occurs in reality), we get coloured hills and possibly regions with blended colours. The blended regions correspond to the tie-zone.

This intuitive up-to-down transformation is actually not used in implementations. When label a and label b tie together in a region, a merged-label $\{a, b\}$ is assigned to this region. The tie-zone is therefore differentiated in several regions according to the labels that could be assigned to them. The Label Merging algorithm is a useful variation of the previous Algorithm 1. Substitute TZ label on lines 15-16 by `MergeLabels($\mathbb{L}[p]$, $\mathbb{L}[v]$)` and the simple label map L by a merged-label map \mathbb{L} that returns a set of labels for each pixel. There is no more one TZ label but so many as the distinct label mergings (four in example of Figure 3.5(b)).

3.3.3 Multipredecessor Optimal Graph

We introduce now a special graph, unique for each image, that will be used in session 3.4.

Roughly speaking, the multipredecessor optimal graph (MOG) $\Gamma = (V, A^*)$ of a weighted graph $G = (V, A, I)$ is the “union” of all the optimal forests $F \in \Phi$. More precisely, it is a directed acyclic

subgraph of G such as its arc set A^* is the set of the optimal oriented arcs, i.e. the union of the (oriented) arcs of all optimal forests $F \in \Phi$.

$$\Gamma = (V, A^*) = (V, \bigcup_{\forall F=(V,A') \in \Phi} A')$$

Note that the graph is unique but is not necessarily a forest. Once we have a lexicographical cost map of the image, i.e. its reconstruction from the seeds and the distance to the border of plateaux, the following local property is valid. A node p is predecessor of a node v in the MOG if and only if p is a neighbour of v such that its lexicographic cost $C_L[p] = (C_1[p], C_2[p])$ is strictly lower than the lexicographic cost $C_L[v] = (C_1[v], C_2[v])$ of v :

$$p \in \mathbb{P}[v] \Leftrightarrow p \in N(v), C_L[p] \prec C_L[v].$$

where $\mathbb{P}[v]$ denotes the set of predecessors of node v .

The number of predecessors by node is no longer restricted to one. But it is bounded by the maximum number of neighbours by node, given by the adjacency definition (e.g., four in the example of Figure 3.7(c)). Furthermore, the number of connected components of the MOG no longer corresponds to the number of seeds as in the case of optimal forest of trees.

An analogy can be observed between the MOG and the ‘lower complete graph’ (definition 3.5 of ref. [RM00]). Both are directed acyclic graphs. In the former case, the lower neighbours in the lexicographic cost map are predecessors of the nodes. In the latter case, only the steepest lower neighbours in the lower complete image are predecessors of the nodes.

3.4 Thinning of the tie-zone based on label frequency

As said in Section 3.1, thin lines (or no line) between segmented regions can be wished. It is why a thinning of the TZ can be useful when the user does not want to add more markers (seeds) to untie a large TZ. Other criteria than path-cost have to untie the TZ to assign this region to existing CBs while preserving the uniqueness property of the TZW. Indeed, the TZ thinning must be unique and independent of the implementation, otherwise it is not worth having firstly applied the TZW. However, as we saw in Section 3.1, there is still a trade-off between uniqueness of the solution and thinness of the TZs, so that perhaps only part of the TZ can be thinned to preserve the uniqueness of the solution.

A first proposal of TZ thinning was introduced in ref. [ALC05]. It consists in applying iteratively the TZW on a topography whose altitude is the number of different labels that tied together. More

labels tied in a region, more disputed it is and higher its altitude is. The topography change creates plateaux with altitude proportional to the number of CBs disputing the tie-zone. By applying such a topography change, the original image is drastically simplified no matter which labels tied and how many times they dispute this region. For example, a CB label may dispute a TZ even if there is only one path reaching the TZ while labels from other CBs have much more paths leading to the same TZ. Consequently, a distance criterion unties the TZ in the next iteration(s) because of the use of lexicographic cost on the created plateaux. One can believe that a unique path to a TZ could be an accidental “leak of water” creating a non-representative TZ. At least, this path has certainly not the same weight as the others in the TZ formation.

This argument motivated the TZ thinning proposed in this section. Indeed, a TZ is created when at least an optimal forest does not assign the same label to a pixel, no matter how many optimal forests are discording with this labelling. Thus, one optimal forest might have as much weight as all the others. We propose to assign the same importance to each optimal forest and keep the most frequent labelling.

Remember that any optimal forest is a solution of the WT by IFT, and is not a biased solution in itself. But the arbitrary choice of one optimal forest would be equivalent to ignoring all the other possibilities of path optimisation in the image-graph. This simplistic choice constitutes a bias. Similarly, giving an equal importance to every label responsible for a TZ would be equivalent to ignoring that perhaps a certain CB is “more frequently linked” to a TZ than other CBs. And this fact reflects the specificity of the image topography, information that should be taken into account.

The TZ thinning proposed here keeps the property of uniqueness and is based on the label frequency: observing all the possible watershed solutions, the TZ can be untied when a region is most frequently assigned to a specific CB. Each optimal forest has the same relevance. Considering the collection of all these possible realisations of the WT, one can deduce the relative frequency for a pixel to be included in a particular CB and finally assign to it the most frequent CB label.

Considering the drop of water paradigm, the TZ corresponds to regions where a drop of water could fall and follow several ways and slide down to different valleys. Imagine now that every time the drop of water can follow several ways, it is split in smaller equal amounts of water which follow the possible ways. We demonstrate that the fractions of water the valleys receive correspond to the respective label frequencies.

Section 3.4.1 demonstrates how relative label frequencies can be computed, by the fragmented drop paradigm (Section 3.4.1) or recursively (Section 3.4.1). Section 3.4.2 explicits the way the most frequent labelling is obtained and special care to take for preserving uniqueness and consistency of the segmentation. Section 3.4.3 presents an algorithm derived from Dijkstra-IFT-TZW algorithm (Algorithm 1) that computes the TZW and the label frequency-based thinning simultaneously.

3.4.1 Computing label frequency

Let $G = (V, A, I)$ be the weighted graph corresponding to an image. Let S be the set of seed nodes (pixels). Let $\Gamma = (V, A^*)$ be the multipredecessor optimal graph (MOG) of G . We denote by $\mathbb{P}(p)$ the set of predecessors of node p in the MOG.

Remember that only one optimal cost map C can result from the IFT of a weighted graph $G = (V, A, I)$. But many predecessor maps, the optimal forests F , can exist and each one is a support for a (distinct or not) label map L when labels are associated with seeds. We wish to find the map of the most frequent labels. We do not search for the most frequent optimal forest (as each distinct optimal forest is observed only once), nor for the forest built with the set of most frequent arcs.

First, we have to compute all the optimal forests $F \in \Phi$ and associate with each pixel the relative frequency of each label. In a final step, the most frequent label will be assigned to each pixel (Section 3.4.2). As we will compute the labels' relative frequency for each pixel considering a collection of optimal forests, we have to know how many optimal forests F exist. Thus, the problem is to count how many optimal forests can derive from Γ . To build an optimal forest from the MOG, we have to choose one and only one predecessor per node except for the optimal tree roots that have no predecessor. As setting a specific predecessor among $\mathbb{P}(p)$ for a node p does not discard choices of predecessor for any other node in the graph, choices of node predecessor are independent events. Therefore, we have $|\Phi|$ possible optimal forests:

$$|\Phi| = \prod_{p \in V, |\mathbb{P}(p)| \neq 0} |\mathbb{P}(p)|$$

The condition $|\mathbb{P}(p)| \neq 0$ is necessary to exclude the tree roots.

Notice that in the label frequency-based thinning, we do not necessarily assign a pixel to the CB from which there are most optimal paths to this pixel. Because this would assume that each linking path is equally frequent for a specific node. But rather, pixel is assigned to the label it is most frequently associated with, when considering all the optimal forests. So, it assumes that each forest is equally frequent, i.e., each possible predecessor arc is equally frequent for a specific node. Figure 3.7 shows that in some cases these assumptions lead to the same result in term of frequency (case of I_6), in general they do not (cases of I_7 and I_8).

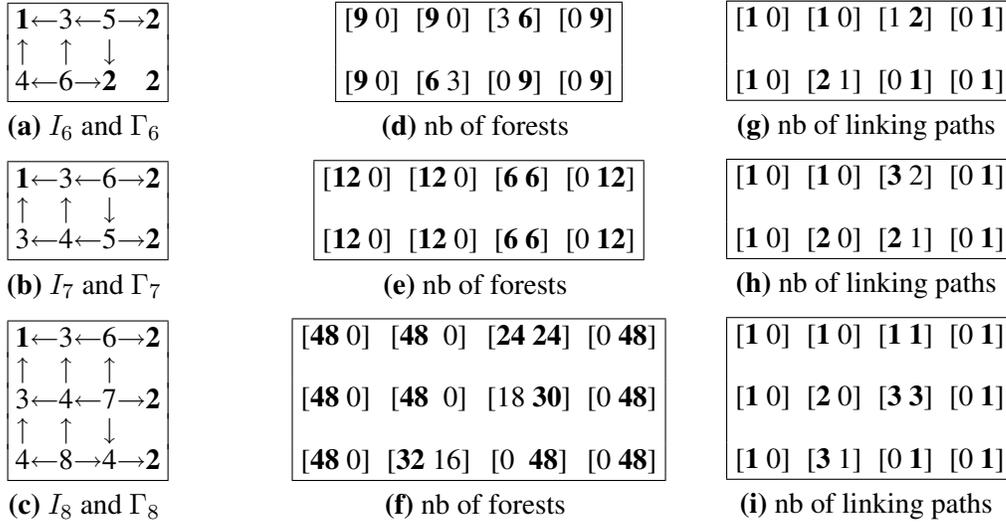


Fig. 3.7: Difference between frequency of forests (or predecessor arcs) and linking paths. (a)-(c): Images and their respective MOG (minima are bold-faced). (d)-(f): For each pixel, $[x y]$ are the numbers of optimal forests in which the pixel is labelled with 1 and 2 respectively. (g)-(i): For each pixel, $[x y]$ are the numbers of paths linking the pixel with the minimum 1 and 2 respectively.

Fragmented drop paradigm

Let us determine the relative frequency $f_p(\lambda)$ of a pixel p being associated with a label λ using the definition of frequency normalised by the number of observations (optimal forests):

$$f_p(\lambda) = \frac{|\Phi_{\pi(p, \lambda)}|}{|\Phi|} \quad (3.1)$$

where $|\Phi_{\pi(p, \lambda)}|$ corresponds to the number of optimal forests that include a (“descending”) path between pixel p and a root with label λ (necessary condition for labelling p with λ).

Let $\Pi(p, \lambda)$ be the set of the paths $\pi(p, \lambda)$ in Γ that link p to a root with label λ . Equation 3.1 is equivalent to:

$$f_p(\lambda) = \frac{\sum_{\pi \in \Pi(p, \lambda)} |\Phi_\pi|}{|\Phi|} \quad (3.2)$$

A path π can be written in this form: $\pi = \langle q_n, q_{n-1}, \dots, q_2, q_1 \rangle$ where q_{i-1} is predecessor of q_i . Let π' be the set of successor vertices of the arcs contained in path π : $\pi' = \{q_n, q_{n-1}, \dots, q_2\}$. The number of optimal forests that contain a particular path π is given by:

$$|\Phi_\pi| = \prod_{p \notin \pi', |\mathbb{P}(p)| \neq 0} |\mathbb{P}(p)| = \frac{\prod_{p \in V, |\mathbb{P}(p)| \neq 0} |\mathbb{P}(p)|}{|\mathbb{P}(q_n)| |\mathbb{P}(q_{n-1})| \dots |\mathbb{P}(q_2)|} = \frac{|\Phi|}{\prod_{q \in \pi'} |\mathbb{P}(q)|}$$

because each pixel q_i in π' must have the specified predecessor q_{i-1} whereas all the other pixels p can have any one of the possible predecessors in $|\mathbb{P}(p)|$. From Eq. 3.2 it follows that:

$$f_p(\lambda) = \frac{\sum_{\pi \in \Pi(p, \lambda)} \frac{|\Phi|}{\prod_{q \in \pi'} |\mathbb{P}(q)|}}{|\Phi|} \quad (3.3)$$

And finally, we obtain:

$$f_p(\lambda) = \sum_{\pi \in \Pi(p, \lambda)} \frac{1}{\prod_{q \in \pi'} |\mathbb{P}(q)|} \quad (3.4)$$

Therefore, for each distinct (“descending”) path between p and a root with label λ , it is necessary to calculate the inverse of the product of predecessor numbers for each node of the path. After summing all the obtained results, we get the relative frequency of having a path “between p and label λ ”, the whole set of optimal forests being observed. We saw in the introduction of this paper (Section 3.1) that, considering the image as a topography, a drop of water falling at a locus (pixel) that belongs to a CB will slide down to the corresponding valley (minimum). Otherwise, it means that the locus belongs to the watershed and the drop of water can slide down to several valleys. Equations 3.1 and 3.4 demonstrate that computing the label frequency for a pixel when all the optimal forests are observed is equivalent to evaluating the amount of the drop of water that reaches the respective labelled CB. Indeed, in Eq. 3.4, the label frequency is equal to the sum of the drop fragments considering all the possible paths between the pixel and the respective CB. Every time the sliding fragmented drop passes through a pixel with many predecessors, the amount of water at that locus is divided in equal parts among all the predecessors to continue the descent until minima (see Figure 3.8). For a pixel in CB, the drop will entirely arrive in the same minimum. For a pixel in the TZ, the drop of water is broken up into at least two minima. In section 3.4.2, we will see that the frequency-based thinning assigns the pixel to the label whose minimum caught the maximum fraction of the drop of water. And it is equivalent to assigning the most frequent label after observing all the labelled optimal forests.

Recursive computation by immersion

We demonstrate now that Eq. 3.4 corresponding to the fragmented drop paradigm is equivalent to a recursive formula (Eq. 3.5) corresponding to an immersion simulation.

In Eq. 3.4 where it appears a sum in relation to the different paths π between p and λ , let us group the paths depending on whether they pass through one or another of the $|\mathbb{P}(p)|$ predecessors v of p —as any path linking p to λ is necessarily the concatenation of the arc $\langle p, v \rangle$ with a path linking v to



Fig. 3.8: Drop of water is sliding down from a TZ pixel and broken up in equal parts whenever it meets an intersection of descending paths (image I_8). (a): Descending paths encountered by the drop from pixel 7 and respective fractions of water sliding down. Minima 1 and 2 catch respectively $3/8$ and $5/8$ of the drop. This result fits with the proportion of optimal forests leading to each labelling (18 and 30, see Figure 3.7(f)). (b): Idem for pixel 8: $2/3$ and $1/3$ are in the same proportion as 32 and 16.

λ. So, grouping according to possible predecessors v and factoring the sum, we get:

$$f_p(\lambda) = \sum_{v \in \mathbb{P}(p)} \sum_{\pi \in \Pi(v, \lambda)} \left[\frac{1}{|\mathbb{P}(p)| \prod_{q \in \pi'} |\mathbb{P}(q)|} \right] = \frac{1}{|\mathbb{P}(p)|} \sum_{v \in \mathbb{P}(p)} \left[\sum_{\pi \in \Pi(v, \lambda)} \frac{1}{\prod_{q \in \pi'} |\mathbb{P}(q)|} \right]$$

According to Eq. 3.4, replace the expression between brackets:

$$f_p(\lambda) = \frac{1}{|\mathbb{P}(p)|} \sum_{v \in \mathbb{P}(p)} f_v(\lambda) \quad (3.5)$$

As we can see, this recursive formula implicitly describes an immersion process where label frequencies are first computed at lower levels (predecessors v) and then at higher levels (p). Note also that the normalisation term allows to take into account all the possible predecessors. In summary, we have three ways of computing label frequency, as Eq. 3.5 is equivalent to Eqs. 3.1 and 3.4. In practice, Eq. 3.5 is used by the frequency-based thinning algorithm proposed in Section 3.4.3.

3.4.2 Computing the map of most frequent labels

Now we have computed the frequencies of each label to be assigned to each pixel, we want to choose the most frequent label for each pixel (see Figure 3.9(a)). To obtain a unique label map, we cannot decide which of two or more labels is assigned to a pixel in case of equal frequency (EF). Indeed, there can be many EF pixels. They are pixels where at least two labels have the same maximum frequency of being assigned to (see Figures 3.9(c)(d)). They constitute the first type of pixel that cannot be thinned with the maximum frequency criterion.

pixel	$f(A)$	$f(B)$	$f(C)$	$L(\text{pixel})$	$P(\text{pixel})$
q_1	0.5	0.3	0.2	A	...
q_2	0.5	0.3	0.2	A	...
q_3	0.2	0.3	0.5	C	...
p	0.4	0.3	0.3	A	q_1 or q_2

(a)

pixel	$f(A)$	$f(B)$	$f(C)$	$L(\text{pixel})$	$P(\text{pixel})$
q_1	0.5	0.4	0.1	A	...
q_2	0.4	0.5	0.1	B	...
q_3	0.2	0.2	0.6	C	...
p	0.37	0.37	0.26	EF	NIL

(c)

pixel	$f(A)$	$f(B)$	$f(C)$	$L(\text{pixel})$	$P(\text{pixel})$
q_1	0.6	0.0	0.4	A	...
q_2	0.2	0.5	0.3	B	...
q_3	0.1	0.5	0.4	B	...
p	0.3	0.33	0.37	ISO	NIL

(b)

pixel	$f(A)$	$f(B)$	$f(C)$	$L(\text{pixel})$	$P(\text{pixel})$
q_1	0.5	0.1	0.4	A	...
q_2	0.2	0.5	0.3	B	...
q_3	0.1	0.5	0.4	B	...
p	0.26	0.37	0.37	ISO	NIL

(d)

Fig. 3.9: Computing the most frequent label. Pixels q_1, q_2, q_3 are possible predecessors of p . Label frequencies f of p are computed according to the recursive formula. Then, label L and predecessor P are assigned when possible. (a): Most frequent label for p is A and two predecessors are possible (see Eq. 3.6). (b): Isolated (ISO) pixel case. Label C is the most frequent for p but no predecessor has the same label (see Eq. 3.7). (c): Equal frequency (EF) case between labels A and B. No predecessor can be chosen (see Eq. 3.8). (d): Other isolated pixel case. Equal frequency case between labels B and C but no predecessor is labelled with C (see Eq. 3.9).

A second type of pixel cannot be labelled: the isolated (ISO) pixels. To ensure that a segmentation is achieved, the labelling must admit at least an optimal forest as its support. And we saw that this forest is necessarily made of the possible arcs of the MOG. If such a support does not exist, it means that the labelling is not a segmentation because some pixels can be disconnected from their seeds. So, the case of isolated pixel occurs when no arc can link the labelled pixel with another one of same label (see Figures 3.9(b)(d)). Figure 3.10 shows the frequency-based thinning of an image with EF and ISO cases.

Observe that the frequency of each label assignment is not a monotonically increasing function. If a label is most frequently assigned to a pixel, it does not mean that it is most frequently assigned to its successor in the path: its frequency may decrease at the expense of other assignment frequencies.

In conclusion, the result of the frequency-based thinning is a thinning of the TZ consisting of assigning pixels of the TZ to the most frequent CB's label when it is unique and when there exists a path linking to the respective CB root within the CB. Otherwise, the pixels have undefined (EF or ISO) labels. Thus, the labelled pixels have a support made of minimum cost trees while some pixels (EF and ISO) are disconnected from these trees. The solution is not necessarily an optimal forest because of these special cases.

Hereafter are presented the conditions (rules) of the labelling process for each pixel p , depicted in Figures 3.9(a)-(d) respectively. The most frequent label map is denoted by L and the predecessor map by P (it is only one of its possible supports). To summarise, when the most frequent label for a node

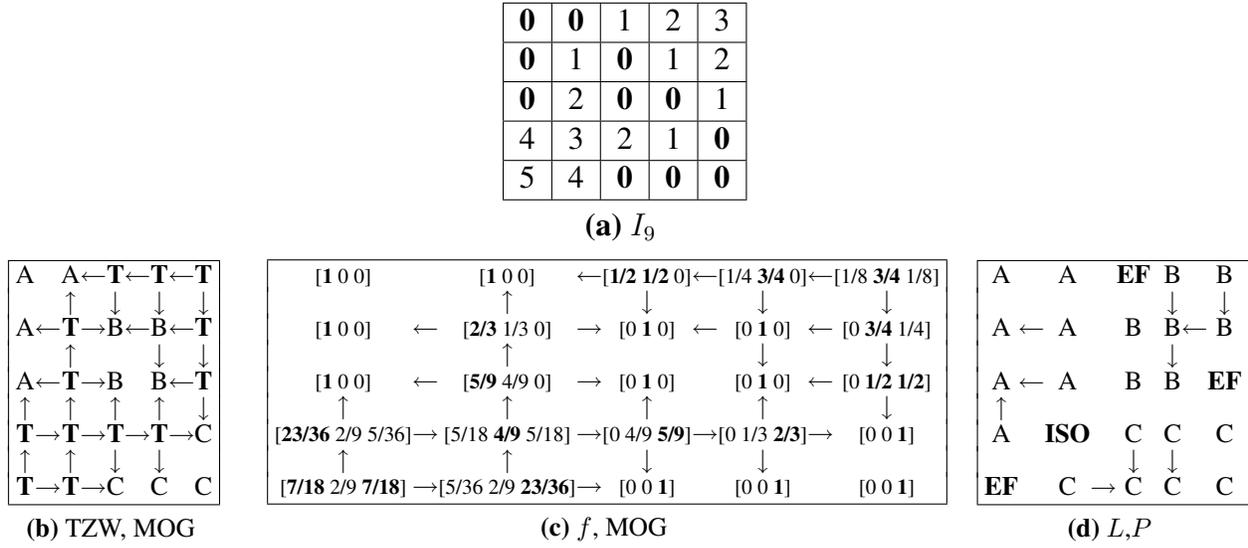


Fig. 3.10: (a): Input greyscale image with 3 minima (in bold). (b): Multipredecessor Optimal Graph (arrows to predecessors) and TZW (label T is tie-zone). (c): MOG and frequency of label assignments. Each vector represents the frequencies of labelling the pixel with A, B and C respectively. Maximum frequency is bold-faced. (d): Resulting label map L and one of its supports P . Observe the cases of equal frequency (EF) and isolated pixel (ISO).

is unique and at least one of its predecessors is assigned to this label, the label is propagated from the predecessor to this node (rule (3.6)). But if no predecessor is assigned to this most frequent label, the node is isolated (rule (3.7)). When there are many most frequent labels for a node and each of them is assigned to at least one of the predecessors, the node is an equal frequency case (rule (3.8)). But if at least one of the equally most frequent labels is not present among the predecessors, the node is considered isolated (rule (3.9)).

$$\left. \begin{array}{l} \exists \lambda_{ML}, \forall \lambda \neq \lambda_{ML}, f_p[\lambda_{ML}] > f_p[\lambda] \\ \exists q \in \mathbb{P}[p], L[q] = \lambda_{ML} \end{array} \right\} \implies \left\{ \begin{array}{l} L[p] \leftarrow \lambda_{ML} \\ P[p] \leftarrow q \end{array} \right. \quad (3.6)$$

$$\left. \begin{array}{l} \exists \lambda_{ML}, \forall \lambda \neq \lambda_{ML}, f_p[\lambda_{ML}] > f_p[\lambda] \\ \forall q \in \mathbb{P}[p], L[q] \neq \lambda_{ML} \end{array} \right\} \implies \left\{ \begin{array}{l} L[p] \leftarrow \text{ISO} \\ P[p] \leftarrow \text{NIL} \end{array} \right. \quad (3.7)$$

$$\left. \begin{array}{l} \exists \Lambda_{ML}, \forall \lambda_{ML} \in \Lambda_{ML}, \forall \lambda \neq \lambda_{ML}, f_p[\lambda_{ML}] \geq f_p[\lambda] \\ \forall \lambda_{ML} \in \Lambda_{ML}, \exists q \in \mathbb{P}[p], L[q] = \lambda_{ML} \end{array} \right\} \implies \left\{ \begin{array}{l} L[p] \leftarrow \text{EF} \\ P[p] \leftarrow \text{NIL} \end{array} \right. \quad (3.8)$$

$$\left. \begin{array}{l} \exists \Lambda_{ML}, \forall \lambda_{ML} \in \Lambda_{ML}, \forall \lambda \neq \lambda_{ML}, f_p[\lambda_{ML}] \geq f_p[\lambda] \\ \exists \lambda_{ML} \in \Lambda_{ML}, \forall q \in \mathbb{P}[p], L[q] \neq \lambda_{ML} \end{array} \right\} \implies \left\{ \begin{array}{l} L[p] \leftarrow \text{ISO} \\ P[p] \leftarrow \text{NIL} \end{array} \right. \quad (3.9)$$

Remember that the frequency-based thinning uses frequencies of label assignment given by the set of optimal forests and alternatively, these frequencies can be viewed as the fractions of a divisible drop of water sliding down from pixel to minima. Under this approach, the frequency-based thinning assigns to each pixel the label of the minimum that caught the greatest fraction of the drop of water if there exists a descending path from the pixel to the minimum that is entirely included in the CB.

3.4.3 Dijkstra-IFT-based algorithm for the label-frequency thinning of the tie-zone

Now the frequency criterion was defined, an algorithm to thin the TZ is proposed (see Algorithm 2). It is based on Algorithm 1. It has therefore the same structure and needs the same input (image-graph (V, A, I) , seeds S , labelling λ and handicap h functions) and the same ordered queue and cost maps (C_1, C_2) . Observe that instead of predecessor and label maps, there are multipredecessor set map \mathbb{P} and label set map \mathbb{L} for internal processing. Frequencies f , most frequent label map L (the frequency-based thinning in output) and its support P (a predecessor map) are computed during the main loop (i.e. the emptying of the ordered queue). The lines with bold-faced number are specific of the frequency-based thinning algorithm and differ from the Label Merging algorithm mentioned in Section 3.3.2.

Let us comment some distinctive particularities of the frequency-based thinning algorithm. In line 1, the frequency map is also initialised with 0. In line 5, whenever a pixel is removed from the ordered queue, the frequency map has to be updated (`UpdateLabelFreq` function). Its definitive frequencies are computed from the frequencies already assigned to its predecessors (the already processed neighbours).

The `UpdateLabelFreq` function (lines 22–28) corresponds to rules (3.6)–(3.9). If only one label can be assigned to a pixel (line 22), frequency of this labelling is 1. A predecessor is assigned but is not unique. If many labels can be assigned (line 23), the frequencies of each labelling are computed from all the predecessors' frequencies by using the recursive formula. Lines 24–26 describe the case of one unique label of maximum frequency. If it has no predecessor with same label (line 26), it is an isolated pixel. Lines 27–28 describe the case of at least two labels of maximum frequency: equal frequency case EF. But if one of the equally most frequent label has no predecessor with same label (line 28), the pixel is considered isolated in some sense.

Coming back to the main loop of the algorithm, we analyse the case of cost equality (former TZ case): when proposed cost is equal to current cost in cost map (lines 13–21). Notice that whenever the sets of labels are different (lines 14–17) or equal (lines 18–21), a new predecessor must be added in the multipredecessor set. The label merging is actually necessary only when the propagating label

set is not already included to the current label set: $\mathbb{L}[v] \cup \mathbb{L}[p] \neq \mathbb{L}[p]$. Elsewhere, the label merging is redundant. And avoiding it is desirable. According to the implementation of these sets, the above criterion of non-inclusion can be more costly than a simple inequality operation. The proposed algorithm uses this alternative criterion ($\mathbb{L}[v] \neq \mathbb{L}[p]$ in line 14) that can allow unnecessary label merging. This choice of criterion will depend on the cost of merging, inclusion and equality test operations on label sets. Besides, observe that the implementation difficulty is with managing the sets \mathbb{L} and \mathbb{P} . The size of \mathbb{P} is variable but limited by the number of neighbours. Implementation of set \mathbb{L} is more problematic: its size depends on the number of labels that tied together. The propagation of the frequencies to the neighbours are another related problem.

Notice that the computations of frequencies, most frequent label map and its support could be integrated in the IFT loop because of the recursive formula for frequencies and the ordered computing: frequencies and label assignment of a pixel only depend on its lower neighbours.

whereas TZW is in Figure 3.11(b). White areas are the tie-zone and grey ones are the CBs. Each CB corresponds to a regional minimum. Observe that in practice, the detection of the markers (set of the seeds with same label) should be achieved manually or automatically by filtering, for example, the minima according to their dynamics [Gri92, Mey96]. Here, the minima were not filtered on purpose. This allows to visualise the location of large TZ and the behaviour of the proposed thinning in this case. The quite large TZ areas of Figure 3.11(b) disappeared in Figure 3.11(c): the frequency-based thinning was applied. We can see that some pixels (in white) remain undefined (EF or ISO cases). Most of them are on the frontier between CBs and constitute thin segments of line. However, two regions are still thick. They can be visualised in the detailed view of Figure 3.11(d). Grey CBs are constituted of optimal trees (in black) rooted to regional minima (nodes contoured by a square) while EF pixels (in white with a central dot) are disconnected of any tree. Most ISO pixels (in white with a central black square) occur by transitivity, because their predecessors are already EF pixels without defined label. Even if they have a most frequent label, there is no labelled descending path to the corresponding CB. EF pixels are blocking their way. Therefore, ISO pixels are also disconnected of any tree.

Note that there may be cases where the frequency-based thinning does not thin TZ. For example, symmetric images whose topography is similar to pyramid or cone may contain only EF pixels and ISO pixels (by transitivity) in their tie-zone.

Figure 3.12 shows one of the optimal forests defined by the WT by IFT. It was randomly obtained from the multipredecessor optimal graph. If we compare this figure with Figures 3.11(b) and (c), the catchment basins are the same, out of the tie-zone. Variations occur in the tie-zone: TZW does not make any arbitrary decision, the thinning unties the TZ according to the frequency criteria and the random optimal forest makes arbitrary decisions. Observe the important differences between the thinning and the random IFT in the tie-zone: e.g., (white) tie-zones in the top right border, near the bottom left corner, on the contours of the wings and in the rear and front parts of the airplane.

Figure 3.13 shows the results given by an implementation of the Vincent and Soille's algorithm for four different image scanings. There are many little differences in the segmentations on the contours, in the wings, and in many other catchment basins of the background. Note that other implementations could choose other internal processing orders and possibly give other solutions than these ones. Observe that the watershed line becomes thicker in the front regions of the airplane. The frequency-based thinning returns EF/ISO labels in these regions. The minima in these regions should be filtered or redefined to remove the tie-zone.

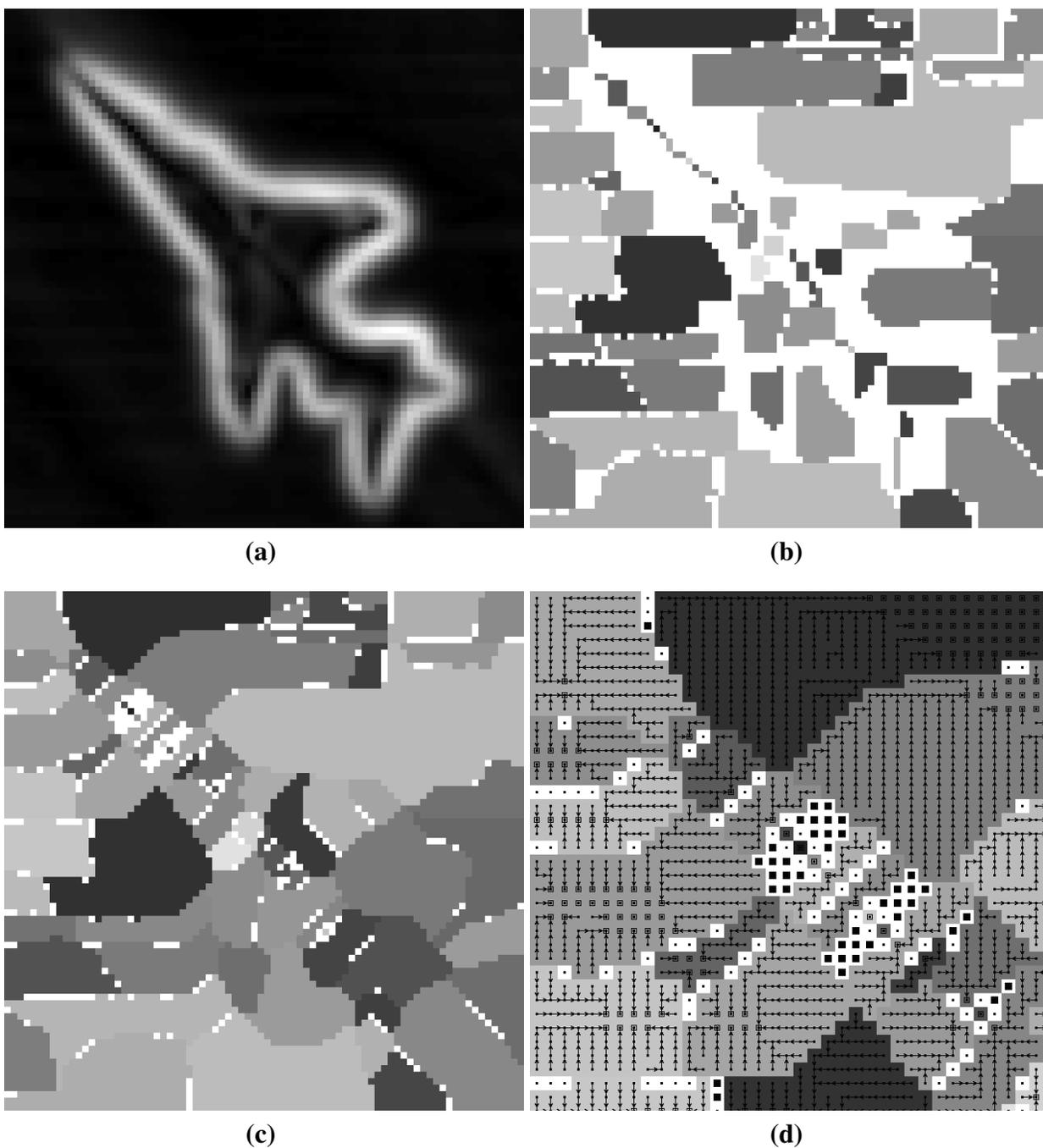


Fig. 3.11: **(a)** Input gradient image of an airplane. **(b)** TZW (TZ in white). **(c)** Frequency-based thinning (EF and ISO pixels in white). **(d)** Detail of the frequency-based thinning showing optimal trees in CBs, EF (white with dot) and ISO (white with black square) pixels. The seeds (nodes contoured by a square) represent the regional minima of the gradient image.

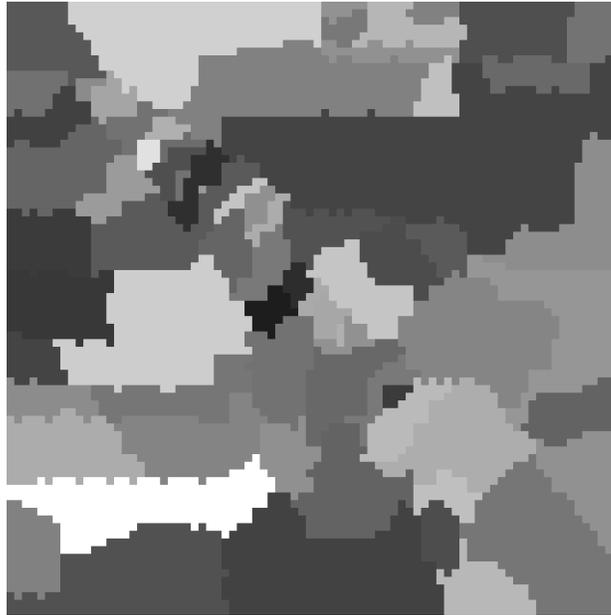


Fig. 3.12: WT corresponding to a random optimal forest (one of the solution of WT by IFT).

3.5 Conclusion

In this paper, we saw that there are many watershed transform (WT) definitions in both continuous and digital spaces and lots of algorithms to compute them. But there is an inadequacy between some algorithms and their respective definition because they impose a watershed solution containing either separating watershed lines, or only regions (catchment basins). Consequently, such algorithms may return different solutions depending on the implementation because they introduce a bias in relation to the definitions. Other definitions like the WT by image foresting transform (IFT) admit several solutions. The corresponding algorithms respect their definition but introduce a bias by choosing one of the multiple solutions. Again, the solution is not unique and depends on the implementation. The tie-zone watershed (TZW) returns a unique and unbiased solution by considering all the possible solutions. As large tie-zones may sometimes appear, we proposed a label frequency-based thinning that leads to a unique solution. If we define our space of observation as the set of the possible solutions (optimal forests), we can compute the frequency of the labels associated with each pixel. The thinning consists in assigning the most frequent label while preserving the segmented region connectivity, i.e., each labelled pixel must be connected to a root by a path whose pixels are all associated with the same label. Notice that the thinning does not necessarily correspond to an optimal forest: some pixels can be simply disconnected from the trees (i.e., unlabelled) because assigning the most frequent label to them would violate the segmented region connectivity. We demonstrated that the label frequency computation can also be explained by the fragmented drop paradigm: a drop of water can be

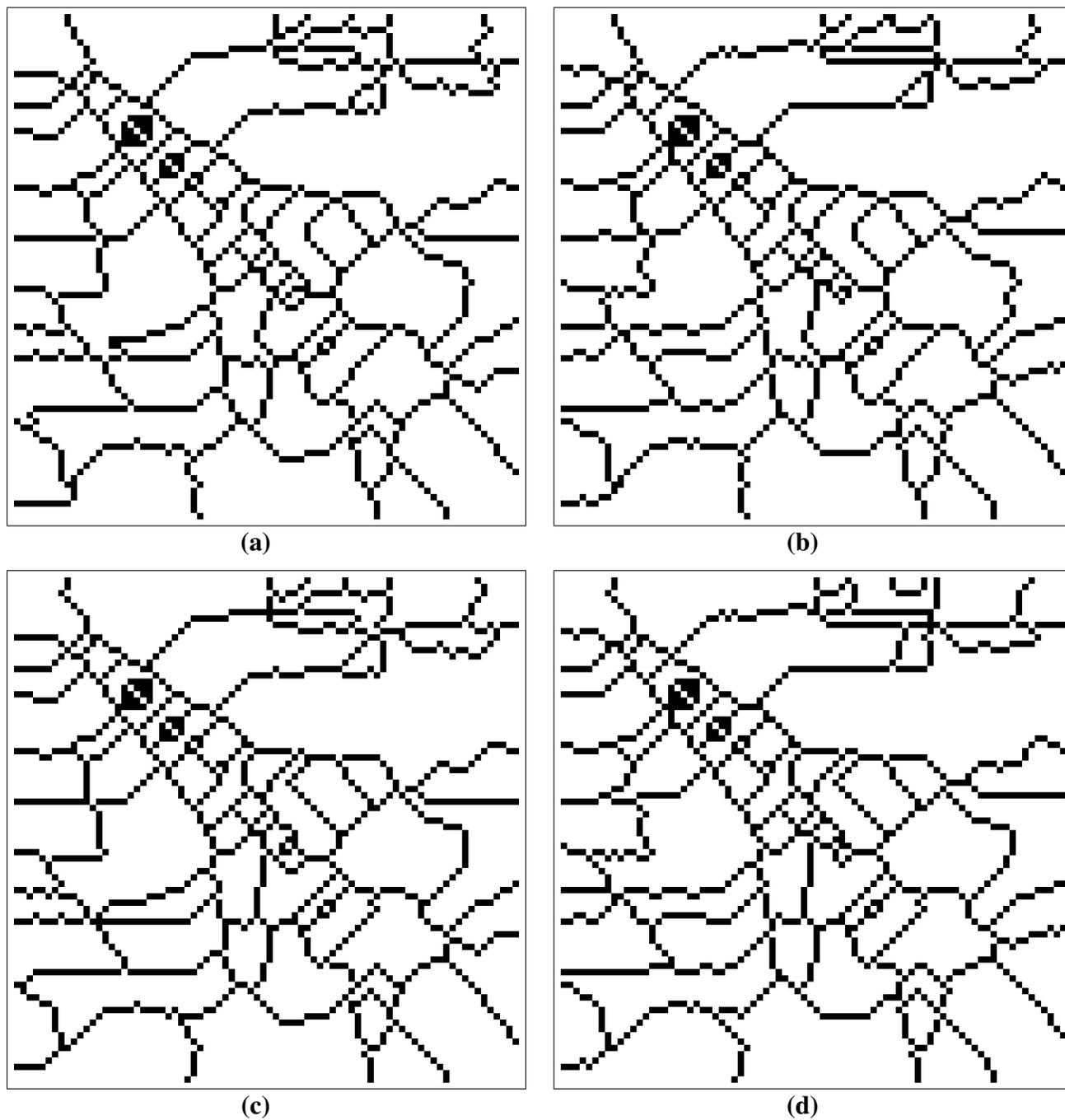


Fig. 3.13: (a)-(d) WT by Vincent-Soille's algorithm (watershed lines in black): four different image scannings.

fragmented and go on until several minima. The fraction of water in each minimum corresponds to the frequency of assigning the corresponding label to the pixel on which the drop fell down. Alternatively, the label frequency can be computed by a recursive formula that simulates an immersion. Finally, we proposed an algorithm that computes at the same time the TZW, the label frequencies and the frequency-based thinning.

Acknowledgements

This work is supported by CAPES. We thank the anonymous reviewers for their comments and suggestions.

Capítulo 4

Transformadas de *Watershed*

Este capítulo contém o artigo intitulado *Watershed by image foresting transform, tie-zone, and theoretical relationships with other watershed definitions* [AL07c]. Nele são analisadas várias definições discretas de transformadas de *watershed* para entender melhor as diferenças das suas soluções. O pivô central que permite relacioná-las (cf. a Figura 1.1) é a transformada de *watershed* via IFT (IFT-WT) e sua transformada em zona de empate (TZ-IFT-WT).

Demonstra-se que (i) a transformada de *watershed* baseada em distância topográfica (TD-WT) corresponde à transformada em zona de empate da transformada de *watershed* baseada em condição local (LC-WT); (ii) a linha de *watershed* possivelmente espessa e não totalmente separadora da TD-WT é contida na zona de empate da TZ-IFT-WT-lex (com custo lexicográfico), enquanto (iii) as bacias de retenção da TD-WT contém as bacias da TZ-IFT-WT-lex; (iv) toda solução da LC-WT também é solução da IFT-WT-lex; (v) toda solução da transformada de *watershed* baseada em floresta geradora mínima (MSF-WT) também é solução da IFT-WT (sem custo lexicográfico).

Portanto, além de única e consistente, a TZ-IFT-WT revela ser uma transformada “segura” ou “confiável”, no sentido em que ela é o denominador comum entre todas essas definições de *watershed*. Nenhuma das transformadas contesta os segmentos (bacias) que ela retorna. As diferenças entre elas ocorrem apenas dentro da zona de empate da TZ-IFT-WT.

Estamos escrevendo atualmente a extensão desses resultados para a transformada de *watershed* por inundação (Flood-WT). Embora pareça bem diferente por ser definida apenas como um processo recursivo e não ter definição direta ou critério de otimalidade global, esta também define bacias que contêm as bacias da TZ-IFT-WT-lex, e uma linha de *watershed* espessa e não separadora inclusa na zona de empate da TZ-IFT-WT-lex.

Abstract

To better understand the numerous solutions related to watershed transform (WT), this paper shows the relationships between some discrete definitions of the WT: the watershed based on image foresting transform (IFT), on topographic distance (TD), on a local condition (LC), and on minimum spanning forest (MSF). We demonstrate that the tie-zone (TZ) concept, that unifies the set of multiple solutions of a given WT, when applied to the IFT-WT, includes all the solutions predicted by the other paradigms. More precisely, the watershed line (possibly thick) of TD-WT is contained in the TZ of the IFT-WT, while the catchment basins of the former contain the basins of the latter. In addition, the TD-WT can be seen as the tie-zone transform of the LC-WT. Furthermore, any solution of LC-WT or MSF-WT is also solution of the IFT-WT.

4.1 Introduction

The *watershed transform (WT)* is a famous and powerful segmentation tool in morphological image processing. First introduced by Beucher and Lantuéjoul [BL79] for contour detection and applied in digital image segmentation by Beucher and Meyer [BM93], it is inspired from a physical principle well-known in geography: if a drop of water falls on a topographic surface, it follows the greatest slope until reaching a valley. The set of points which lead to the same valley is called a (*catchment*) *basin*. *Watershed lines* separate different basins. In the WT, an image is seen as a topographic surface where gray level corresponds to altitude. In practice, the topography is made of a gradient of the image to segment. In this case, it is expected that a region with low gradient, a valley, corresponds to a rather homogeneous region and possibly to the same object. Ideally, basins correspond to segmented objects separated by watershed lines.

Many definitions and numerous algorithms for WT exist in literature. Furthermore, multiple WT solutions are sometimes returned by an algorithm according to its implementations or even by the theoretical definition itself. This disconcerting fact motivated the investigation of the relationships between theoretical WT definitions.

Definitions in continuous space have been proposed [BL79, Mey94b, NS94, Prê93] and consider the watershed as a skeleton by influence zones (SKIZ) generalized to gray-scale images. In discrete space (of interest in this paper), there are many definitions which can be classified in five main paradigms.

The *WT based on local condition (LC-WT)* mimics the intuitive drop of water paradigm. The inclusion of a pixel to a basin is achieved by iteratively respecting a local condition of label continuity along a path of steepest descent that reaches the basin minimum. It is why this definition includes al-

gorithms of “arrowing”, “rain simulation”, “downhill”, “toboggan”, “hill climbing” [RM00, LTHS06, RLLV07]. The variation among them is due to processing strategy (ordered or unordered data scanning, depth- or breadth-first, union-find) and data structure.

The *WT based on flooding* has a recursive definition [VS91] that simulates the immersion of a topography representing the image. At each flooding level, growing catchment basins invade flooded regions that belong to their respective influence zone. The watershed corresponds to the SKIZ.

The *topological WT* [CB97a] cannot be viewed as a generalized SKIZ but in fact, as the ultimate homotopic thinning that transforms the image while preserving some topological properties as the number of connected components of each lower cross-section and the saliency between any two (basin) minima.

The *WT based on path-cost minimization* associates a pixel to a catchment basin when the topographic distance is strictly minimum to the respective regional minimum in the case of the *WT by topographic distance (TD-WT)* [Mey94b]; or it builds a forest of minimum-path trees, each tree representing a basin, in the case of the *WT by image foresting transform (IFT-WT)* [LF00, FSL04].

The *WT based on minimum spanning forest (MSF-WT)* associates a graph to an image and builds a MSF [Mey94a], i.e. a spanning forest minimizing the sum of the weights of the arcs used for its construction. Trees correspond to basins.

Table 4.1 summarizes some characteristics of these WT definitions. Only flooding-WT and TD-WT definitions (not the related algorithms) return unique solution (Figure 4.1(b)(i)), but the concept of tie zone (TZ) can be applied to the IFT-WT to unify the set of multiple solutions by creating litigious zones when solutions differ.

The LC-WT, IFT-WT (Figure 4.1(e)-(h)) and MSF-WT, are sometimes called “region”-WT because all pixels are assigned to basins, by definition. Watershed lines are considered as located between basin pixels, but can be visualized by *ad-hoc* algorithms. The other definitions are known as “line”-WT because some pixels are labeled as watershed. Yet, except for the topological WT definition (Figure 4.1(c)(d)), they do not define lines that consistently separate basins but, instead, possibly thick and disconnected watershed lines.

Observe that among these paradigms, TD-WT, IFT-WT and MSF-WT are based on a global optimality criterion. Both IFT-WT and MSF-WT are only defined in discrete space. The other paradigms attempt to mimic a continuous definition, i.e. they may be defined in both discrete and continuous spaces.

This paper shows the relationships between the discrete definitions of IFT-WT, TD-WT, MSF-WT and LC-WT. We show that the TZ watershed, derived from the solutions of the IFT-WT, contains all the solutions predicted by the other paradigms.

In Section 4.2, we present the IFT-WT formalism, and the TZ concept. Section 4.3 recalls the

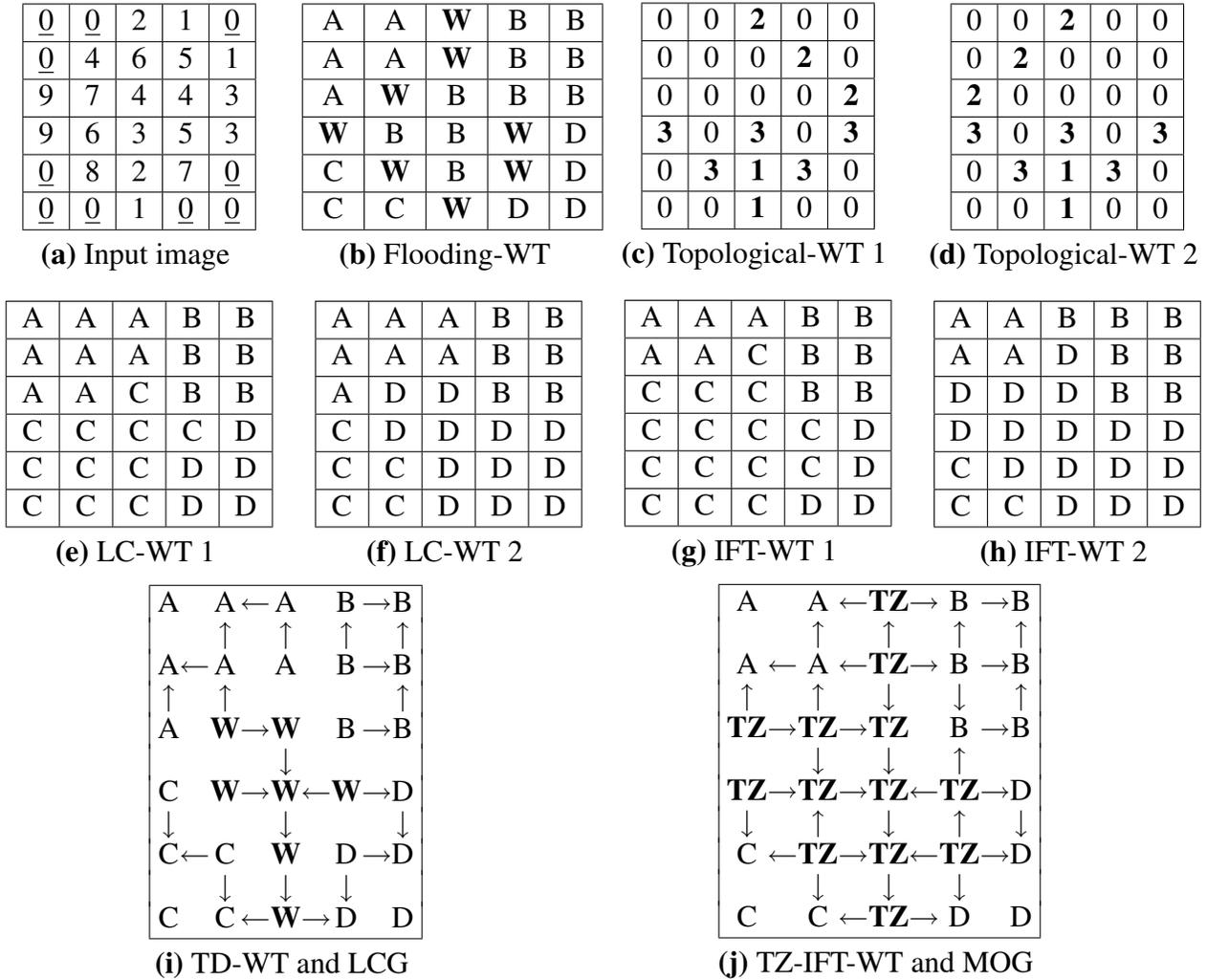


Fig. 4.1: (a): Lower-complete input grayscale image ($I = I_{LC}$) with four minima. (b)–(j): Its WTs using 4-adjacency, according to definitions from literature. Label map is shown (**W** represents watershed line and **TZ** tie-zone) except for topological WT where watershed lines are valued. (c)–(h) show only two of the possible solutions. Watershed line in (b)(i) is not separating. Arrows (pointing to predecessors) represent the lower complete graph (i) and multipredecessor optimal graph (j).

Tab. 4.1: Characteristics of the main watershed transform (WT) definitions.

<i>Watershed definitions</i>	Unique solution	Watershed pixels	Separating lines	Thin lines	Grayscaled lines
LC-WT	no	no	—	—	—
Flooding-WT	yes*	yes	no	no	no
Topological-WT	no	yes	yes	no	yes
TD-WT	yes*	yes	no	no	no
IFT-WT	no	no	—	—	—
TZ-IFT-WT	yes	tie-zone	no	no	no
MSF-WT	no	no	—	—	—

* The strict definitions have a unique solution but the algorithms derived in [BM93, VS91] do not respect the definitions and, therefore, return multiple solutions.

definition of LC-WT and demonstrates that any solution of LC-WT is also solution of the IFT-WT. Section 4.4 shows that the watershed region of TD-WT is contained in the TZ (derived from the IFT-WT), and the basins of the former contain the basins of the latter. In addition, the TD-WT can be seen as the TZ transform of the LC-WT. Finally, Section 4.5 demonstrates that any solution of MSF-WT is also solution of the IFT-WT.

4.2 The Image Foresting Transform (IFT)

The IFT is a general framework based on graph theory in which an image is seen as a graph and pixels (or voxels) as its nodes. This transform returns a *shortest path forest* (SPF) from an input image-graph. Depending on the path-cost function utilized and other input parameters (adjacency, arc weights), the IFT can compute different image processing operations [FSL04, FCL01]: distance transforms, connected filters, interactive object delineation (“live-wire”), segmentation by fuzzy connectedness [AL06] and segmentation by watershed.

4.2.1 Watershed by Image Foresting Transform (IFT-WT)

Under the IFT framework, an *image* is interpreted as a weighted graph $G = (V, A, w)$ consisting of a set V of *nodes* or *vertices* that represent image pixels, a set A of *arcs* weighted by w , a function from A to some nonnegative scalar domain. $N(v)$ denotes the *neighborhood* of node v , i.e. the set of nodes adjacent to it. Nodes u and v are adjacent when the arc $\langle u, v \rangle$ belongs to A . A graph (V', A') is *subgraph* of (V, A) if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$. A *forest* F of G is an acyclic subgraph F of G . *Trees* are connected components of the forest (any two nodes of a tree are connected by a

path). A *path* $\pi(u, v)$ from node u to node v in graph (V, A, w) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said *simple* if all its nodes are different from each other. A path with terminal node v is denoted by π_v . The path π_v is *trivial* when it consists of a single node $\langle v \rangle$. Otherwise, it can be defined by a path resulting from the concatenation $\pi_u \cdot \langle u, v \rangle$. A *path-cost function* f assigns to each path π a path cost $f(\pi)$, in some totally ordered set of cost values.

Let $S \subseteq V$ be a set of particular nodes s_i called *seeds*. For a given weighted graph (V, A, w) and a set S of seeds, the *image foresting transform (IFT)* returns a forest F of (V, A, w) such that (i) there exists for each node $v \in V$ a unique and simple path $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path is *optimum*, i.e., has a minimum cost for linking v to some seed of S , according to the specified path-cost function f . In other words, the IFT returns a shortest (cheapest in fact) path forest (SPF), also called *optimal forest* in this paper, where each tree is rooted to a seed. Although path costs are uniquely defined, the IFT may return many optimal forests because many paths of same minimum cost may exist for some nodes.

The *watershed transform by IFT (IFT-WT)* assumes that (i) the seeds correspond to regional minima of the image (or to imposed minima, i.e. markers [BM93]); (ii) the *max-arc* path-cost function f_{\max} is used:

$$\begin{aligned} f_{\max}(\langle v \rangle) &= h(v) \\ f_{\max}(\pi_u \cdot \langle u, v \rangle) &= \max \{ f_{\max}(\pi_u), w(u, v) \} \end{aligned} \quad (4.1)$$

where $h(v)$ is a fixed but arbitrary handicap cost [LFZ02] for any paths starting at pixel v , and $w(u, v)$ is the weight of arc $\langle u, v \rangle \in A$, ideally higher on the object boundaries and lower inside the objects. Usual arc weight functions are: $w_1(u, v) = |I(u) - I(v)|$, $I(u)$ being the intensity of pixel u (cf. the so-called *watershed by dissimilarity* [LF00]); $w_2(u, v) = G(v)$, where $G(v)$ is the (morphological) gradient of image I at pixel v (cf. the *IFT-WT on gradient* [LF00, FSL04]). With this arc weight function, the max-arc path-cost function of Equation 4.1 can be simplified into: $f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) = \max \{ G(v_1), G(v_2), \dots, G(v_n) \}$. Note that the final cost map is unique and corresponds to the morphological superior reconstruction of the gradient image from the seeds using a flat structuring element. However, the forests and then the labelings may be multiple. Observe that a forest can be simply represented by a predecessor map P where $P(v)$ is the predecessor of node v in the minimum path. A label map L assigns to each node v the label $L(v)$ of the corresponding minimum-path root. The catchment basins correspond to the (labeled) trees: $CB_{IFT}(s_i) = \{v \in V, L(v) = L(s_i)\}$.

The so-called ‘‘plateau problem’’ is reported in WT literature for the internal non-minimum plateau

pixels, i.e. non-minimum¹ pixels which have no lower neighbor. It can be solved by *lower completion* (cf. Definition 3.4 of [RM00]): a *lower complete image*² I_{LC} is computed from I by taking into account the geodesic distance of such internal pixels to the lower boundary of the plateau; then WT is applied on I_{LC} .

In IFT-WT, a two-component *lexicographic cost function* $f_{lex} = (f_{max}, f_d)$ is used [LF00] to avoid a prior lower completion but has strictly the same role [AL07b]. The first component, of highest priority, is the max-arc function representing the flooding process. The second one corresponds to the geodesic distance to the lower boundary of the plateau and makes different waters propagate on plateau at a same speed rate:

$$f_d(\langle v_1, v_2, \dots, v_n \rangle) = \max_{k \in [0, n-1]} \{k, f_{max}(\langle v_1, \dots, v_n \rangle) = f_{max}(\langle v_1, \dots, v_{n-k} \rangle)\} \quad (4.2)$$

4.2.2 Tie Zone

The choice of a single IFT-WT solution when many are possible is arbitrary and can be seen as a bias. Indeed, variations from one solution to another are sometimes significant and even unacceptable for some applications (e.g. reliable measures on segmented structures). In some images, an entire region is reached passing by a bottleneck pixel [AL05] and consequently included to the basin that first invades the bottleneck (like in Figure 4.1(g)(h)). This problem is not related to the plateau problem and corresponds “to special pixel configurations which are not so rare in practice” as referred by [VS91].

It is why the *tie-zone concept* was proposed [ALC05, AL07b] to unify the multiple solutions of a WT. Briefly speaking, considering all possible solutions derived from a specific WT definition, parts segmented in the same manner remain as catchment basins whereas differing parts are put in the tie zone (TZ). So, the TZ may be thick as well as empty.

In the case of IFT-WT, the *tie-zone watershed by IFT* (TZ-IFT-WS), returns a unique partition (cf. Figure 4.1(j)) of the image such that: A node is included in catchment basin $CB_{TZ-IFT}(s_i)$ when it is linked by a path to a same seed s_i in all the optimal forests (Φ denotes the set of the optimal forests F), otherwise it is included in the tie zone TZ :

$$\begin{aligned} CB_{TZ-IFT}(s_i) &= \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\} \\ TZ_{IFT} &= V \setminus \bigcup_i CB_{TZ-IFT}(s_i) \end{aligned} \quad (4.3)$$

The area of the TZs, their distribution and number and distribution of their sources, the so-called

¹Pixels which do not belong to regional minima.

²The improper term “image without plateau” is sometimes used instead.

bottlenecks, can be correlated with the robustness of a segmentation, i.e. with the degree of confidence a particular segmentation by WT has [AL05].

4.2.3 Multipredecessor Optimal Graph and Lower Complete Graph

We introduce now a special graph, unique for each image, that will be used in Section 4.3. Roughly speaking, the *multipredecessor optimal graph* (MOG) of a weighted graph is the “union” of its optimal forests. More precisely, it is a directed acyclic subgraph of (V, A) such that its arc set A'' is the union of the (oriented) arcs of all the optimal forests $F \in \Phi$ (cf. Figure 4.1(j)):

$$MOG : (V, A'') = (V, \bigcup_{\forall F=(V,A') \in \Phi} A')$$

Once we have the lexicographical cost map of the image, i.e. a lower complete image, the following local property is valid: node p is predecessor of node v in the MOG if and only if p is neighbor of v with optimal lexicographic cost strictly lower than that of v (the superscript * denotes optimal paths).

$$\langle v, p \rangle \in A'' \Leftrightarrow p \in \mathbb{P}(v) \Leftrightarrow p \in N(v), f_{lex}(\pi_v^*) \succ f_{lex}(\pi_p^*) \quad (4.4)$$

where $\mathbb{P}(v)$ denotes the set of predecessors of node v , as the number of predecessors by node is no longer restricted to one as for the forests.

Another property of the MOG is that if we independently choose one predecessor by non-minimum node, we obtain an optimal forest ($A' \subseteq A'' \subseteq A$).

The *lower complete graph* (V, A''') (LCG, cf. Definition 3.5 of [RM00]) is analog to the MOG. Both are directed acyclic graphs built from the lower complete image. While all the *lower neighbors* in the lower complete image are predecessors of a node in the MOG, only the *steepest lower neighbors* are considered for a node in LCG (cf. Figure 4.1(i)).

$$\langle v, p \rangle \in A''' \Leftrightarrow p \in \mathbb{P}_{steepest}(v) \Leftrightarrow \begin{aligned} & p \in N(v), I_{LC}(v) > I_{LC}(p), \\ & \frac{I_{LC}(v) - I_{LC}(p)}{d(v,p)} = \max_{q \in N(v)} \frac{I_{LC}(v) - I_{LC}(q)}{d(v,q)} \end{aligned} \quad (4.5)$$

$d(p, q)$ being the distance between p and q . From Equation 4.4 and Equation 4.5, we deduce that $\mathbb{P}_{steepest}(v) \subseteq \mathbb{P}(v)$. Consequently, $A''' \subseteq A'' \subseteq A$ and the LCG (V, A''') of an image-graph (V, A) is a subgraph of its MOG (V, A'') .

4.3 Watershed based on a Local Condition

As we said in Section 4.1, the *watershed transform based on a local condition* (LC-WT) is of “region” type because it has no watershed pixels [BBM⁺97, BM98]. It may have multiple solutions (cf. Figure 4.1(e)(f)). It assigns to each pixel the label of some minimum m_i , so as to form a partition of the image whose disjoint sets are the basins $CB_{LC}(m_i) = \{v \in V, L(v) = L(m_i)\}$.

As observed in refs. [RM00, BBM⁺97], this WT definition is particularly well-suited for parallel implementations because it is based on a local condition. However, the overall WT computation is still a global operation. The meaning of locality in this definition is that one may subdivide an image in blocks, do a labeling of basins in each block independently, and make the results globally consistent in a final merging step.

Definition 1 (Watershed based on local condition) *For any lower complete image I_{LC} , a function L assigning a label to each pixel is called a watershed segmentation if:*

1. $L(m_i) \neq L(m_j) \quad \forall i \neq j$, with $\{m_k\}$ the set of minima of I_{LC} ;
2. for each pixel v with $\mathbb{P}_{steepest}(v) \neq \{\}$, $\exists p \in \mathbb{P}_{steepest}(v)$ with $L(v) = L(p)$.

the condition $\mathbb{P}_{steepest}(v) \neq \{\}$ meaning that v has at least one lower neighbor.

In other words, we can obtain a LC-WT by independently choosing one predecessor by non-minimum node in the precomputed LCG, and assigning a different basin label to each tree of the disjoint-set forest we obtained.

As the LCG (V, A''') generating such forests is a subgraph of the MOG (V, A'') generating any optimal forest, we conclude straightaway that these forests are optimal forests. Therefore: **any LC-WT is also an IFT-WT.**

4.4 Watershed based on Topographic Distance

We recall here the definition of *WT by topographic distance* (TD-WT) and some propositions from [Mey94b] for completeness.

Definition 2 (Watershed transform by topographic distance) *Let I be a gray-scale image, I_{LC} its lower completion, and $\{m_i\}$ the set of minima of I . Basin of I for minimum m_i and watershed are respectively:*

$$\begin{aligned}
 CB_{TD}(m_i) &= \{v \in V, \forall j \neq i, I_{LC}(m_i) + T_{I_{LC}}(v, m_i) < I_{LC}(m_j) + T_{I_{LC}}(v, m_j)\} \\
 W_{TD} &= V \setminus \bigcup_i CB_{TD}(m_i)
 \end{aligned} \tag{4.6}$$

$T_{I_{LC}}(p, q)$ being the *topographic distance* [Mey94b] between p and q :

$$T_{I_{LC}}(p, q) = \min_{\forall \pi(p, q)} T_{I_{LC}}^{\pi(p, q)}(p, q); \quad T_{I_{LC}}^{\pi(p, q)}(p = p_1, q = p_n) = \sum_{i=1}^{n-1} \text{cost}(p_i, p_{i+1})$$

$$\text{cost}(p_i, p_{i+1}) = \begin{cases} LS(p_i)d(p_i, p_{i+1}), & \text{if } I_{LC}(p_i) > I_{LC}(p_{i+1}) \\ LS(p_{i+1})d(p_i, p_{i+1}), & \text{if } I_{LC}(p_i) < I_{LC}(p_{i+1}) \\ \frac{1}{2} [LS(p_i) + LS(p_{i+1})] d(p_i, p_{i+1}), & \text{if } I_{LC}(p_i) = I_{LC}(p_{i+1}) \end{cases}$$

The *lower slope* $LS(p)$ of I_{LC} at a pixel p is defined as the maximal slope linking p to any of its neighbors of lower altitude.

We call (p_1, p_2, \dots, p_n) a *path of steepest descent* from $p_1 = p$ to $p_n = q$ if $p_{i+1} \in \mathbb{P}_{\text{steepest}}(p_i)$ for $i = 1, \dots, n - 1$. A pixel p belongs to the *upstream* of q if there exists a path of steepest descent from p to q .

Proposition 1 (from [Mey94b]) *Let $I_{LC}(p) > I_{LC}(q)$. A path π from p to q is of steepest descent if and only if $T_{I_{LC}}^{\pi}(p, q) = I_{LC}(p) - I_{LC}(q)$. If a path π from p to q is not of steepest descent, $T_{I_{LC}}^{\pi}(p, q) > I_{LC}(p) - I_{LC}(q)$.*

This proposition implies that paths of steepest descent are the geodesics (shortest paths) of the topographical distance function. Consequently, from Definition 2 $CB_{TD}(m_i)$ is the set of points in the upstream of a single minimum m_i , i.e., there is (at least) one path of steepest descent to m_i and no path of steepest descent to any other minimum. The watershed consists of the points p which are in the upstream of at least two minima, i.e., there are at least two paths of steepest descent starting from p which lead to different minima.

4.4.1 Relationship with Local-Condition Watershed

The forests representing the possible LC-WT generated from the LCG (Section 4.3) are made of paths of steepest descent. By strict analogy with Equation 4.3, we can conclude that: ***TD-WT is the tie-zone transform of LC-WT.***

Proof: A node is included in catchment basin $CB_{TD}(m_i)$ when it is linked by a path to a same minimum m_i in all the forests made of steepest paths (the set of solutions for LC-WT, e.g. Figure 4.1(e)(f)), otherwise it is included in the tie zone W_{TD} (cf. Figure 4.1(i)). As a consequence, we have also $CB_{TD}(m_i) \subseteq CB_{LC}(m_i)$ (cf. Figure 4.1(i)), as demonstrated in Theorem 2 of [BBM⁺97].

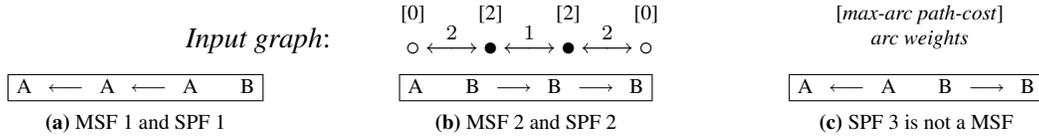


Fig. 4.2: A weighted graph with two markers (\circ) and its 3 possible SPF-max and 2 MSF (total weight = 3). SPF 3 is not a MSF (total weight = 4).

4.4.2 Relationship with Tie-Zone Watershed by IFT

We saw in Section 4.3 that the set of LC-WT solutions is a subset of the set of IFT-WT solutions, so *the tie zone derived from the LC-WT solutions, i.e. W_{TD} , is a subset of TZ_{IFT} : $W_{TD} \subseteq TZ_{IFT}$.*

Proof: If pixel $p \in W_{TD}$, there are at least two paths of steepest descent from p to different minima. These paths belong to the LCG and to the MOG too (LCG is subgraph of MOG). So, there exist at least two optimal forests containing these paths leading to different minima. Consequently, $p \in TZ_{IFT}$.

Besides, *the catchment basins defined by TZ-IFT-WT are subsets of the corresponding basins defined by TD-WT: $\forall m_i, CB_{TZ-IFT}(m_i) \subseteq CB_{TD}(m_i)$.*

Proof: If pixel $p \in CB_{TZ-IFT}(m_i)$, all the paths from p in the MOG lead to minimum m_i . So do the paths from p in the LCG (because LCG is subgraph of MOG, $\mathbb{P}_{steepest}(v) \subseteq \mathbb{P}(v), \forall v$). So, $p \in CB_{TD}(m_i)$.

4.5 Watershed based on a Minimum Spanning Forest

The WT introduced in [Mey94a] is in fact a *WT from markers* (some significant minima are selected to avoid oversegmentation). It uses a weighted *neighborhood graph* whose nodes are the primitive catchment basins corresponding to regional minima of the image. Arcs are placed between neighbor catchment basins and weighted by the altitude of the *pass* between them. A *watershed based on minimum spanning forest (MSF-WT)* is defined on this weighted graph: the many possible MSFs on the graph define partitions that are considered solutions of this WT. Each tree of the MSF is a catchment basin of the MSF-WT.

A tree (V, T) is a *minimum spanning tree (MST)* of graph (V, A, w) if its total weight $\sum_{t \in T} w(t)$ (sum of the weight of its arcs) is minimum. It is unique when all the arc weights of the graph are different. A *minimum spanning forest (MSF)* is a forest whose total weight (sum of the weight of its arcs) is minimum and where each node is linked to a seed $s_i \in S$ by a unique simple path. The MSF problem for weighted graph (V, A, w) can be solved by constructing the MST of (V^*, A^*, w^*) where a fictitious root node z and arcs of weight -1 linking z to each seed were added. In a final step, these

negative arcs will be removed to obtain a MSF.

Theorem 1 (Minimum spanning tree [GM84]) (V, T) is a tree of minimum weight for graph (V, A, w) if and only if for every arc $u \in A - T$ the cycle μ^u (such that $\mu^u \subset T + \{u\}$) satisfies: $w(u) \geq w(v)$, $\forall v \in \mu^u$ ($v \neq u$).

Now, we demonstrate that the set of MSF solutions is a subset of the set of IFT-WT solutions defined by the same weighted graph using the max-arc path cost³ and seed handicaps $h(s_i) = 0$.

Theorem 2 (Shortest-path forest and minimum spanning forest) Any minimum spanning forest (MSF) is also a shortest-path forest (SPF-max) using max-arc path cost f_{\max} . Reciprocal is false (cf. examples and counter-example in Figure 4.2).

$$F \text{ is a MSF} \Rightarrow F \text{ is a SPF-max (or IFT-WT)}$$

Proof: Suppose that F is a MSF and T the corresponding MST using a fictitious root z . Suppose that there exists a path π from p to z , π belongs to T and π is non optimal in the SPF-max sense (i.e. using f_{\max}). Suppose that there exists another path π' from p to z such that $f_{\max}(\pi') < f_{\max}(\pi)$. Then for every arc v in π' , its weight $w(v) \leq f_{\max}(\pi') < f_{\max}(\pi)$. Now, there exists an arc u in π' , u not in T (because T has no cycle: p and z are linked by only one simple path). Therefore, $w(u) < f_{\max}(\pi)$. Now, T is a MST. Therefore, from Theorem 1, $w(u) \geq f_{\max}(\pi \cdot \pi') \geq f_{\max}(\pi)$, $\pi \cdot \pi'$ being the cycle μ^u formed by concatenation of the two paths. That is a contradiction with the previous conclusion. So, any MSF is necessarily SPF-max.

4.6 Conclusion and Future Works

In this paper, we used the IFT-WT and the TZ concept (that unifies the set of multiple solutions of a given WT) to relate some discrete WT definitions and, thereby, better understand the differences between the multiple solutions given by such definitions. We demonstrate that (i) the TD-WT corresponds to the tie-zone transform of the LC-WT; (ii) the possibly thick and not separating watershed line of TD-WT is contained in the TZ of the TZ-IFT-WT (with lexicographic cost function), while (iii) the catchment basins of the former contain the basins of the latter; (iv) any solution of LC-WT is also solution of the IFT-WT; (v) any solution of MSF-WT is also solution of the IFT-WT (with max-arc path-cost function).

³Until now, lexicographic path-cost $f_{lex} = (f_{\max}, f_d)$ was used for IFT-WT.

We are preparing an extended version of this paper which will also include the comparative analysis of flooding-WT and topological-WT definitions with IFT-WT and TZ, as well as some issues on related algorithms.

Capítulo 5

Dualidade com a Conexidade Nebulosa

Este capítulo contém o artigo intitulado *Duality between the Watershed by Image Foresting Transform and the Fuzzy Connectedness Segmentation Approaches* [AL06]. Faz-se uma releitura das abordagens bem-sucedidas de segmentação por conexidade nebulosa e por *watershed* analisando-as por meio da IFT e da zona de empate. Percebe-se que elas resolvem problemas duais: enquanto a transformada de *watershed* por IFT agrupa pixels em um mesmo segmento quando a dissimilaridade entre eles é mínima, os métodos baseados em conexidade nebulosa agrupam pixels com similaridade máxima. Portanto, intuitivamente, é razoável que esse métodos sejam duais. Analisa-se minuciosamente as condições para tal dualidade.

A extração iterativa de objetos nebulosos relativos (*iterative relative fuzzy object extraction*, IR-FOE) corresponde exatamente à transformada em zona de empate do *watershed* via IFT (TZ-IFT-WT, sem custo lexicográfico): Ambas produzem a mesma segmentação (que corresponde a uma floresta de caminhos ótimos), uma vez que existe uma função estritamente decrescente que permite passar das ponderações do grafo de entrada de uma para o grafo de entrada da outra (ou seja, inverte a relação de ordem entre as ponderações dos arcos).

Um algoritmo eficiente é proposto para a TZ-IFT-WS e pode ser utilizado para a IRFOE. Ele é um pouco mais complexo que o algoritmo proposto para a TZ-IFT-WT-lex, pois não possui custo lexicográfico para lidar com os platôs da imagem. Logo, a técnica de *union-find* tem que ser utilizada.

Além disso, a extração de objetos nebulosos relativos (*relative fuzzy object extraction*, RFOE) pode ser obtida dualmente por comparação de reconstruções morfológicas superiores, e a extração de objetos nebulosos (*fuzzy object extraction*, FOE) por simples limiarização dessas reconstruções.

Na prática, as metodologias padrões das abordagens morfológica e nebulosa são um pouco diferentes. O *watershed* é geralmente calculado sobre um gradiente da imagem possivelmente filtrado. Desse gradiente são calculadas as ponderações dos arcos do grafo usado pela IFT. A abordagem nebulosa gera essas ponderações a partir de funções de afinidade nebulosa que podem ser, às vezes,

dependentes da escala estimada. Mas uma vez fixados os pesos dos arcos numa abordagem, uma simples inversão destes permite resolver o problema na abordagem dual. Portanto, nada impede integrar a modelagem dos pesos/afinidades de um método em outro.

Enfim, mostra-se que os pesos dos arcos geralmente assimétricos no contexto da IFT podem ser vistos como simétricos sob algumas condições. A TZ-IFT-WT pode assim aproveitar as propriedades de robustez da segmentação desenvolvidas na abordagem nebulosa. Percebe-se que essas duas abordagens desenvolveram ferramentas diferentes, e que os avanços de ambas poderão ser aproveitados melhor, já que a ponte entre elas é agora clara.

No Capítulo 6, demonstra-se que a TZ-IFT-WT é equivalente à TZ-MSF-WT, baseada nas florestas geradoras mínimas. Conseqüentemente, a IRFOE pode ser vista como a sobreposição de florestas geradoras mínimas.

Estamos escrevendo atualmente a extensão desse resultado da dualidade para o método de segmentação nebulosa de objetos múltiplos (*multiple object fuzzy segmentation*, MOFS). Este permite a definição de afinidades de arco múltiplas e assimétricas. No caso geral, não segue nenhum critério global de otimalidade, mas apenas uma otimização “temporária” em cada iteração do processo recursivo que define a MOFS. No entanto, quando se restringe o método no caso de afinidade única, são restaurados o critério global de otimização, assim como a dualidade, mas, desta vez, com a fusão de rótulos (*label merging*, LM), variante da TZ-IFT-WT. Com efeito, a MOFS permite a atribuição de vários rótulos por pixel.

Abstract

This paper makes a rereading of two successful image segmentation approaches, the fuzzy connectedness (FC) and the watershed (WS) approaches, by analyzing both by means of the Image Foresting Transform (IFT). This graph-based transform provides a sound framework for analyzing and implementing these methods. This paradigm allows to show the duality existing between the WS by IFT and the FC segmentation approaches. Both can be modeled by an optimal forest computation in a dual form (maximization of the similarities or minimization of the dissimilarities), the main difference being the input parameters: the weights associated to each arc of the graph representing the image. In the WS approach, such weights are based on the (possibly filtered) image gradient values whereas they are based on much more complex affinity values in the FC theory. An efficient algorithm for both FC and IFT-WS computation is proposed. Segmentation robustness issue is also discussed.

5.1 Introduction

This paper deals with two successful methods of semi-automatic segmentation. On the one hand, the segmentation by *fuzzy connectedness* (FC) comes from the fuzzy set theory and was proposed by Udupa and Samarasekera [US96] and has been improved [HC01, SU01, SUO00, USL02] and is still being developed [ZUS06]. It is based on the hangingness among elements within an object. So, this region-based technique detects “similarity” between pixels and return quite “homogeneous” regions. Each object is represented by a fuzzy set where a membership degree is associated to each element.

On the other hand, the segmentation by *watershed* (WS) comes from the mathematical morphology theory [DL03]. First proposed by Beucher and Lantuéjoul [BL79], the watershed transform has been successfully applied to image segmentation and successive variants and improvements of the original paradigm have been proposed [BM93, CB97a, LF00, LFZ02, Mey94b, MB90, NS94, RM00, VS91]. The WS approach detects frontier (the watershed lines) between regions of little dissimilarity (the catchment basins). It can be viewed as a computation of influence zones and achieved by successive morphological dilations. The most famous implementations [MB90, VS91] adopt a flooding strategy of the image viewed as a topographic surface.

The WS can be obtained by other paradigms like the Image Foresting Transform (IFT). First introduced by Lotufo and Falcão [LF00], this graph-based paradigm can model a WS transform by computing a forest of minimum-cost paths. It returns segmented regions, the trees, where the “dissimilarity” between each node was “minimized”.

The main contribution of this paper is to show the duality between the IFT-WS and the FC segmentation methods, and give an algorithm that solves both problems. The following concepts make the duality possible: (1) the interpretation of WS as a graph optimization problem in IFT framework [LF00] (strictly speaking, the other WS transforms cannot be seen like dual of FC methods); (2) the necessary existence of tie-zones [ALC05] in this optimization problem; and (3) the use of symmetric arc weights in the IFT-WS. This last condition is apparently not satisfied [LF00]. However, we show in this paper that the asymmetric arc weight generally used in IFT-WS can be seen as a symmetric one. Consequently, the robustness properties demonstrated in the FC framework [USL02] are valid for the IFT-WS approach too.

The paper is organized as follows. The IFT notations, definitions and algorithms are recalled in section 5.2. Section 5.3 presents the WS in the IFT framework, possible symmetric arc weights and a new algorithm for IFT-WS that takes into account the tie-zones. Section 5.4 presents the FC segmentation, shows the duality between the FC and the IFT-WS approaches and discusses the segmentation robustness issue. Note that due to space limitation, the duality is presented without all the necessary demonstrations. An extended paper with the required formalism is in preparation.

5.2 The Image Foresting Transform

The IFT is a general framework based on graph theory in which an image is interpreted as a graph and pixels as its nodes. To simplify, we call *pixel* or *spel* any spatial element: point, pixel or voxel. The key idea of this transform is to obtain, according to a path-cost function, a *shortest path forest* from an input image-graph. Depending on the path-cost function and some other input parameters (adjacency, arc weights), the IFT can compute different image processing operations [FSL04]: distance transforms, connected filters, interactive object delineation (“live-wire”) and both WS and FC segmentations as shown in sections 5.3 and 5.4.

5.2.1 Notation and definitions

Under the IFT framework, an *image* is interpreted as a weighted graph $G = (V, A, I)$ consisting of a set of *nodes* or *vertices* V that represent the image pixels, a set of arcs A and an *intensity map* I , that is, a function from V to some scalar or vectorial domain (e.g., \mathbb{Z} in the case of grayscale digital images, \mathbb{Z}^3 for RGB color images). The intensity of vertex v is denoted by $I(v)$.

An *adjacency relation* A is a binary irreflexive relation between vertices of V (e.g., 4- or 8-adjacency in 2D and 6- or 26-adjacency in 3D). The set of vertices adjacent to or neighbor of vertex u is denoted by $N(u)$. If vertices u and v are adjacent, then the *arc* $\langle u, v \rangle$ belongs to A .

A *path* $\pi(u, v)$ from a node u to a node v in a graph (V, A, I) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said *simple* if all its nodes are different from each other. A path with terminal node v is denoted by π_v . The path π_v is *trivial* when it consists of a single voxel $\langle v \rangle$. Otherwise, it can be defined by a path resulting from the concatenation $\pi_u \cdot \langle u, v \rangle$ of its longest prefix π_u with terminus u and the last arc $\langle u, v \rangle \in A$. A *path-cost function* f assigns to each path π a path cost $f(\pi)$, in some totally ordered set of cost values, whose maximum element is denoted by $+\infty$.

A graph $G' = (V', A')$ is *subgraph* of G if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$. A *forest* F of G is an acyclic subgraph F of G . *Trees* are connected components of the forest.

Let $S \subseteq V$ be a set of particular nodes s_i called *seeds*. For a given weighted graph $G = (V, A, I)$ and a set S of seeds, the *Image Foresting Transform* returns a forest F of G such that (i) there exists for each node $v \in V$ a unique and simple path $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path is *optimum*, i.e., has a minimum cost for linking v to some seed of S , according to the specified path-cost function f .

In other words, the IFT returns a shortest (cheapest in fact) path forest where each minimum-cost tree is rooted to a seed (but possibly not all seeds are roots of a tree). In this paper, the shortest-path forest is also referred to as an *optimum-path forest* or an *optimal forest*. In general, there may be

many paths of minimum cost leading to a given pixel and then, many optimal forests; only the path costs are uniquely defined.

Observe that, if we independently pick an optimum path for each pixel, the union of those paths may not be a forest. Moreover, as claimed in [FSL04], certain graphs and cost functions may not even admit any optimum-path forest. Sufficient condition for the existence of the IFT is to use a *smooth* path-cost function, i.e., a function f satisfying the three following conditions, given that for any pixel $v \in V$, there is an optimum path π_v ending at v which either is trivial, or has the form $\pi_u \cdot \langle u, v \rangle$: (i) $f(\pi_u) \leq f(\pi_v)$; (ii) π_u is optimum, and (iii) for any optimum path π'_u ending at u , $f(\pi'_u \cdot \langle u, v \rangle) = f(\pi_v)$.

Usually, the path cost depends on local properties of the image—such as color, gradient, and pixel position—along the path. Two popular examples of smooth functions are the *additive* path-cost function f_{sum} and the *max-arc* path-cost function f_{max} , defined by:

$$\begin{aligned} f_{sum}(\langle v \rangle) &= h(v) \\ f_{sum}(\pi_u \cdot \langle u, v \rangle) &= f_{sum}(\pi_u) + w(u, v) \\ f_{max}(\langle v \rangle) &= h(v) \\ f_{max}(\pi_u \cdot \langle u, v \rangle) &= \max\{f_{max}(\pi_u), w(u, v)\} \end{aligned} \tag{5.1}$$

where $\langle u, v \rangle \in A$, $h(v)$ is a fixed but arbitrary handicap cost for any paths starting at pixel v , and $w(u, v)$ is a fixed nonnegative weight assigned to the arc $\langle u, v \rangle$.

5.2.2 Algorithms

The efficient ordered queue-based algorithm for IFT proposed in [FSL04, LF00] is essentially Dijkstra's algorithm [Dij59], extended for multiple sources and a more general path-cost function. It computes three attributes for each vertex $v \in V$: its *predecessor* $P[v]$ in the optimum path, the *cost* $C[v]$ of that path from the tree root to v , and the corresponding *root label* $L[v]$.

Note that other algorithms are able to compute the IFT. For example, the ordered queue is not necessary. One can process the image data in raster-scan and anti raster-scan order alternatively until stability of the result (algorithm not presented here and based on Berge's one [Ber58]).

5.3 Watershed segmentation under the IFT framework

The watershed approach is often compared to a flooding simulation. If the pixel intensity represents the altitude, the image corresponds to a topographic surface. Holes are punched at some marked places. Then, the topography is inundated in water. Water springs from the holes (markers) and create catchment basins corresponding to objects. Watershed lines are dams built for separating the catchment basins growing from different sources (markers).

In order to define objects using the IFT, we assign a distinct label to each object and select at least one seed pixel per object (including background). After that, the IFT outputs an optimum-path forest where each object is represented by a set of trees rooted at seeds with the same label.

The path-cost function should be such that pixels of a given object are “more strongly” connected to its internal seeds than to any other. A suitable example is the function f_{\max} when the $w(u, v)$ is a *dissimilarity* function between u and v , usually computed based on properties of the input scene I . Ideally, function $w(u, v)$ must be higher on the object boundaries and lower inside the objects.

5.3.1 Watershed transforms by IFT

Many watershed transforms can be obtained by IFT according to the arc weight function. For example, the so-called *watershed by dissimilarity* [LF00] uses a symmetric arc weight function: $w_1(u, v) = |I(u) - I(v)|$.

Other example is the *IFT-watershed on gradient* [FSL04, LF00]. In this watershed transform, the arc weight that should be equal to some gradient intensity is: $w_2(u, v) = G(v)$, where $G(v)$ is the *morphological gradient* [DL03] of image I at pixel v . This algorithm simulates the flooding by the use of an ordered queue, and looks like the Beucher and Meyer’s ordered queue algorithm [BM93, MB90]. With this arc weight function, the max-arc path-cost function of Equation 5.1 can be simplified into:

$$f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) = \max \{h(v_1), G(v_2), \dots, G(v_n)\} \quad (5.2)$$

In this case, the IFT computes a region watershed transform where the trees of the forest, i.e., the disjoint sets of the forest, correspond to the catchment basins. Observe that all vertices are covered by this forest and no watershed line is returned, but only regions. Note also that the final cost map is unique and corresponds to the *morphological superior reconstruction* [DL03] of the gradient image from the seeds using a flat structuring element. However, the forest and then the labeling may be multiple. Indeed, the IFT can result in many optimal forests because many paths of minimum cost are sometimes possible.

It may be desirable to deal with symmetric arc weight, as in the watershed by dissimilarity. Instead of using the previous asymmetric arc weight, we can use the symmetric weight $w_3(u, v) =$

$\max \{G(u), G(v)\}$. In this case, the path-cost function of Equation 5.1 becomes:

$$\begin{aligned} f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) &= \max \{h(v_1), \max \{G(v_1), G(v_2)\}, \\ &\quad \dots, \max \{G(v_{n-1}), G(v_n)\}\} \\ &= \max \{h(v_1), G(v_1), \dots, G(v_n)\} \end{aligned} \quad (5.3)$$

To get equivalent results for watersheds on gradient with asymmetric and symmetric arc weights (Equations 5.2 and 5.3), we can define the handicap function as: $h(s) = G(s)$ for any seed s and $h(v) = \infty$ elsewhere. The role of the handicap function can be found in [LFZ02].

5.3.2 Tie-zone watershed

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image-graph. The *tie-zone watershed* (TZWS), introduced by Audigier, Lotufo and Couprie [ALC05], returns a unique partition by definition. Indeed, this novel watershed paradigm avoids the introduction of bias due to a specific implementation [AL05]. We recall the definition for completeness of the paper.

A node is included in a specific catchment basin CB_i when it is linked by a path to a same seed s_i in all the optimal forests, otherwise it is included in the tie-zone T :

$$CB_i = \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\} \quad (5.4)$$

$$T = V \setminus \bigcup_i CB_i \quad (5.5)$$

where the set of all optimal forests is denoted by Φ .

Considering all the optimal forests, the nodes of the tie-zone can be linked to different seeds by paths of equal minimum cost. The TZWS labeling solution is defined without ambiguity because it synthesizes all the possible labelings due to optimal forests.

The algorithm presented in [ALC05] and based on Dijkstra's computes the TZWS in the case of a *lexicographic path-cost function*. This function has two components of decreasing priority: the max-arc path-cost function and the distance to the flat zone border. This lexicographic function has the effect of reducing part of the tie-zone that occurs on flat-zones. In the case of the max-arc path-cost function without lexicographic component, the cited algorithm cannot manage the flat zones correctly and cannot return the TZWS.

We propose here two algorithms for computing the TZWS in the case of the max-arc path-cost function without lexicographic second component. First, it is possible to obtain the TZWS from

any optimal forest by means of a sequential algorithm. Scan all the pixels and for each pixel, try to propagate each neighbor's cost to it. If the proposed cost is equal to the previous one and if the pixels have different labels, then assign the pixel to the tie-zone. Repeat this sequential scanning until stabilization of the labeling.

The second algorithm (cf. Algorithm 3) is based on Dijkstra's and utilizes a *union-find* strategy to deal with the flat zones. Indeed, the Dijkstra-IFT algorithm without union-find would not manage the following case: when two pixels of different labels meet together in a flat zone of the cost map, the entire flat zone should belong to the tie-zone. But the ordered queue used does not handle already processed pixels. It is why the union-find is added to the Dijkstra-IFT algorithm: each pixel needs to point at the root of the flat zone it belongs to. When pixels meet in a flat zone (lines 15-19) and form a tie-zone, it is sufficient to merge ("union", line 19) their respective flat zone roots ("find") and label the resulting root as a tie-zone. When the ordered queue is empty (each pixel has been processed only once), a scanning of the entire image is necessary (line 22) to get the final labeling.

Algorithm 3: TZWS algorithm by union-find

Inputs: image (V, A, I) , neighborhood N (derived from A),
seeds S , handicap h (infinity for non-seed nodes),
labeling λ and arc weight w functions.

Outputs: cost C , label L , predecessor P maps.

Auxiliary Data: empty ordered queue Q , state flag $done$,
cost variable c , flat zone root map R .

```

1.  $\forall p \in V, C[p] \leftarrow h(p); R[p] \leftarrow p; done(p) \leftarrow \text{FALSE};$ 
2.  $\forall p \notin S, L[p] \leftarrow \text{NIL}; P[p] \leftarrow \text{NIL};$ 
3.  $\forall s \in S, L[s] \leftarrow \lambda(s); P[s] \leftarrow s; \text{Insert}(s, Q, C[s]);$ 

4. while  $\text{IsEmpty}(Q) = \text{FALSE},$ 
5.    $v \leftarrow \text{RemoveMin}(Q); done(v) \leftarrow \text{TRUE};$ 
6.   if  $L[\text{Find}(P[v])] = \text{TZ}, L[v] \leftarrow \text{TZ};$ 
7.    $\forall p \in N(v),$  and  $done(p) = \text{FALSE},$ 
8.      $c \leftarrow \max\{C[v], w(v, p)\};$ 
9.     if  $c < C[p],$ 
10.       $C[p] \leftarrow c; L[p] \leftarrow L[v]; P[p] \leftarrow v;$ 
11.      if  $p$  in  $Q, \text{Remove}(p, Q);$ 
12.       $\text{Insert}(p, Q, c);$ 
13.      if  $C[p] = C[v], R[p] \leftarrow \text{Find}(v);$ 
14.    else, if  $c = C[p],$ 
15.      if  $C[p] = C[v],$  /* flat zone */
16.        if  $\text{Find}(p) \neq \text{Find}(v),$ 
17.          if  $L[\text{Find}(p)] \neq L[\text{Find}(v)],$ 
18.             $L[\text{Find}(v)] \leftarrow \text{TZ};$ 
19.             $R[\text{Find}(p)] \leftarrow \text{Find}(v);$  /* union */
20.          else, if  $L[\text{Find}(p)] \neq L[\text{Find}(v)],$ 
21.             $L[\text{Find}(p)] \leftarrow \text{TZ}; L[p] \leftarrow \text{TZ};$ 

22.  $\forall p \in V, L[p] \leftarrow L[\text{Find}(p)];$ 

```

$\text{Find}(v)$: /* with path compression */

```

while  $R[R[v]] \neq R[v], R[v] \leftarrow \text{Find}(R[v]);$ 
return  $R[v];$ 

```

5.4 Fuzzy Connectedness Segmentation under the IFT Framework

First, we report the key ideas of the fuzzy connectedness (FC) approach [US96] and relate to the IFT formalism (section 5.4.1). Different FC-based segmentation methods are presented (section 5.4.2). The segmentation robustness is discussed in section 5.4.3. Finally, we illustrate the concepts presented (section 5.4.4).

The segmentation by FC aims to create fuzzy sets from an image to represent the objects. In this case, a fuzzy set \mathcal{O} is a set O of spatial elements (called *spels*; i.e. points, pixels or voxels depending on the data dimension) assigned by the degree of belongingness (membership) $\mu_{\mathcal{O}}$ to an object. This degree is a real number between 0 and 1. When it is 0, it means that the spel does definitely not belong to the object. When it comes close to 1, it certainly belongs to the object. In case of multiple object segmentation, a degree of belongingness to each object can be assigned to each spel. And, the final object extraction can be obtained by several ways: by applying a minimum threshold to this degree or by simply assigning the pixel to the object with highest degree.

The idea of the FC is that the objects correspond to sets of close spels (in intensity and position), i.e. spels that are “more strongly connected” or have a “greater fuzzy connectedness”. In general, seeds representative of each object are given as an input of the method that returns sets of spels hanging together with the seeds.

5.4.1 Key ideas on fuzzy connectedness

In this section, we sum up several basic concepts of FC approach that are fully developed in [USL02] and reinterpret them from the IFT point of view.

A *fuzzy digital space* is defined as (Z^n, α) where Z^n is a set of spels in an n -dimensional space (but it could be any set of elements) and α is a *fuzzy spel adjacency*. The adjacency is a reflexive and symmetric relation. It associates to each pair of spels a value μ_{α} bounded by 0 and 1 and generally given by a nonincreasing function of their Euclidean distance. Therefore, a spel is adjacent to itself and has an adjacency value of 1 (reflexivity). And any two spels whose fuzzy adjacency value is not zero are said *adjacent*. Note that the adjective “fuzzy” is used when a real value bounded by 0 and 1 is assigned to a spel of a set or to a pair of spels.

Like IFT, FC also uses the concept of vertices (spels) and arcs if we consider that –by the symmetry of the adjacency– there exists an undirected arc between any two adjacent (or neighbor) spels. Note that the arcs are weighted by the adjacency value that only depends on spatial configuration between spels. But this positive weight does not matter –once we know that it is positive– if we are

only interested in scanning all neighbors of a spel.

FC methods are based on two fuzzy relations:

- A local fuzzy relation κ called *affinity* on the set of spels. The affinity value μ_κ assigned to a pair of spels is based on the nearness of spels in space and in intensity (or in features derived from intensities). Affinity represents local “hanging togetherness” of spels.
- A global fuzzy relation K called *connectedness*, on the set of spels, based on affinity κ . The connectedness value μ_K assigned to a pair of spels is the strength of the strongest path linking these spels. The strength of a path is the lowest affinity along the path.

The affinity is a reflexive and symmetric fuzzy relation whose valuing function may depend on: (i) the fuzzy adjacency (spatial distance) α between the spels; (ii) the homogeneity ψ , whose value is bounded by 0 and 1 and is greater when both spels belong to the same homogeneous region; (iii) intensity-based features ϕ , (when the intensity feature is closer to an expected value of the features for a given object); (iv) the actual location of the spels, a shift-variant value not used in practice. A typical functional form for the affinity value μ_κ is: $\mu_\kappa = \mu_\alpha \sqrt{\mu_\psi \mu_\phi}$ where μ_α is 1 for the hard 4- or 6-adjacency relation. Note that μ_ψ and μ_ϕ are evaluated in the neighborhood of each pixel and that neighborhoods may depend on space-variant local scales. Details of the design of such affinity functions to correctly represent the nearness or similarity between spels are given in [SUO00].

The affinity value between two spels can be viewed as the weight of a virtually added arc shared by these spels. Imagine now that, instead of valuing the similarity between spels, we value their dissimilarity (with the complement or inverse of the affinity value for example). This dual arc weight can be seen as a cost or penalty to pixels that are not sufficiently near. It is exactly the case of the arc weight used in IFT. For the special case of WS by IFT, this weight is given by some gradient function (i.e., heterogeneity measure) of the image to segment. And for non-adjacent vertices, no arc is considered or, virtually, the arc has an infinite weight.

In FC methods, the *strength of a path* is defined as the minimum affinity value of any pair of spels in the path (see the so-called “fuzzy κ -net” from [USL02] that assigns this strength to each possible path). In IFT framework, a *path-cost* is assigned to each path. In the case of WS by IFT, this cost corresponds to the maximum arc, which is exactly the dual of the path strength.

Associated to a given affinity κ , the fuzzy connectedness K assigns to a pair of spels the value μ_K defined as the maximum path strength when all paths linking these two spels are considered. Dually, in the IFT case, paths of minimum cost are computed.

The duality between FC and IFT-WS concepts is summarized in the upper part of Table 5.1. Correspondences between main notations used in IFT-WS and in [USL02] are given. The lower part shows the dual segmentation methods presented in the next section. Note that all the FC (and IFT-WS) methods have variants if other affinities (arc weights) are used (e.g., a scale-based affinity, other

Tab. 5.1: Duality in concepts and segmentation methods.

Fuzzy connectedness (FC)	Watershed (WS) by IFT
fuzzy spel affinity $\mu_\kappa(c, d)$	arc weight $w(c, d)$
path p_{cd} : $\langle c = c_1, c_2, \dots, c_L = d \rangle$	path $\pi(c, d)$: $\langle c = c_1, c_2, \dots, c_L = d \rangle$
path strength: $\min_{1 < l \leq L} \mu_\kappa(c_{l-1}, c_l)$	(max-arc) path cost: $\max_{1 < l \leq L} w(c_{l-1}, c_l)$
strength of FC $\mu_K(c, d)$: $\max_{\forall p_{cd}} \{\text{path strength}\}$	optimal path cost $C[d]$: $\min_{\forall \pi(c, d)} \{\text{path cost}\}$
FOE: threshold θ_{\min}	reconstruction (IFT cost map): threshold θ_{\max}
RFOE (strict max FC)	independent reconstructions from each seed (strict min cost) $\implies \text{CORES}$
IRFOE (iterative blocking)	TZWS: one synchronous IFT (seeds compete together) $\implies \text{CATCHMENT BASINS}$

gradient).

5.4.2 Segmentation by fuzzy connectedness

Fuzzy object extraction by connectedness thresholding

Now FC has been defined, many fuzzy object extractions are possible. Initially, Udupa and Samasekera [US96] proposed a fuzzy object extraction, FOE, based on a threshold θ . For a given object seed o , after computing for each spel the FC with the seed, set a minimum threshold on the strength of connectedness to obtain the set of object spels. The *fuzzy $\kappa\theta$ -object* \mathcal{O} is defined by:

$$\mathcal{O} = (O_{\theta_\kappa}, \mu_{\mathcal{O}})$$

$$O_{\theta_\kappa} = \{v \in V, \mu_K(o, v) \geq \theta\} \quad (5.6)$$

$$\mu_{\mathcal{O}}(v) = \begin{cases} \eta(I(v)), & \text{if } v \in O_{\theta_\kappa}, \\ 0, & \text{otherwise} \end{cases} \quad (5.7)$$

Note that the extracted object spels have an associated value given by a function η of their intensity while the remaining spels have value 0. In the case of a *hard* segmentation (by *fuzzy* connectedness), the function η is a constant (1 for binary segmentation or the object label λ for N-ary segmentation, e.g.). Observe also that the strength of connectedness between any two object spels is greater or equal than the threshold value; and the strength of connectedness between an object spel and any other spel out of the object is less than the threshold.

In IFT context, FOE is equivalent to (i) computing the IFT from a unique object seed on the “dual graph” (composed by the same vertices and arcs but with “dual” weight—complement to 1, e.g.), then (ii) setting the “dual” maximum θ_{\max} threshold on the cost map (i.e., the superior reconstruction of the image) and finally (iii) applying the function η to the selected spels if a fuzzy object is expected.

Relative fuzzy object extraction

Udupa, Saha and Lotufo [USL02] proposed then another object extraction method: the relative fuzzy object extraction (RFOE). In this method, strength of connectedness to the object and also to all co-objects are considered. Indeed, the objects are let to compete among themselves in having spels as their members. The spels will belong to the object that has the highest strength. Therefore, the object definition depends on how the spels hang together among themselves relative to others. This method does not need any threshold.

The *relative fuzzy κ -object* \mathcal{O} is defined as the set of spels more strongly connected to the object seed o than to the background (or co-object) seed b :

$$\begin{aligned} \mathcal{O} &= (P_{ob_\kappa}, \mu_{\mathcal{O}}) \\ P_{ob_\kappa} &= \{v \in V, \mu_K(o, v) > \mu_K(b, v)\} \\ \mu_{\mathcal{O}}(v) &= \begin{cases} \eta(I(v)), & \text{if } v \in P_{ob_\kappa}, \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5.8)$$

In the IFT framework, the RFOE is dually equivalent to independently compute IFT-WS from each seed $t \in S$ (one-by-one) and finally assign to each pixel v the label $\lambda(s)$ corresponding to the seed s that links it with a *strict* minimum path-cost $C_s[v]$, considering all cost maps C_t (superior reconstructions). Each labeled region we obtain is called *core* K_s (just as in [USL02]) of the seed s :

$$K_s = \{v \in V, C_s[v] < C_t[v], \forall t \in S, t \neq s\} \quad (5.9)$$

where C_s corresponds to the superior reconstruction of the image from seed s . Note that there can be pixels with no label. They do not belong to any object because there are many paths of same (non-strict) minimum cost linking to different seeds.

So, instead of using the RFOE algorithm of [USL02], we can follow this procedure: (i) For the extraction of n objects (background included), compute n superior reconstructions of the “dual” image-graph from seeds of different objects. (ii) Then compute the strict minimum of the reconstructions at each pixel. The cores are obtained.

Iterative relative fuzzy object extraction

We saw that some spels can have same connectedness with two different objects and therefore do not belong to any object. Sometimes the path, where the connectedness to an object was computed, passes through the other object (core). For the RFOE method, these two paths tie together. In fact, one of these paths invades the other object for trying to link a disconnected region to the core it comes from. The idea of the iterative relative fuzzy object extraction (IRFOE) proposed in [USL02] is to avoid this “passing through” by *blocking* any paths that try to pass through an object but come from other objects. This is exactly the same concept implicitly used in IFT. Indeed, the blocking is inherent to the forest of optimal paths: when a node is assigned to a particular tree, it cannot be used by any other growing tree.

The effect of the IRFOE is to expand iteratively the initial objects, the cores, obtained by RFOE, by reducing the chances of ties thanks to the blocking strategy. Nevertheless, not all ties are untied because some regions are linked to many seeds with same connectedness but without any path overlapping. Under the IFT framework, they correspond to the tie-zone defined in Equation 5.5 while the objects defined by IRFOE correspond to the CBs of Equation 5.4.

In practice, the IRFOE algorithm extracts first the cores and at the second iteration, any path from other seeds passing through a core is penalized by a low connectedness value (zero). In fact, affinity between any spel of an object obtained at iteration i is automatically set to zero at the next iteration j . This constitutes a barrier to protect the objects of iteration i . In other words, pixels that have been conquered in previous iteration cannot be used by paths from other object seeds in the subsequent iterations.

It is shown that the objects iteratively defined are noncontracting: object of iteration $i < j$ is included in object of iteration j . Moreover, the objects maintain their disjointness at every iteration. Observe that these properties are valid in the TZWS-IFT case: the cores are always included in the catchment basins and these are disjoint.

Thus, alternatively to the IRFOE presented in [USL02], the algorithm of TZWS by union-find proposed in section 5.3.2 finds the optimal paths without overlapping in an ordered way and returns directly the catchment basins without needing iterative object extraction.

Variants of the fuzzy connectedness approach

First, note that the RFOE of [USL02] extracts only one object relative to a background (the co-objects). It must theoretically be applied n times if n objects have to be segmented. The generalization of this method was proposed in [SU01]: the multiple relative fuzzy object extraction (MRFOE) processes many objects together. Let S be the set of object seeds and o a particular seed. The set of

seeds different from o is denoted by $b(o) = S \setminus \{o\}$. Therefore, the definition of relative fuzzy object in Equation 5.8 becomes:

$$P_{ob(o)_\kappa} = \{v \in V, \mu_K(o, v) > \mu_K(o', v), \forall o' \in b(o)\}$$

This does not change anything in the correspondence with multiple reconstructions where strict minima among them determine which object is better linked to a spel.

In addition, the MRFOE method [SU01] allows individual affinity for each object. Indeed, one drawback of the RFOE method is that the same affinity relation must be used for different objects. “This restriction somewhat compromises the effectiveness of the segmentation that can be achieved” [SU01]. On the other hand, if different affinities are used for different objects, then most of the properties we report in the next section will not hold. It is why the MRFOE method combines the multiple object affinities in a single affinity so that the theoretical results are valid as well as more effective practical segmentation is achieved.

Saha and Udupa [SU00] proposed scale-based affinity functions to improve the segmentation results of the fuzzy connectedness methods. Zhuge, Udupa and Saha [ZUS06] have recently introduced the vectorial scale-based relative fuzzy object extraction (VSRFOE) where the method of MRFOE with scale-based affinities is generalized from scalar images to vectorial images.

Finally, we can cite the approach of Herman and Carvalho [HC01] that proposes a multiseeded segmentation by FC. This method is somewhat different from those reported before because multiple affinities are indeed allowed and spels may belong to many objects with same maximum FC.

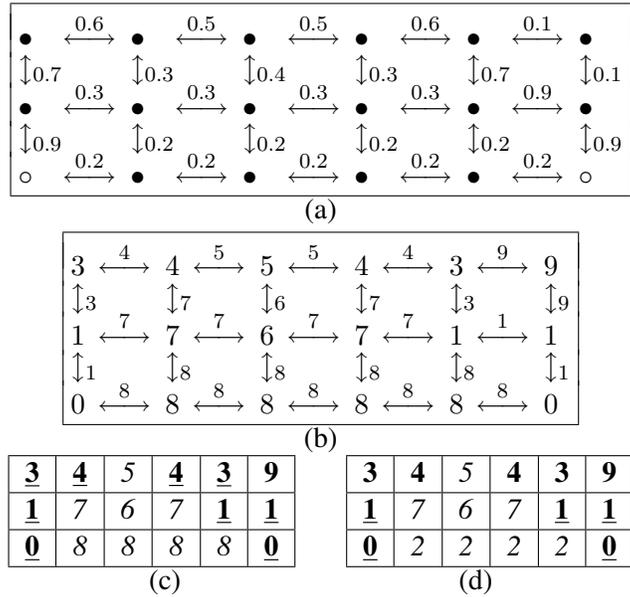
5.4.3 Robustness of the segmented objects

A series of useful properties and theorems are demonstrated in [USL02] and are valid for symmetric affinities (like in RFOE) or arc weights (like w_1 or w_3 of section 5.3.1). We recall only two of them and translate them in IFT-WS terms.

Any pair of spels in the defined object is linked by a best path entirely contained in the object. This property (Prop.3.2 of [USL02]) is also valid in the case of IFT-WS for both cores and catchment basins because the trees that represent them are composed of optimal paths by definition.

More interestingly, Theorem 3.4 of [USL02] guarantees the *robustness* of the objects (iteratively defined by the IRFOE) with respect to their respective seeds. Indeed, if the user designates other object seeds but still in their respective cores, the same objects will be obtained at any iteration of the IRFOE. Observe that the seeds must belong to the cores to guarantee the robustness of the objects. In IFT-WS framework, the robustness of the cores and the catchment basins is also guaranteed as long as the seeds are picked in the respective cores and, obviously, the max-arc path-cost function and a

symmetric arc weight function are used. Note that the lexicographic cost (used in [ALC05, LF00]) could not be used because it would untie flat zones (of the tie-zone) differently if seeds were chosen in different places, and so, robustness would not be valid anymore.



cores - **catchment basins** - *tie-zone*

Fig. 5.1: Images and object extractions.

5.4.4 Illustration

Figure 5.1(a) presents an example of image with fuzzy affinities between the spels “•”. Two seeds “○” were picked. Figure 5.1(b) shows a gradient image with the respective symmetric arc weights w_3 (section 5.3.1). We could also see the weights of the arcs of this image-graph as the “dual” weights of those from image (a). In this example, we applied $w(c, d) = \lceil k(1 - \mu_\kappa(c, d)) \rceil$ with $k = 10$ to compute the complement of the affinity and get an integer dissimilarity measure. Figure 5.1(c) is the output of the TZWS algorithm. The tie-zone is in italic whereas the two catchment basins (CB) are boldfaced. They correspond to two seeds placed in “0”. But two any other seeds chosen in the respective cores (underlined) would lead to the same cores and CBs (robustness). Disregarding the numbers, (c) corresponds also to the segmentation of (a) (duality). Observe the “blocking” applied by the right-hand core on the paths to the spel “9” coming from the left-hand seed and having an equal cost of 9. Figure 5.1(d) presents the segmentation of another image, which differs from (b) only in the flat zone of “8”. The cores are reduced to 2 and 3 spels while the CBs are the same. Now, the blocking is also applied for paths passing by the flat zone “2” and going to “3” and “4”.

5.5 Conclusion

In this paper, we talked about two successful segmentation methods. We showed the duality between the watershed by image foresting transform (IFT-WS) and the fuzzy connectedness (FC) segmentation approaches using the IFT framework. The IFT, an optimal forest computation, is a sound framework for understanding both. As the IFT-WS method finds the paths that link spels of an object by minimizing the dissimilarity among them, the FC method finds the paths that link spels of an object by maximizing the similarity (connectedness) among them. The methods solve dual problems.

Only parameters of the problems really differ because the methods were historically developed in different contexts. For example, FC works on fuzzy sets (fuzzy adjacency and fuzzy segmented object) that are in general not used in IFT-WS but could be. The outstanding difference of parameters is the design of the arc weight functions (IFT-WS) and their analog affinity functions (FC). In WS approach, the image is generally filtered and a gradient or dissimilarity operator is applied. In the FC approach, no preprocessing is applied but, instead, complex affinity functions determine the arc affinity. Examples of the sophistication of such affinity functions are given in [SU00] where the affinity can be scale-based. Observe that these sophistications could be modeled and integrated in the WS approach as preprocessing steps for computing other arc weights than the usual.

In addition, we saw that the duality is established when the WS is used in its IFT formulation with tie-zone (TZ). The tie-zone, that includes spels that have same connectedness/cost to at least two objects, must be taken into account to guarantee a unique segmentation. So, the new efficient algorithm of TZWS with union-find is another contribution of this paper: it can compute both FC and WS segmentation methods.

As a last contribution, we showed that the usual asymmetric arc weight for IFT-WS can be viewed as a symmetric one, so that all properties demonstrated for FC –in particular, the segmentation robustness– hold for IFT-WS too. In future, we think that both research domains –FC and WS– can benefit from the other one’s advance using the duality.

Capítulo 6

Robustez em Relação às Sementes e Conjuntos Mínimos de Sementes

Este capítulo contém o artigo intitulado *Seed-Relative Segmentation Robustness of Watershed and Fuzzy Connectedness Approaches* [AL07a]. Analisa-se a questão da robustez de três métodos de segmentação em relação às sementes escolhidas em entrada: a extração iterativa de objetos nebulosos relativos (IRFOE), a transformada em zona de empate do *watershed* via IFT (TZ-IFT-WT) e a transformada em zona de empate do *watershed* via floresta geradora mínima (TZ-MSF-WT). Procura-se saber o quanto uma modificação das sementes escolhidas influencia esses métodos de segmentação. Sabe-se que a escolha manual sempre é sujeita a variações intra- e inter-operadores. Por outro lado, a escolha automática é muitas vezes o resultado de um processo dependente de parâmetros a serem ajustados.

Retomando o conceito de objeto robusto introduzido no quadro da conexidade nebulosa [USL02], define-se o núcleo (*core*) de uma semente como a região dentro da qual a semente pode ser deslocada sem alterar a sua zona de influência (a extensão da árvore, por exemplo), logo, a segmentação.

Demonstra-se que a TZ-IFT-WT é igual à TZ-MSF-WT ao contrário do que se poderia pensar intuitivamente, já que toda MSF-WT é IFT-WT mas o contrário não. Em seguida, conclui-se que os núcleos são idênticos nos dois métodos e na IRFOE. Nota-se que, ao contrário do que se pensava, os núcleos são incluídos nas bacias de retenção mas diferentes delas, como mostra a Figura 6.1: Um grafo ponderado e suas duas sementes (assinaladas por \square) definem dois núcleos (em cinza claro) contidos em bacias de retenção (em cinza escuro), assim como uma zona de empate (em branco). Os custos dos caminhos mínimos (representados por traços mais espessos) entre um nó e cada uma das sementes são indicados entre parênteses acima do nó. Observa-se que os núcleos podem ser determinados apenas pela comparação desses custos, ao contrário da zona de empate e das bacias. Quando as sementes são deslocadas dentro dos seus núcleos respectivos, a partição permanece idêntica Figu-

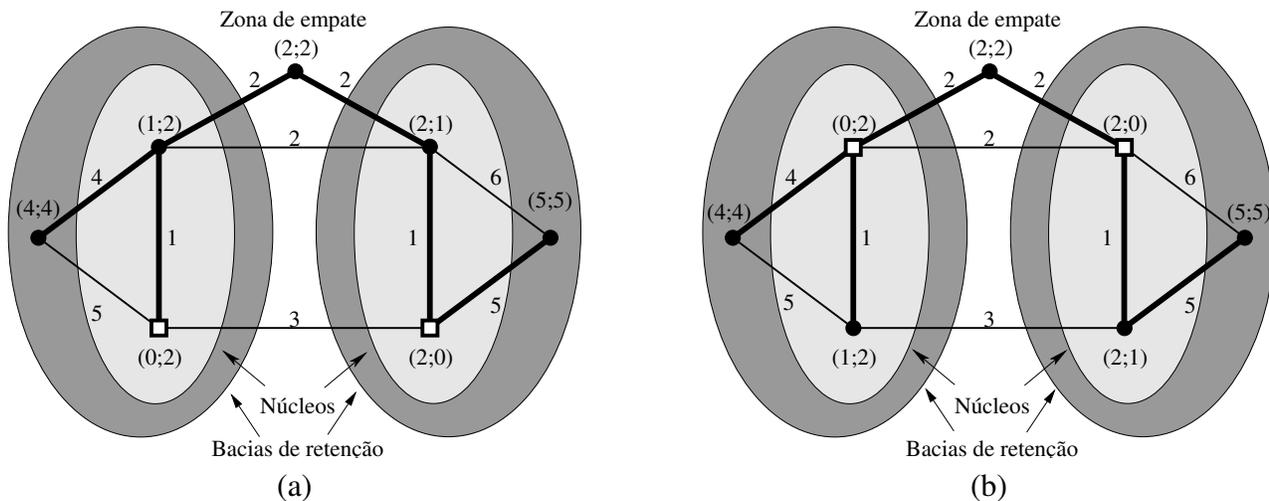


Fig. 6.1: Núcleos, bacias e zona de empate de um grafo ponderado e suas sementes: As sementes não alteram a segmentação quando ficam nos seus núcleos.

ras 6.1(a) e (b).

A determinação dos núcleos oferece informações valiosas. Núcleos com área grande podem avisar o usuário que as sementes manualmente escolhidas irão gerar uma segmentação muito estável, mesmo que a escolha de sementes sofra variações intra- e inter-usuários. Núcleos com pouca extensão significam que um leve deslocamento das sementes irão modificar a segmentação. Além disso, as características dos núcleos (área, distribuição) podem servir de realimentação num processo de detecção automática de sementes (o efeito de uma pré-filtragem, por exemplo).

Às vezes, várias sementes têm o mesmo rótulo e a extensão conjunta das suas árvores define o segmento. Neste problema de robustez, não importa se uma das árvores compondo o segmento tem sua extensão modificada, desde que a extensão global das árvores de mesmo rótulo seja constante. Esse problema, inverso da segmentação, é resolvido pelos conjuntos mínimos de sementes [LS02]: Pelo menos uma semente tem que ser escolhida em cada “região receptiva não redundante” (NRRR) com o devido rótulo. Sementes suplementares podem ser inseridas em outros lugares (“regiões receptivas redundantes”, “zona morta”) com rótulos adequados, pois não afetam a segmentação. Prova-se que as regiões receptivas não redundantes são equivalentes aos núcleos das sementes de qualquer conjunto mínimo de sementes. Portanto, os conjuntos mínimos de sementes garantem não somente a constância da segmentação (rotulação), mas ainda a constância das extensões das árvores que compõem cada segmento. A adição de sementes suplementares apenas afeta a extensão individual de árvores, mas não a extensão do segmento (rótulo).

O algoritmo proposto em [Sil01], baseada nos algoritmos de árvore geradora mínima, calcula regiões receptivas não redundantes apenas no caso de segmentações que sejam agrupamentos de “bacias primitivas”, isto é, as bacias obtidas por supersegmentação do *watershed* clássico (corresponde

às bacias dos mínimos regionais da imagem). Propõe-se um algoritmo para o cálculo das regiões receptivas não redundantes para qualquer segmentação de entrada. Extendemos, portanto, o problema do nível de bacia primitiva para o nível de pixel.

A visualização das regiões receptivas não redundantes, redundantes e zonas mortas poderia ajudar o usuário em rotinas de segmentação interativa. Por exemplo, ele poderia selecionar apenas as regiões satisfatórias de uma primeira segmentação, calcular os conjuntos mínimos de sementes, os quais o guiaria para refinar a segmentação.

Abstract

This paper analyzes the robustness issue in three segmentation approaches: the iterative relative fuzzy object extraction, the watershed transforms (WT) by image foresting transform and by minimum spanning forest. These methods need input seeds, which can be source of variability in the segmentation result. So, the robustness of these segmentation methods in relation to the input seeds is focused. The core of each seed is defined as the region where the seed can be moved without altering the segmentation result. We demonstrate that the core is identical for the three methods providing that the tie-zone transform has previously been applied on these methods. Indeed, as the two WT approaches do not return unique solution, the set of possible solutions has to be considered in a unified solution. So does the tie-zone transform. As opposed to what we could think, we show that the core is included in but different from the catchment basin. We also demonstrate that the tie-zone transforms of these WTs are always identical. Furthermore, the framework of minimal sets of seeds, an inverse problem of segmentation, is extended to the pixel level and related to the cores. A new algorithm for the computation of minimal seed sets is finally proposed.

IRFOE	iterative relative fuzzy object extraction
WT	watershed transform
IFT	image foresting transform
SPF	shortest path forest
SPF-max	shortest path forest using the max-arc path cost
MSF	minimum spanning forest
TZ	tie zone
MSS	minimal seed set
NRRR	non-redundant receptive region

Tab. 6.1: Table of abbreviations

6.1 Introduction

Watershed transform [BM93] and fuzzy connectedness [US96] are successful approaches to segment an image. Both need seeds on the input to define the segments of interest. Seeds can be either picked by the user or automatically selected. The choice of seeds is however a source of variability. Indeed, manual choice can vary depending on the user and the moment. Automatic selection of seeds also depends of filtering or other processing which generally depends on parameters. It is why we study in this paper the theoretical problem of segmentation robustness in relation to the input seeds: the characterization of the cores. The core of each seed is defined as the region where the seed can be moved without altering the segmentation result. As opposed to what we could think, the core is different from the catchment basin. We also analyze the inverse problem of segmentation, i.e., the possible sets of seeds that correspond to a target segmentation: the so-called minimum sets of seeds.

Three segmentation methods are focused: the iterative relative fuzzy object extraction (IRFOE) [USL02], the watershed transform by image foresting transform (IFT-WT) [FSL04] and the watershed transform by minimum spanning forest (MSF-WT) [Mey94b]. As the two WT approaches do not return unique solution, which is contrary to the idea of robustness, we use their respective tie-zone transforms [ALC05] (TZ-IFT-WT and TZ-MSF-WT) which unify the set of possible solutions.

This paper has many theoretical contributions: a general characterization of the cores for the three segmentation methods, the demonstration of equivalence between TZ-IFT-WT and TZ-MSF-WT methods, the relationship between cores and non-redundant receptive regions under the framework of minimal seed sets, and the extension of minimal seed sets at pixel level with a new algorithm for their computation.

The organization of the paper is as follows. We recall the basic definitions of the three segmentation methods based on watershed and fuzzy connectedness in Section 6.2. Section 6.3 deals with robustness problem: Core definitions for IRFOE and TZ-IFT-WT are recalled in Section 6.3.1. Section 6.3.2 demonstrates that the sets of shortest-path forests with max-arc path cost and minimum spanning forests give identical segmentation, then shows that the core is identical for the three methods (IRFOE, TZ-IFT-WT and TZ-MSF-WT). Section 6.4 recalls the minimal seed set framework, extends it at pixel level, relates it to cores and presents the new algorithm.

6.2 Seeded segmentation methods

In this section, we briefly recall the basic definitions of the three segmentation paradigms focused in the paper: the watershed transform by image foresting transform (IFT-WT), the watershed transform by minimum spanning forest (MSF-WT), and the iterative relative fuzzy object extraction

(IRFOE).

6.2.1 Watershed by Image Foresting Transform

The watershed transform (WT) is a famous and powerful segmentation tool in morphological image processing, first introduced by Beucher and Lantuéjoul [BL79] for contour detection and applied in digital image segmentation by Beucher and Meyer [BM93].

Under the image foresting transform (IFT) framework [FSL04], the watershed transform is viewed as a graph optimization problem: creation of a shortest-path forest (SPF).

An image is interpreted as a weighted graph $G = (V, A, w)$ consisting of a set V of nodes representing the image pixels, a set A of arcs weighted by w , a function from A to some nonnegative scalar domain. $N(v)$ denotes the neighborhood of node v , i.e. the set of nodes adjacent to it. Nodes u and v are adjacent when the arc $\langle u, v \rangle$ belongs to A . A graph (V', A') is subgraph of (V, A) if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$. A forest F of G is an acyclic subgraph F of G . Trees are connected components of the forest (any two nodes of a tree are connected by a path). A path $\pi(u, v)$ from node u to node v in graph (V, A, w) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said simple if all its nodes are different from each other. The path is trivial when it consists of a single node $\langle v \rangle$. A path-cost function f assigns to each path π a path cost $f(\pi)$, in some totally ordered set of cost values.

Let $S \subseteq V$ be a set of particular nodes s_i called seeds. For a given weighted graph (V, A, w) and a set S of seeds, the *image foresting transform (IFT)* returns a forest SPF of (V, A, w) such that (i) there exists for each node $v \in V$ a unique and simple path $\pi(s_i, v)$ in SPF from a seed node $s_i \in S$ to v and (ii) each such path is optimum, i.e., has a minimum cost for linking v to some seed of S , according to the specified path-cost function f .

The *watershed transform by IFT (IFT-WT)* assumes that the *max-arc* path-cost function f_{\max} is used:

$$\begin{aligned} f_{\max}(\langle v_1 \rangle) &= 0 \\ f_{\max}(\langle v_1, v_2, \dots, v_n \rangle) &= \max_{i=1, \dots, n-1} \{w(v_i, v_{i+1})\} \end{aligned} \quad (6.1)$$

where $w(u, v)$ is the symmetric weight of arc $\langle u, v \rangle \in A$, ideally higher on the object boundaries and lower inside the objects.

Usual arc weight functions are: $w_1(u, v) = |I(u) - I(v)|$, $I(u)$ being the intensity of pixel u ; $w_2(u, v) = \max \{G(u), G(v)\}$, where $G(u)$ is the (morphological) gradient of image I at pixel u .

A label map L assigns to each node v the label $L(v)$ of the corresponding minimum-path root.

The catchment basins correspond to the (labeled) trees: $CB_{IFT}(s_i) = \{v \in V, L(v) = L(s_i)\}$. Note that the final cost map C is unique and corresponds to the morphological superior reconstruction of the image from the seeds using a flat structuring element. However, the forests and then the labelings may be multiple.

Instead of choosing one of the many labelings, the *tie-zone concept* [ALC05] can be applied to unify the multiple solutions of a WT: Regions where possible labelings differ are put in the tie zone (TZ). In the case of IFT-WT, the *tie-zone watershed by IFT* (TZ-IFT-WT), returns a unique partition of the image made of catchment basins and tie zone:

$$\begin{aligned} CB_{TZ-IFT}(s_i) &= \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\} \\ TZ_{IFT} &= V \setminus \bigcup_i CB_{TZ-IFT}(s_i) \end{aligned} \quad (6.2)$$

where Φ denotes the set of the shortest-path forests F (IFTs).

6.2.2 Watershed by Minimum Spanning Forest

Another watershed transform can be defined from the minimum spanning forest. A minimum spanning forest (MSF) for graph (V, A, w) is a forest (V, A^*) whose total weight $\sum_{a \in A^*} w(a)$ (sum of the weights of its arcs) is minimum and where each node $v \in V$ is linked to a seed $s_i \in S$ by a unique simple path. The *watershed transform by minimum spanning forest* (MSF-WT) introduced in [Mey94a] is applied on a weighted neighborhood graph whose nodes are the primitive catchment basins corresponding to regional minima of the image, and whose arcs link neighbor catchment basins and are weighted by the altitude of the pass between them. In fact, the same paradigm can be applied on graph where nodes are pixels and arcs have higher weight between dissimilar nodes (like a gradient). The many possible MSFs on a weighted graph define partitions that are solutions of this WT: Each tree of the MSF is a catchment basin of the MSF-WT, and nodes can be labeled in accordance with their seed label. Tie-zone transform can also be applied to unify the multiple solutions and get a unique labeling.

For a given weighted graph, the set of MSF-WT solutions is a subset of the set of IFT-WT solutions using the max-arc path cost.

Proposition 2 ([AL07c]) *Any minimum spanning forest (MSF) is also a shortest-path forest (SPF-max) using max-arc path cost f_{\max} .*

Reciprocal is false as illustrated in Fig. 6.2: Three SPF-max can be built from a weighted graph and its two seeds A and B (represented by \circ). Cutting any arc results in a SPF-max. Yet, we must cut one of the arcs of weight 2 to obtain MSFs of total weight 3 (cf. Fig. 6.2(a)(b)).

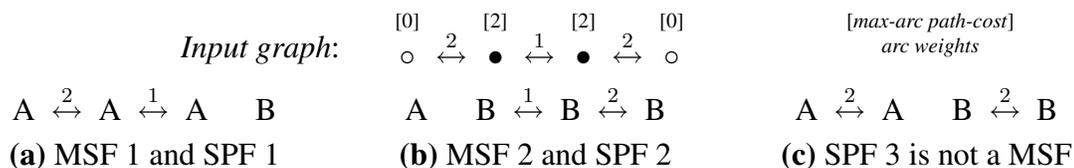


Fig. 6.2: A weighted graph with two seeds (\circ) and its 3 possible SPF-max and 2 MSFs.

6.2.3 Fuzzy connectedness approach

Many approaches of segmentation by fuzzy connectedness have been proposed. We focus on one of them: the iterative relative fuzzy object extraction (IRFOE) [USL02]. It is shown in ref. [AL06] that this approach is the dual of the TZ-IFT-WT: Fuzzy affinity $\mu_{\kappa}(u, v)$ (similarity measure) between nodes u and v plays the role of arc weight $w(u, v)$; path strength defined as the minimum affinity in the path corresponds to the max-arc path cost; and the strength of fuzzy connectedness $\mu_{\kappa}(s, u)$ between two nodes (maximum strength of the paths between them) is the dual of the optimal (minimum) path cost. The subsequent robustness results developed in the IFT-WT framework will therefore be valid for the IRFOE approach too.

Observe that this approach uses a unique symmetric affinity function. If multiple asymmetric affinity functions are used as in ref. [HC01], the duality with TZ-IFT-WT is no longer valid. Such a segmentation does not respect any global optimality criterion, but is similar to a flooding process where optimization is only achieved temporarily at each flooding level but not necessarily at the end of the segmentation process. Although it is possible to define cores for robustness of such a paradigm, we will not deal with it because more complex and longer mathematical treatment would be necessary.

6.3 Robustness regions for seeds

Now the three segmentation methods of interest have been presented, the seed-relative robustness issue can be dealt with. The problem addressed in refs. [USL02, AL06] is to find the *cores*, i.e., the regions where seeds can be moved without altering the segmentation (or labeling). In other words, the spans of the trees must be the same after moving the seeds within the cores (trees may change). We recall the theoretical characterization [USL02, AL06] in the case of IFT-WT and its dual IRFOE, then, extend the core to the case of MSF-WT.

6.3.1 Cores for the shortest-path forests and the iterative relative fuzzy object

In the case of IRFOE, the core relative to a seed corresponds to its relative fuzzy object (RFO) [USL02], i.e., the set of pixels that have strictly higher strength of connectedness with this seed than

with any other seed:

$$K_s = \{v \in V, \mu_K(s, v) > \mu_K(t, v), \forall t \in S, t \neq s\} \quad (6.3)$$

The robustness property of the core was thoroughly demonstrated in ref. [USL02].

By duality, the core relative to a seed for the TZ-IFT-WT was also defined [AL06]: it is the set of pixels that have at least a path from this seed with strictly lower cost than any path from other seeds. As the cost of the minimum path from a seed $s \in S$ to any pixel v can be given by the superior reconstruction $C_s[v]$ of the image from this seed, the core K_s relative to a seed s is defined by:

$$K_s = \{v \in V, C_s[v] < C_t[v], \forall t \in S, t \neq s\} \quad (6.4)$$

Note that the pixels with paths of same (non-strict) minimum cost linking to different seeds do not belong to any core. They can belong either to the tie-zone (if these paths are optimal, i.e., for all path nodes, the cost is minimum, not only for the final node); or to the protected zone (complement of the core in a catchment basin) of a catchment basin.

An equivalent definition of the core uses the pass-value concept. The pass-value of a path is equal to its highest arc: $\epsilon(\pi) = \max_{(p,q) \text{ in } \pi} \{w(p, q)\} = f_{\max}(\pi)$. The pass-value between two nodes p and q is equal to the minimum pass-value of the paths between them: $\epsilon(p, q) = \min_{\forall \pi: p \rightsquigarrow q} \{\epsilon(\pi)\} = C_p[q]$. The pass-value of a seed s is equal to the minimum of the pass-values between s and any other seed: $\epsilon(s) = \min_{\forall t \in S, t \neq s} \{\epsilon(s, t)\}$. The core is then defined by:

$$K_s = \{v \in V, \epsilon(s, v) < \epsilon(s)\} \quad (6.5)$$

6.3.2 Cores for the minimum spanning forests

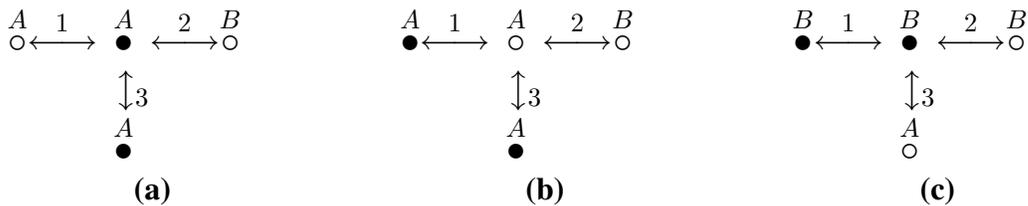


Fig. 6.3: Non-robustness of the MSF when the seed (o) is moved within a tree.

As opposed to what one could think (p. 81, [Mey94a]), the catchment basins defined by the MSF-WT are not robustness regions. Figure 6.3 shows a counter-example. The MSF-WT labeling of the graph is stable when the seed (represented by o) of tree A is moved within its core (cf. Fig. 6.3(a)-(b):

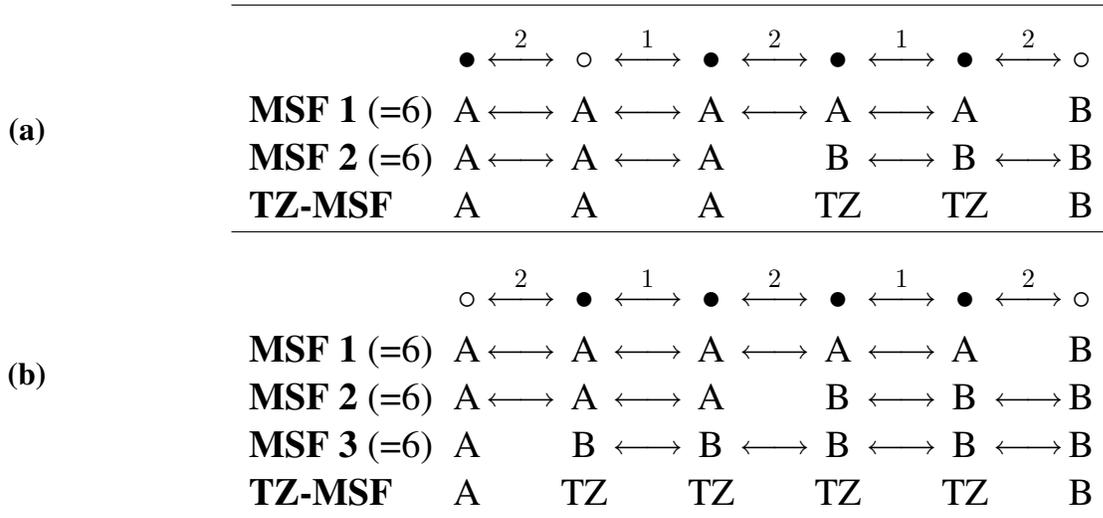


Fig. 6.4: Non-robustness of the set of MSFs when the seed (\circ) is moved within a tree.

after cutting arc 2, minimum weight is 4). It does nevertheless change when seed A is moved out of its core, though still within its initial tree (cf. Fig. 6.3(c): after cutting arc 3, minimum weight is 3).

Many MSF-WTs are sometimes possible. In this case, we should consider a unified partition (TZ-MSF-WT) built from these solutions by tie-zone transform. However, the catchment basins, complement of the tie-zone, do neither constitute a robustness region for the seed choice. Figure 6.4(a) shows a counter-example where two MSFs of weight 6 can be built from a graph with two seeds A and B . If seed A is moved out of its core, though within its catchment basin (Fig. 6.4(b)), a third MSF of weight 6 is possible, which modifies the TZ-MSF.

Therefore, we demonstrate that the robustness regions are contained in the trees and correspond to the cores characterized by Eqs. 6.4 and 6.5. Indeed, the result of the segmentation by TZ-MSF-WT is stable when we modify the position of a seed within the core of its tree: It will not change the labeling associated to the set of MSFs. To prove it, we propose the following theorem.

Theorem 3 *Let (V, A, w) be a weighted graph, w a nonnegative weight function. Let $S \subseteq V$ be a set of seeds. The tie-zone transform of the minimum spanning forests is equal to the tie-zone transform of the shortest path forests using the max-arc path cost function f_{\max} :*

$$\begin{aligned} TZ_{MSF} &= TZ_{SPF-\max} \\ CB_{TZ-MSF}(s) &= CB_{TZ-SPF-\max}(s), \forall s \in S \end{aligned}$$

Consequently, if the seeds s are chosen randomly in their respective cores K_s , the TZ-IFT-WT

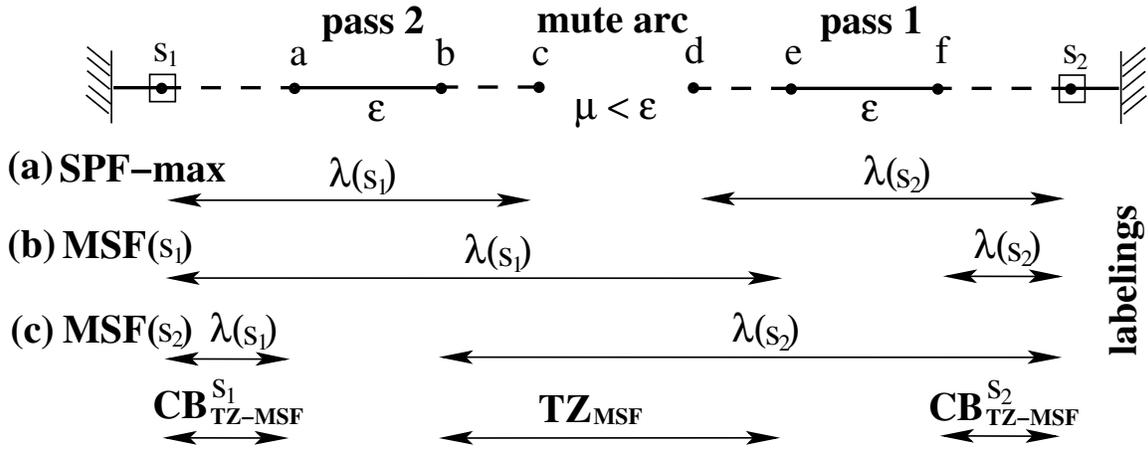


Fig. 6.5: Extreme pass and mute arcs in cycle σ^{cd} .

(TZ-SPF-max) is unchanged, and so is the TZ-MSF-WT. Therefore the cores for the TZ-MSF-WT method are characterized by the same Eqs. 6.4 and 6.5.

Demonstration of Theorem 3: If $p \in TZ_{MSF}$, there are at least two MSFs that give different labeling for p . According to Proposition 2, these MSFs are also IFT-WT. As there are at least two IFT-WTs that give different labelings for p , $p \in TZ_{IFT-WT}$. If $p \in CB_{TZ-IFT-WT}(s)$, all the IFT-WTs give the same labeling for p . So, all the MSFs, which are necessarily IFT-WTs, give the same labeling for p . So, $p \in CB_{TZ-MSF}$. So we have: $TZ_{MSF} \subseteq TZ_{IFT-WT}$ and $CB_{TZ-IFT-WT}(s) \subseteq CB_{TZ-MSF}(s), \forall s \in S$.

Now we demonstrate that the IFT-WTs (or shortest path forests with f_{\max} , SPF-max) that are not MSFs do not affect the tie zone of the MSFs. Note that the MSF of a weighted graph (V, A, w) can be viewed as a minimum spanning tree of the same graph to which a fictitious root node z was added. Arcs of weight -1 link z to each seed. The minimum spanning tree satisfies the following theorem:

Theorem 4 (Minimum spanning tree [GM84]) (V, T) is a tree of minimum weight for graph (V, A, w) if and only if for every arc $u \in A \setminus T$ the cycle σ^u (such that $\sigma^u \subset T \cup \{u\}$) satisfies: $w(u) \geq w(v)$, $\forall v \in \sigma^u$ ($v \neq u$).

Let $F = (V, A')$ be an SPF-max of graph (V, A, w) that is not MSF (cf. Fig. 6.5(a)). F is not MSF. So, considering the derived tree $(V \cup \{z\}, T)$ and Theorem 4, there exists at least one arc $\langle c, d \rangle \in A \setminus A'$ such that: if this arc is added, a cycle $\sigma^{cd} \subset T \cup \{\langle c, d \rangle\}$ is created and $w(c, d) = \mu < \epsilon$, $\epsilon = \max_{u \in \sigma^{cd}} \{w(u)\}$. Suppose the cycle includes two seeds s_1 and s_2 and $\langle a, b \rangle$ is one of the arcs of maximum weight in the cycle: $w(a, b) = \epsilon$. Without loss of generality, say that it is the nearest arc from s_1 with such a weight, so it is between s_1 and $\langle c, d \rangle$ in F (cf. Fig. 6.5). We call it the extreme pass arc for s_2 . The path cost $C[c]$ at c in F is: $C[c] = f_{\max}(\pi(s_1, c)) = \epsilon$.

As F is an SPF-max, $f_{\max}(\pi(s_1, c)) \leq f_{\max}(\pi(s_2, c))$. So, $\epsilon \leq \max\{f_{\max}(\pi(s_2, d)), \mu\}$. If $f_{\max}(\pi(s_2, d)) \leq \mu$, the previous inequality would lead to $\epsilon \leq \mu < \epsilon$, which is impossible. Consequently, $\epsilon \leq f_{\max}(\pi(s_2, d))$. As ϵ is the maximum weight in the cycle σ^{cd} , the path cost $C[d]$ at d in F is necessarily: $C[d] = f_{\max}(\pi(s_2, d)) = \epsilon$. Hence, there exists at least one arc $\langle e, f \rangle$ between d and s_2 in F , such that: $w(e, f) = \epsilon$. Suppose that, if there are many of such arcs, it is the nearest one from s_2 . We call it the extreme pass arc for s_1 .

Now, let us add arc $\langle c, d \rangle$ and remove $\langle e, f \rangle$, the extreme pass arc for s_1 , so as not to introduce a cycle in the forest. Therefore, labels of nodes d to e change: $\lambda(s_1)$ instead of $\lambda(s_2)$ (cf. Fig. 6.5(b)). The total weight of the forest decreases by $\epsilon - \mu$. But path costs at nodes d to e remain equal to ϵ . So, the forest remains a SPF-max.

Then, we search for other arcs $\langle c, d \rangle$ and $\langle e, f \rangle$ and make substitution again to iteratively decrease the weight of the forest until a minimum weight. When no more such arcs are found, the SPF-max will have become an MSF (cf. Theorem 4). Observe that the cycle σ^{cd} created may include only one seed or even no seed. In this case, we make the same substitution to decrease the weight of the forest, but this does not affect the labeling.

In the case of including two seeds, suppose that at each substitution, advantage is given to s_1 . So, the $MSF(s_1)$ will show the maximum extension of s_1 (cf. Fig. 6.5(b)). Now, the same series of substitutions is made but giving advantage to $s_2, s_3, \dots \in S$ to get $MSF(s_2), MSF(s_3), \dots$ (cf. Fig. 6.5(c)). Clearly, any labeling of node in an SPF-max has already been proposed by some of the $MSF(s_i)$ labelings. Thus, the labelings given by the SPF-max that are not MSFs do not affect the tie zone transform of the MSFs. Therefore, the tie-zone transforms of SPF-max (IFT-WT) and MSF are identical. ■

6.4 Minimal sets of seeds

We now look at the inverse problem of segmentation by watershed: the minimal seed set (MSS) problem. Given a partition obtained by WT, it consists in finding the minimal sets of seeds that reproduce the same segmentation by WT. When first addressed in ref. [LS02], it was developed only in the case of segmentation by MSF-WT where the graph nodes were primitive catchment basins (i.e., resulting from a classical watershed from regional minima). The problem is relevant and particularly interesting in the applications of video segmentation. From an initial segmentation made by the user, it is possible to minimize the set of seeds to recover this input segmentation (first frame). Then, we only have to propagate this minimal set of seeds to segment the next frames, estimating the new position of few seeds. Thus, it decreases the dimension of the estimation problem. Here, we extend the problem of MSS to any given partition where each segmented region corresponds to a connected

region with differentiated label. Thus, the input partition can vary at the pixel level precision, instead of the primitive basin level. We propose an algorithm to compute the non-redundant receptive regions in each of which at least one pixel seed has to be selected to recover the input partition. We also show the relation between the particular regions of the MSS and WT seed robustness frameworks: core, catchment basin, tie-zone, non-redundant receptive region, redundant receptive region, and dead zone.

6.4.1 Definitions

For completeness of the article, we first recall the main definitions of ref. [LS02].

Let L be the aimed *labeling* representing a segmentation (or partition) of the image-graph (V, A, w) : each segment or part of the partition is connected and represented by a different label $\lambda \in \Lambda$. The *frontier nodes* $F(\lambda)$ of a segment λ are those nodes neighbors of other segment(s):

$$F(\lambda) = \{p \in V, \exists q \in N(p), L[q] \neq L[p] = \lambda\} \quad (6.6)$$

The *strength* $\mathcal{S}(p)$ of a frontier node p is the minimum cost of the paths that connect it to other segments, i.e. the minimum weight of the arcs linking to neighbors of different label:

$$\mathcal{S}(p) = \min \{w(p, q), q \in N(p), L[q] \neq L[p]\} \quad (6.7)$$

The *receptive region* $RR(p)$ associated to frontier node p is the set of nodes of the same segment which are linked to p by a path whose cost is lower than the strength of p :

$$RR(p) = \{q \in V, L[q] = L[p], \exists \pi(p, q), f_{\max}(\pi(p, q)) < \mathcal{S}(p)\} \quad (6.8)$$

Property 1 *If the receptive regions of two frontier nodes of a segment intersect, that of greater strength contains the other: $p, q \in F(\lambda), RR(p) \cap RR(q) \neq \emptyset, \mathcal{S}(p) \geq \mathcal{S}(q) \Rightarrow RR(p) \supseteq RR(q)$.*

A receptive region is *redundant* if it strictly contains another receptive region:

$$RR(p) \text{ redundant} \Leftrightarrow \exists q, RR(q) \subset RR(p) \quad (6.9)$$

otherwise, it is said *non-redundant*. The *dead zone* $DZ(\lambda)$ of a segment λ consists of the nodes of the segment which are in none of its receptive region:

$$DZ(\lambda) = \{p \in V, L[p] = \lambda, p \notin RR(q), \forall q \in F(\lambda)\} \quad (6.10)$$

We build a *minimal seed set* by choosing one seed node in each non-redundant receptive region of each segment, assigned by the respective label. Labeling L is thereby recovered by WT from any minimal seed set. Additional seeds in either dead zone, redundant or non-redundant regions have no effect on segmentation, as long as they are assigned to their respective label.

6.4.2 Relationship with cores and tie-zone

Looking at these definitions, we see that the minimum seed set determination must be related to the problem of finding cores. Indeed, they are very close problems. Both aim to a goal partition stability. The great difference between the core robustness and the minimum seed set problems is that the former deals with the stability of the overall tree spans (not the trees) while the latter focus on the stability of the label span. In general, many seeds (therefore, trees) can share the same label of the target partition. In sum, they are similar problems but their level of application is different. Moreover, their inputs are different: for given tree seeds, the cores are determined as the regions where they can move while either these seeds and the receptive regions where they can move are unknown.

In fact, we demonstrate that the core problem is implicit in the MSS problem: Once the MSS problem has been solved, one has to pick a seed in each non-redundant receptive region (NRRR). At that moment, the core problem is automatically solved for all seeds: *the core corresponds to the respective NRRR*.

Demonstration: Let $RR(p), p \in F(\lambda)$ be a NRRR and p the seed we picked to constitute part of a minimum seed set S . From relation 6.9, there does not exist q such that $RR(q) \subset RR(p)$, i.e. if $\forall q \in F(\lambda), RR(p) \cap RR(q) \neq \emptyset, \Rightarrow RR(p) \subseteq RR(q)$. And according to Property 1, $\mathcal{S}(p) \leq \mathcal{S}(q)$ ¹ for such frontier nodes q . Using the pass-value definition (cf. Section 6.3.1) and Eq. 6.8, we have:

$$RR(p) = \{v \in V, L[v] = L[p], \epsilon(p, v) < \mathcal{S}(p)\}. \quad (6.11)$$

As $\mathcal{S}(p)$ is the minimum weight of arcs linking to a node of different label (cf. Eq. 6.7), the pass-value between seed p and any other seed of different label s is:

$$\epsilon(p, s) \geq \mathcal{S}(p), \forall s \in S, L[s] \neq L[p]. \quad (6.12)$$

Consider now the seeds $t \in S$ that have the same label $L[t] = L[p]$ but do not belong to the same NRRR: $t \notin RR(p)$. Hence, $\forall \pi(p, t), f_{\max}(\pi(p, t)) \geq \mathcal{S}(p)$ (cf. Eq. 6.8) and so:

$$\epsilon(p, t) \geq \mathcal{S}(p), \forall t \in S, L[t] = L[p]. \quad (6.13)$$

¹We will take advantage of this result in our algorithm (Section 6.4.3).

It results from Eqs. 6.11, 6.12 and 6.13 that: $RR(p) = \{v \in V, L[v] = L[p], \epsilon(p, v) < \epsilon(p)\}$, which is equivalent to core K_p (cf. Eq. 6.5). ■

If more seeds than the minimum number are chosen, the cores, which make the spans of each individual tree stable, do no longer correspond to the NRRRs and must be determined separately. The labeling, yet, will be stable. If just a seed per NRRR is picked and assigned by its respective target label, the set of NRRRs referring to a label can be viewed as a “distributed minimum core relative to the label”.

Observe that there is no tie-zone in the MSS because the target partition has only labeled regions without tie-zone. Indeed, allowing multiple NRRRs with the same label avoids the creation of TZ. As the TZ transform applied on any minimal set of seeds results in the target partition without tie-zone, the catchment basins (that correspond to tree spans) are the same for MSF-WT, IFT-WT, TZ-MSF-WT and TZ-IFT-WT (due to Theorem 3).

6.4.3 Algorithm

The brute force algorithm for computing the MSS would consist in computing the receptive regions of all frontier nodes, and their possible intersections to conserve only the non-redundant ones. Thus, the image pixels might be processed many times. The proposed algorithm uses two optimizations so as to process each pixel at most only once, and less when there are redundant regions or dead zone:

1. During the detection of receptive regions $R(p)$, nodes whose path cost is greater than or equal to strength $S(p)$ are not processed.
2. The receptive regions $R(p)$ are detected in increasing order of strength $S(p)$.

The input graph must have symmetric arcs, the nodes may represent either pixels or primitive catchment basins. In lines 1 and 2, frontier pixels are detected and strengths computed, according to Eqs. 6.6 and 6.7. Then, for each label λ , the receptive regions $R(p)$ are detected in increasing order of strength $S(p)$ (lines 3–5).

The `DetectRR` function (lines 7–18) has a main loop (lines 9–17) that uses an ordered queue Q to propagate frontier node index p in the nodes u whose path cost is less than the strength of p . Note that index is propagated to the only neighbors with same label $L[u] = L[p]$ and which have not already been definitively processed (line 13). Minimum path cost is computed and updated in lines 14–17 and node u is inserted in Q . The first optimization is made in line 15. Propagation is interrupted for the nodes with path cost greater than the strength s of p . These nodes are neither inserted nor removed from Q . So, not all the nodes should be processed, as it is the case of the nodes in the dead zones for example.

Each node is removed at most once from queue Q (lines 10-12). Map P , initialized with value NIL, associates to the processed (removed) node v the index of the propagated frontier pixels p if $v \in RR(p)$ (line 12). But before that, we check if this node v has already been associated to another $RR(p')$ (line 11). If so, the propagation loop is interrupted (second optimization). It means that $RR(p')$ intersects $RR(p)$, and as $S(p') \leq S(p)$ (increasing processing order), $RR(p)$ is redundant. It is not worth computing it. In this case, p is not a frontier node that propagated its index into a non-redundant receptive region. So, it does not belong to the set NRI of indices of frontier nodes resulting in a non-redundant receptive region (line 18).

In a final step (line 6), we copy the index of the respective frontier node located in $P[p]$ into map $NR[p]$, when p belongs to an NRRR. Indeed, map P contains some frontier node indices whose propagation was interrupted.

Algorithm 4: Non-redundant receptive region (NRRR) detection

Inputs: - image (V, A, w) (neighborhood N derived from A);
 - labeling $L : V \rightarrow \Lambda$.

Output: - non-redundant receptive region map NR (returns for each node in NRRR the index of the respective frontier node, NIL if it is not in NRRR; initialized with NIL).

Auxiliary Data: - map P that returns for the processed nodes the index of the propagated frontier pixels, NIL otherwise (initialized with NIL);
 - set NRI of indices of frontier nodes resulting in a non-redundant receptive region (empty at initialization).

1. $F \leftarrow \text{Find_Frontier_Pixels}(L, V, N)$;
2. $S \leftarrow \text{Find_Strength}(F, w, N, L)$;
3. $\forall \lambda \in \Lambda$,
4. $\forall p \in F$ **and** $L[p] = \lambda$, **in increasing strength $S(p)$ order**,
5. $P, NRI \leftarrow \text{DetectRR}(p, S(p), L, P, NRI, w, N)$;
6. $\forall p \in V$, **if** $P[p] \in NRI$, $NR[p] \leftarrow P[p]$;

$\text{DetectRR}(p, s, L, P, NRI, w, N)$:

Auxiliary Data: empty ordered queue Q , state flag *done*,
 cost map C , cost variable c .

7. $\forall v \in V$, $done(v) \leftarrow \text{FALSE}$; $C[v] \leftarrow \infty$;
8. $C[p] \leftarrow 0$; $\text{Insert}(p, Q, C[p])$; $nobreak \leftarrow \text{TRUE}$;
9. **while** $\text{IsEmpty}(Q) = \text{FALSE}$,
10. $v \leftarrow \text{RemoveMin}(Q)$;
11. **if** $P[v] \neq \text{NIL}$, $nobreak \leftarrow \text{FALSE}$; **Break**;
12. $P[v] \leftarrow p$; $done(v) \leftarrow \text{TRUE}$;
13. $\forall u \in N(v)$, $done(u) = \text{FALSE}$ **and** $L[u] = L[p]$,
14. $c \leftarrow \max\{C[v], w(v, u)\}$;
15. **if** $c < C[u]$ **and** $c < s$,
16. **if** u in Q , $\text{Remove}(u, Q)$;
17. $C[u] \leftarrow c$; $\text{Insert}(u, Q, C[u])$;
18. **if** $nobreak = \text{TRUE}$, $NRI \leftarrow NRI \cup \{p\}$;

6.5 Conclusion

This paper analyzed the problem of robustness relative to input seeds for three segmentation approaches: the iterative relative fuzzy object extraction, the watershed transforms by image foresting transform and by minimum spanning forest. It dealt with the characterization of the seed cores on the one hand, and with the minimal sets of seed, on the other hand.

New theoretical results which were not intuitive nor trivial were demonstrated. They allow to link around the robustness topic problems and approaches that were known as a-priori different, like shortest-path forest, minimum spanning forest, fuzzy connectedness and watershed. So we showed that the set of shortest-path forests with max-arc path cost gives the same segmentation as the set of minimum spanning forests, and, therefore, that they have the same core characterization as in the fuzzy connectedness framework. The equivalence between cores of a minimal seed set and non-redundant receptive regions was proved too. Moreover, a new algorithm was proposed for the computation of minimal seed sets at pixel level.

We hope these theoretical results will contribute to the understanding and evaluation of segmentation robustness. Robustness measures based on core area and distribution could be useful to prevent not very repeatable segmentation results, or evaluate the effect of a prefiltering stage. The visualization of minimum seed sets, redundant receptive regions and dead zones could help the user with interactive segmentation routines. For example, he/she could select only the satisfactory regions of a first segmentation, and compute the minimal seed sets, which could guide him/her for a refined segmentation by showing that it is or not worth picking seeds with some label in some region.

Capítulo 7

Conclusão

Esta tese tem várias contribuições originais relacionadas ao *watershed*, principal método de segmentação da morfologia matemática. Elas podem ser divididas em duas linhas de pesquisa.

Primeiro, investigou-se o problema de se ter uma transformada de *watershed* que tivesse uma solução única e consistente com sua definição. Por isso, foi proposta a transformada em zona de empate no quadro da transformada imagem-floresta (IFT). A zona de empate unifica as múltiplas soluções relativas a um método específico de *watershed*. A unicidade da solução permite não depender do algoritmo ou da sua implementação. Portanto, proporciona robustez à segmentação em relação à implementação do método. No entanto, a zona de empate pode ser vista como uma linha de *watershed* que não é necessariamente separadora e possivelmente espessa. Os pixels da zona de empate não são atribuídos a nenhum segmento em especial, o que pode ser indesejável. Propôs-se, então, um afinamento da zona de empate que mantivesse uma solução única. Diante do desafio de se obter uma transformada de *watershed* única, consistente e com linha fina (ou nenhuma linha no caso ideal), percebeu-se que há um compromisso entre unicidade da solução e espessura da linha de *watershed* ou da zona de empate.

Segundo, uma análise teórica das diversas transformadas de *watershed* e das técnicas de segmentação baseadas em conexidade nebulosa permitiu o melhor entendimento de suas diferenças e semelhanças. A introdução da zona de empate foi um passo decisivo e necessário para descobrir as relações entre estes métodos, nunca estudadas antes.

Essas contribuições de natureza teórica têm também suas conseqüências práticas. Elas permitem desenvolver, por exemplo, algoritmos mais rápidos e com qualidade (resultado fiel à definição). Possibilita ainda transpor e aproveitar técnicas desenvolvidas por cada um dos paradigmas como, por exemplo, o pré-processamento da imagem que cria afinidades baseadas na escala local de sua textura [SU00].

Em conseqüência das relações teóricas que interligam as transformadas de *watershed*, a segmen-

tação dada pela transformada em zona de empate do *watershed* via IFT tornou-se o denominador comum dos *watersheds* estudados: Ela retorna os segmentos mínimos, incontestáveis, logo, confiáveis. Estes são úteis para aplicações em que a confiabilidade nas medidas dos objetos segmentados é crucial.

De maneira complementar, uma zona de empate espessa pode ser um indicador de ambigüidade na segmentação. A Figura A.4(f) mostra, por exemplo, que a zona de empate, após uma filtragem dos mínimos regionais, é reduzida aos contornos finos dos objetos de interesse. Uma zona de empate fina denota a ausência de ambigüidade e, então, uma filtragem robusta: Neste caso, qualquer algoritmo de *watershed* poderia ser utilizado sem introdução de viés. Logo, a zona de empate pode servir de realimentação num processo de filtragem automática.

Como outra repercussão prática, aprofundou-se a questão da robustez da segmentação por *watershed* em relação às sementes de entrada. A visualização dos núcleos pode ser uma realimentação útil num processo de segmentação automático ou semi-automático.

Inúmeros métodos de segmentação [HEMK98, KK03, MOS03, ZJ04] ou compressão de imagens [MEF02] partem de uma supersegmentação dada por um algoritmo de *watershed* qualquer (muitas vezes o algoritmo popular de Vincent e Soille [VS91]) a partir de todos os mínimos regionais da imagem. Em seguida, eles propõem fusionar as micro-bacias da supersegmentação usando diversos critérios [AC01, Pat05, NJ06, DSFM06] até se obter uma segmentação aceitável. Esses métodos fazem a hipótese que todos os contornos interessantes dos objetos são contidos nesta supersegmentação. Ora, isto não é necessariamente verdade, já que a forma das micro-bacias retornadas por tal algoritmo (cf. Figura A.4(a)) é viesada quando a zona de empate é espessa (cf. Figura A.4(d)). Utilizar as bacias mínimas (complementares da zona de empate) seria uma hipótese mais segura, que não eliminaria contornos possivelmente interessantes, quando a aplicação pede alta precisão de medida [LCO⁺05].

Enfim, os resultados inéditos sobre problemas clássicos da teoria de grafos poderiam vir a ser utilizados em domínios de aplicação totalmente diferentes do processamento de imagens, onde a modelagem por floresta ótima é presente.

Entre os trabalhos futuros que se inscrevem na continuidade desta tese, citam-se, além das aplicações mencionadas acima: o estudo da zona de empate no caso do *watershed* topológico para ver possíveis relações com os outros métodos; a definição de medidas de robustez baseadas nos núcleos e na zona de empate para aplicações práticas; o estudo e a quantificação da robustez em relação ao ruído na imagem de entrada, isto é, a influência do ruído de entrada na segmentação final.

Referências Bibliográficas

- [AB94] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.
- [AC01] P.S.U. Adiga and B.B. Chaudhuri. An efficient method based on watershed and rule-based merging for segmentation of 3-d histo-pathological images. *Pattern Recognition*, 34(7):1449–1458, July 2001.
- [AL05] R. Audigier and R. Lotufo. Tie-zone watershed, bottlenecks and segmentation robustness analysis. In *XVIII Brazilian Symp. on Computer Graph. and Image Proc. (SIBGRAPI'05)*, pages 55–62, Natal, Brazil, October 2005. IEEE Press.
- [AL06] R. Audigier and R. Lotufo. Duality between the watershed by image foresting transform and the fuzzy connectedness segmentation approaches. In *XIX Brazilian Symp. on Computer Graph. and Image Proc. (SIBGRAPI'06)*, pages 53–60, Manaus, Brazil, October 2006. IEEE Press.
- [AL07a] R. Audigier and R. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'07)*, Belo Horizonte, Brazil, October 2007. IEEE Press. Accepted.
- [AL07b] R. Audigier and R. Lotufo. Uniquely-determined thinning of tie-zone watershed based on label frequency. *Journal of Mathematical Imaging and Vision*, 27(2):157–173, February 2007.
- [AL07c] R. Audigier and R. Lotufo. Watershed by image foresting transform, tie-zone, and theoretical relationships with other watershed definitions. In *8th International Symposium on Mathematical Morphology (ISMM'07)*, Rio de Janeiro, Brazil, October 2007. Accepted.

- [ALC05] R. Audigier, R. Lotufo, and M. Couprie. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, volume 2, pages 654–657, Genova, Italy, September 2005.
- [ALF04] R. Audigier, R. Lotufo, and A. Falcão. On integrating iterative segmentation by watershed with tridimensional visualization of MRIs. In *XVII Brazilian Symp. on Computer Graph. and Image Proc. (SIBGRAPI'04)*, pages 130–137, Curitiba, Brazil, October 2004. IEEE Press.
- [ALF06] R. Audigier, R. Lotufo, and A.X. Falcão. 3D visualization to assist iterative object definition from medical images. *Computerized Medical Imaging and Graphics*, 30(4):217–230, June 2006.
- [BBM⁺97] A. Bieniek, H. Burkhardt, H. Marschner, M. Nölle, and G. Schreiber. A parallel watershed algorithm. In *Proc. 10th Scandinavian Conference on Image Analysis (SCIA'97)*, pages 237–244, Lappeenranta, Finland, June 1997.
- [Ber58] C. Berge. *Théorie des graphes et ses applications*. Dunod, Paris, France, 1958.
- [Ber05] G. Bertrand. On topological watersheds. *Journal of Mathematical Imaging and Vision*, 22(2-3):217–230, May 2005. Special issue on Mathematical Morphology.
- [BL79] S. Beucher and Ch. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing, Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
- [BM93] S. Beucher and F. Meyer. The Morphological Approach to Segmentation: The Watershed Transform. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, Inc., New York (NY), USA, 1993.
- [BM98] A. Bieniek and A. Moga. A connected component approach to the watershed segmentation. In H. J. A. M. Heijmans and J. B. T. M. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 215–222. Kluwer Acad., Dordrecht, The Netherlands, 1998.
- [BM00] A. Bieniek and A. Moga. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907–916, June 2000.
- [Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

- [CB97a] M. Couprie and G. Bertrand. Topological grayscale watershed transformation. In *SPIE Vision Geometry V Proceedings*, volume 3168, pages 136–146, 1997.
- [CB97b] M. Couprie and G. Bertrand. Topological grayscale watershed transformation. In *SPIE Vision Geometry VI Proceedings*, volume 3168, pages 136–146, 1997.
- [CHK05] Bruno M. Carvalho, Gabor T. Herman, and T. Yung Kong. Simultaneous fuzzy segmentation of multiple objects. *Discrete Applied Mathematics*, 151(1-3):55–77, 2005.
- [CNB05a] M. Couprie, L. Najman, and G. Bertrand. Algorithms for the topological watershed. In Pascal Lienhardt Eric Andres, Guillaume Damiand, editor, *Proceedings of the 12th International Conference on Discrete Geometry for Computer Imagery (DGCI 2005)*, volume 3429, pages 172–182, Poitiers, France, January 2005. April 11-13.
- [CNB05b] M. Couprie, L. Najman, and G. Bertrand. Quasi-linear algorithms for the topological watershed. *Journal of Mathematical Imaging and Vision*, 22(2-3):231–249, May 2005. Special issue on Mathematical Morphology.
- [Coh91] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, March 1991.
- [Dij59] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959.
- [DL03] E. Dougherty and R. Lotufo. *Hands-on Morphological Image Processing*. SPIE – The International Society for Optical Engineering, Bellingham (Washington), USA, August 2003.
- [DSFM06] A. Duarte, A. Sánchez, F. Fernández, and A.S. Montemayor. Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic. *Pattern Recogn. Lett.*, 27(11):1239–1251, 2006.
- [FCL01] A.X. Falcão, B.S. Cunha, and R.A. Lotufo. Design of connected operators using the image foresting transform. *SPIE on Medical Imaging*, 4322:468–479, February 2001.
- [FM81] K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3–16, 1981.
- [FSL04] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, January 2004.

- [GM84] M. Gondran and M. Minoux. Trees and arborescences. In *Graphs and algorithms*, chapter 4, pages 129–159. John Wiley & Sons Ltd., 1984.
- [Gri92] M. Grimaud. A new measure of contrast: the dynamics. In E. Dougherty & J. Serra P. Gader, editor, *Image Algebra and Morphological Image Processing III*, volume 1769, pages 292–305, San Diego, CA, USA, 1992. SPIE.
- [GW93] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, New York (NY), EUA, 1993.
- [HC01] Gabor T. Herman and Bruno M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(5):460–474, 2001.
- [HEMK98] K. Haris, S. N. Estradiadis, N. Maglaveras, and A. K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12):1684–1699, 1998.
- [HG85] R. M. Haralick and Shapiro. L. G. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1):100–132, 1985.
- [KK03] Jong-Bae Kim and Hang-Joon Kim. Multiresolution-based watersheds for efficient image segmentation. *Pattern Recogn. Lett.*, 24(1-3):473–488, 2003.
- [KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [LCO⁺05] Gang Lin, Monica K. Chawla, Kathy Olson, John F. Guzowski, Carol A. Barnes, and Badrinath Roysam. Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. *Cytometry Part A*, 63(1):20–33, 2005.
- [LF00] R.A. Lotufo and A.X. Falcão. The Ordered Queue and the Optimality of the Watershed Approaches. In *5th International Symposium on Mathematical Morphology*, pages 341–350, Palo Alto (CA), USA, June 2000. Kluwer Academic.
- [LFZ02] R.A. Lotufo, A.X. Falcão, and F.A. Zampiroli. IFT-watershed from gray-scale marker. In *XV Brazilian Symp. on Computer Graph. and Image Proc.*, pages 146–152, Fortaleza, Brazil, October 2002. IEEE Press.
- [LS02] R.A. Lotufo and W. Silva. Minimal set of markers for the watershed transform. In *6th International Symposium on Mathematical Morphology (ISMM)*, pages 359–368, Sydney, Australia, 2002. CSIRO publications.

- [LTHS06] Y.C. Lin, Y.P. Tsai, Y.P. Hung, and Z.C. Shih. Comparison between immersion-based and toboggan-based watershed image segmentation. *IEEE Transactions on Image Processing*, 15(3):632–640, March 2006.
- [MB90] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1):21–46, 1990.
- [MEF02] S. Makrogiannis, G. Economou, and S. Fotopoulos. Region oriented compression of color images using fuzzy inference and fast merging. *Pattern Recognition*, 35(9):1807–1820, September 2002.
- [Mey94a] F. Meyer. Minimum spanning forests for morphological segmentation. In J. Serra and P. Soille, editors, *Mathematical morphology and its applications to image processing*, pages 77–84. Kluwer Academic Publishers, 1994.
- [Mey94b] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, (38):113–125, 1994.
- [Mey96] F. Meyer. The dynamics of minima and contours. In P. Maragos, R. W. Schafer, and M. A. Butt, editors, *Mathematical Morphology and its Application to Image and Signal Processing (3rd)*, Computational Imaging and Vision, pages 329–336. Kluwer Academic Publishers, Boston, 1996.
- [MOS03] Norberto Malpica, Juan E. Ortuño, and Andrés Santos. A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recogn. Lett.*, 24(9-10):1545–1554, 2003.
- [MR98] A. Meijster and J. B. T. M. Roerdink. A disjoint set algorithm for the watershed transform. In *Proc. IX European Signal Processing Conference (EUSIPCO'98)*, pages 1665–1668, Rhodes, Greece, September 1998. IEEE Computer Society.
- [NC03a] L. Najman and M. Couprie. Watershed algorithms and contrast preservation. In *Discr. Geom. Comp. Imagery*, volume 2886 of *LNCS*, pages 62–71. Springer, 2003.
- [NC03b] L. Najman and M. Couprie. Watershed algorithms and contrast preservation. In *Discrete geometry for computer imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 62–71. Springer, 2003.
- [NCB05] L. Najman, M. Couprie, and G. Bertrand. Watersheds, mosaics and the emergence paradigm. *Discrete Applied Mathematics*, 147(2-3):301–324, April 2005. Special issue on DGCI.

- [NJ06] Hieu T. Nguyen and Qiang Ji. Improved watershed segmentation using water diffusion and local shape priors. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 985–992, Washington, DC, USA, 2006. IEEE Computer Society.
- [NS94] L. Najman and M. Schmitt. Watershed of a continuous function. *Signal Process.*, 38(1):99–112, 1994.
- [Ots79] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, March 1979.
- [Pap92] T.N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Trans. Signal Process.*, 40(4):901–914, 1992.
- [Pat05] Luis Patino. Fuzzy relations applied to minimize over segmentation in watershed algorithms. *Pattern Recogn. Lett.*, 26(6):819–828, 2005.
- [PP93] N.R. Pal and S.K. Pal. A review on image segmentation techniques. 26(9):1277–1294, September 1993.
- [Prê93] F. Prêteux. On a distance function approach for gray-level mathematical morphology. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, chapter 10, pages 323–349. Marcel Dekker, Inc., New York (NY), USA, 1993.
- [RLLV07] V. Osma Ruiz, J.I. Godino Llorente, N. Saenz Lechon, and P. Gomez Vilda. An improved watershed algorithm based on efficient computation of shortest paths. *Pattern Recognition*, 40(3):1078–1090, March 2007.
- [RM00] J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41(1-2):187–228, January 2000. Special issue on mathematical morphology.
- [Sil01] W. Silva. Marcadores mínimos usando *watershed*. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, dezembro de 2001.
- [SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [SSWC88] P. K. Sahoo, S. Soltani, A. K.C. Wong, and Y. C. Chen. A survey of thresholding techniques. *Comput. Vision Graph. Image Process.*, 41(2):233–260, 1988.

- [SU01] P.K. Saha and J.K. Udupa. Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82(1):42–56, April 2001.
- [SU00] P.K. Saha, J.K. Udupa, and D. Odhner. Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation. *Computer Vision and Image Understanding*, 77(2):145–174, February 2000.
- [US96] Jayaram K. Udupa and Supun Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graph. Models Image Process.*, 58(3):246–261, May 1996.
- [USL02] J.K. Udupa, P.K. Saha, and R.A. Lotufo. Relative fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11):1485–1500, November 2002.
- [VS91] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.
- [ZJ04] L. Zhi and Y. Jie. Interactive video object segmentation: fast seeded region merging approach. *Electronics Letters*, 40(5):302–304, 2004.
- [ZUS06] Y. Zhuge, J.K. Udupa, and P.K. Saha. Vectorial scale-based fuzzy-connected image segmentation. *Computer Vision and Image Understanding*, 101(3):177–193, March 2006.

Apêndice A

Zona de Empate, Gargalos e Análise da Robustez da Segmentação

Este capítulo contém o artigo intitulado *Tie-Zone Watershed, Bottlenecks and Segmentation Robustness Analysis* [AL05]. Ele desenvolve o novo conceito de gargalo (*bottleneck*) para entender a origem da zona de empate. De uma maneira intuitiva, o gargalo é o ponto de contato entre as águas vindo de vales diferentes que estão sendo inundados. O artigo propõe visualizar e quantificar a influência desses gargalos (peso e distribuição) sobre a zona de empate e, logo, sobre a robustez da segmentação em relação ao método de *watershed* utilizado e sua implementação.

Este artigo foi uma análise inicial para entender a criação da zona de empate e mostrar que não dependia da presença de platôs na imagem. Porém, ele acabou não sendo tão importante quanto esperado para a articulação lógica dos resultados principais desta tese. Por isso, foi colocado em anexo.

Abstract

In a recent paper [ALC05], a new type of watershed (WS) transform was introduced: the tie-zone watershed (TZWS). This region-based watershed transform does not depend on arbitrary implementation and provides a unique (and thereby unbiased) optimal solution. Indeed, many optimal solutions are sometimes possible when segmenting an image by WS. The TZWS assigns each pixel to a catchment basin (CB) if in all solutions it belongs to this CB. Otherwise, the pixel is said to belong to a tie-zone (TZ). An efficient algorithm computing the TZWS and based on the Image Foresting Transform (IFT) was also proposed.

In this article, we define the new concept of “bottlenecks” in the watermerging paradigm. Intuitively, the bottlenecks are the first contact points between at least two different wave fronts. They are

pixels in the image where different colored waters meet and tie and from which may begin, therefore, the tie-zones. They represent the origin points or the access of the tie-zones (regions that cannot be labeled without making arbitrary choices). If they are preferentially assigned to one or another colored water according to an arbitrary processing order, as occurs in most of watershed algorithm, an entire region (its influence zone – the “bottle”!) is conquered together. The bottlenecks play therefore an important role in the bias that could be introduced by a WS implementation. It is why we show in this paper that both tie-zones and bottlenecks analysis can be associated with the robustness of a segmentation.

A.1 Introduction

The watershed (WS) transform is a well-known and powerful segmentation tool for morphological image processing. It was first introduced by Beucher and Lantuéjoul [BL79] for contour detection and applied in image segmentation by Beucher and Meyer [MB90]. Nowadays, there are many definitions and algorithms of watershed transforms in literature. Roerdink and Meijster [RM00] give a comparison of some of them. The algorithm of Vincent and Soille [VS91] is based on immersion simulation: the image is represented by a topography inundated by water that springs from regional minima. The watershed lines are dams constructed for separating the growing catchment basins (CB) corresponding to minima. The algorithm of Meyer [Mey94b] computes the WS transform by solving a shortest path problem with respect to a topographical distance function.

In the numerous WS algorithms, variations may first occur in the input: all regional minima; only imposed minima to avoid oversegmentation (WS from markers [MB90]); or grayscale markers [LFZ02] to specify the depth (handicap) of some imposed minima. Then, the output may be of different types: ‘line-algorithms’ return separating WS lines that are sometimes valued (as in the topological watershed [CB97b, NC03a] which conserves the saliency between minima) whereas ‘region-algorithms’ return labeled regions (the CBs) that form a partition of the image.

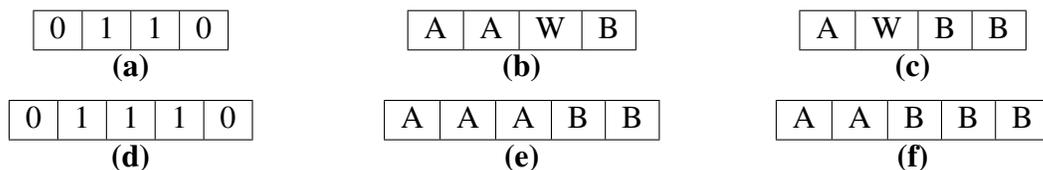


Fig. A.1: Original images (a)(d) and two possible labeled WS outputs (raster or anti-raster scan) of a line-algorithm (b)(c) or a region-algorithm (e)(f). W represents the WS line.

In lots of (line- or region-) algorithms, the result varies with implementations (scanning order and other arbitrary processing order) or may be inconsistent with the WS definition as observed

in [RM00]. This variation due to implementation can be insignificant in some cases (1 pixel bias for the line/region position, see Fig. A.1) but in other cases, it becomes considerable: in some images an entire region is reached passing by a *bottleneck* pixel and consequently included to the first (or other arbitrary) CB that invades the bottleneck (like in Fig. A.2(g)–(j)). Thus, the problem does not occur only on plateaux. Furthermore, it is of theoretical interest having a unique solution for the WS transform. These arguments encouraged the investigation of a WS definition that would result in a unique and consistent solution.

The Image Foresting Transform (IFT), introduced by Falcão, Lotufo and Stolfi [FSL04] and based on Dijkstra algorithm [Dij59], provides a sound framework for the efficient implementation of many image processing operators [FCL01]. For instance, the WS transform is computed as a problem of trees of minimal paths.

Figure A.2 shows a *buttonhole* case. It will be commented in both sections A.2 and A.3. Figure A.2(a) presents the input image used to compute the different watersheds, forests and maps of Figures A.2(b)–(j). It is important to remark that buttonhole cases “correspond to special pixel configurations which are not so rare in practice” as referred by [NC03a, VS91].

This paper is organized as follows. In section A.2, an overview of the IFT framework is given to define in this context the Tie-Zone Watershed (TZWS) that results in a unique solution, regardless of implementation. Then, the efficient algorithm introduced in [ALC05] is recalled. The bottlenecks, access of the tie-zones, are defined and characterized in section A.3. Finally, as an application of these new concepts, the TZ and bottleneck analysis of real images is done in section A.4 and associated to the robustness of the segmentations.

A.2 The Tie-Zone Watershed (TZWS)

A.2.1 Overview of the Image Foresting Transform (IFT)

Under the IFT framework, an image is seen as a weighted graph $G = (V, A, I)$ where each pixel (or voxel in 3D) is represented by a node or vertex $v \in V$ with intensity $I(v)$ (I is a map from V to \mathbb{Z} for digital image). An arc $\langle u, v \rangle \in A$ exists between vertices u and v when the corresponding pixels are adjacent according to the defined adjacency (usually 4- or 8-adjacency in 2D and 6- or 26-adjacency in 3D). A path from a node u to a node v in a graph (V, A, I) is a sequence $\langle u = v_1, v_2, \dots, v_n = v \rangle$ of nodes of V such that $\forall i = 1 \dots n - 1, \langle v_i, v_{i+1} \rangle \in A$. A path is said simple if all its nodes are different from each other. Let $S \subseteq V$ be a set of particular nodes s_i called seeds or markers.

For a given weighted graph $G = (V, A, I)$ and a set S of seeds, an IFT is a *directed forest* F of

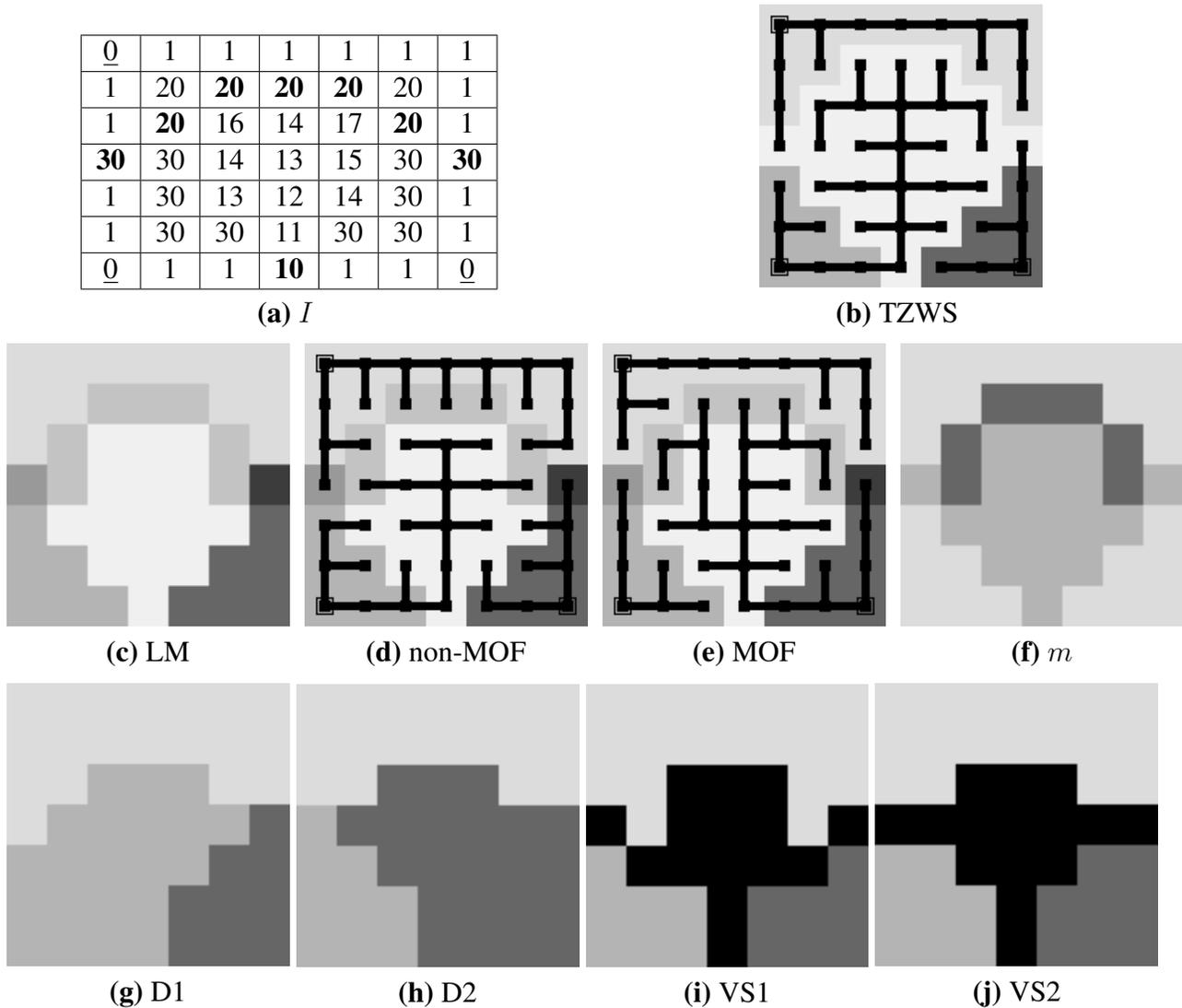


Fig. A.2: (a): Input grayscale image with 3 (underlined) minima and 8 (bold) bottlenecks. (b): TZWS using 4-adjacency: 3 CBs (gray), TZ (white), forest (black). (c)–(e): Result of the Label Merging algorithm obtained with either a non-MOF (d) or a MOF (e). (f): Map of multiplicity. (g)–(h): Watersheds by Dijkstra-IFT varying scanning-order. (i)–(j): Watersheds (black) by Vincent and Soille’s algorithm (raster or anti-raster scan).

G , i.e. a directed acyclic subgraph¹ F of G , such that (i) there exists for each node $v \in V$ a *unique and directed simple path* $\pi(s_i, v)$ in F from a seed node $s_i \in S$ to v and (ii) each such path has a *minimum cost* among all possible paths in G linking v to any seed of S , according a specified path cost function f_C .

Assume that the arcs $\langle u, v \rangle$ are weighted with the gray-level $I[v]$ of the pixel corresponding to v . Assume that the seed nodes correspond to the regional minima of the image (or to imposed minima, i.e. markers). If the path cost function is defined as the ‘maximum arc’ function f_{max} ,

$$f_{max}(\langle v_1, v_2, \dots, v_n \rangle) = \max \{h(v_1), I(v_2), \dots, I(v_n)\}$$

where h is a fixed but arbitrary *handicap cost* [LFZ02], the IFT computes a region-WS transform where each tree of the forest² corresponds to a CB. Note that all vertices (pixels) are covered by this forest. The IFT can result in many optimal forests because many paths of minimum cost are sometimes possible. The set of all optimal forests is denoted by Φ .

The optimality of the WS by IFT was proved in [LF00] where a two-component lexicographic cost function f_{LC} was proposed to mimic the flooding process and handle with plateaux too: $f_{LC} = (f_{max}, f_d)$. The first component, of highest priority, is the max-arc function and represents the flooding process. The second one makes different waters propagate on plateau at a same speed rate:

$$\begin{aligned} f_d(\langle v_1, v_2, \dots, v_n \rangle) &= \max_{k \in [0, n-1]} \{k, C[v_n] = C[v_{n-k}]\} \\ C[v_n] &= f_{max}(\langle v_1, v_2, \dots, v_n \rangle) \end{aligned}$$

This lexicographic path cost, inspired from Meyer’s topographical distance strategy [Mey94b], is very simple to compute using a priority FIFO queue, avoids a prior lower completion on image with plateaux, and provides partitions that seem to be more equitable (on plateaux) than when only the maximum cost is used.

A.2.2 The Tie-Zone Watershed (TZWS) transform

As we saw in the previous section, many optimal forests and so, many partitions may correspond to an input image-graph. We propose then a new definition of watershed transform in the IFT context which results in a unique partition.

A node is included in a specific catchment basin CB_i when it is linked by a path to a same seed

¹The graph $G' = (V', A')$ is a subgraph of G if $V' \subseteq V$, $A' \subseteq A$ and $A' \subseteq V' \times V'$.

²A tree of the forest F is a connected component of F .

s_i in all the optimal forests, otherwise it is included in the Tie-Zone T :

$$CB_i = \{v \in V, \forall F \in \Phi, \exists \pi(s_i, v) \text{ in } F\}$$

$$T = V \setminus \bigcup_i CB_i$$

If a node is in the tie-zone, it means that it could be included in different CBs without affecting the forest optimality. CBs are only the common part of all optimal solutions whereas differing parts are considered TZ. Therefore, the tie-zone existence prevents from making any arbitrary choice between optimal solutions. Consequently, the TZWS solution is defined without ambiguity.

Note that this definition does not produce watershed lines but only regions: catchment basins and tie zone. They form together a *unique* optimal partition of the image. If all pixels are assigned to catchment basins, the tie zone will be empty. This situation can occur when the lexicographic path-cost function unties growing CBs on plateaux. So, the watershed transform possibly does not contain any tie zone.

Unlike in the WS by IFT, each CB corresponds to a tree *or* part of it, while the TZ is composed of many terminal parts of trees as in the example of Fig. A.2(b).

A.2.3 Algorithm

In this section, we present an efficient algorithm that labels the image in order to obtain a TZWS. It is based on Dijkstra's shortest path algorithm [Dij59] and utilizes an ordered queue Q where each bucket has a FIFO policy. Note that the second component C_2 of the lexicographic cost is *not* intrinsically computed by the FIFO policy and must be explicit in the TZWS by IFT in order to prevent 1-pixel bias.

The algorithm input is: the image as a weighted graph $G = (V, A, I)$, the seed node set S with associated labeling function λ and handicap function h . The priority queue Q is initially empty: `DequeueMin` removes from Q and returns the node of minimum cost; `Enqueue(p, c)` inserts node p in Q at priority (cost) c bucket. We denote the neighborhood of a node $p \in V$ by: $N_G(p) = \{q \in V, \langle p, q \rangle \in A\}$. Label map L corresponds to the TZWS result, map P gives each node's predecessor in the tree and maps C_1, C_2 give the lexicographic cost of an optimal path from a seed to each node.

The beginning of the algorithm (lines 1 to 11) is identical with the IFT algorithm in [FSL04]. Lines 12 to 16 are TZWS-specific. In line 12, the second component of lexicographic cost is incremented, as water propagates on plateau. Lines 13 to 16 detect the nodes where paths from (at least) two seeds ($L[p] \neq L[v]$) tie, i.e. have same costs (C_1 and C_2).

Algorithm 1: TZWS by IFT with lexicographic path cost.

1. $\forall p \in V, C_2[p] \leftarrow 0; \quad done(p) \leftarrow \text{FALSE};$
2. $\forall p \notin S, C_1[p] \leftarrow \infty; \quad L[p] \leftarrow \text{NIL}; \quad P[p] \leftarrow \text{NIL};$
3. $\forall p \in S, C_1[p] \leftarrow h(p); L[p] \leftarrow \lambda(p); \quad P[p] \leftarrow p; \quad \text{Enqueue}(p, h(p));$
4. **while** QueueNotEmpty,
5. $v \leftarrow \text{DequeueMin}; done(v) \leftarrow \text{TRUE};$
6. $\forall p \in N_G(v)$ **and** $done(p) = \text{FALSE},$
7. $c \leftarrow \max\{C_1[v], I[p]\};$
8. **if** $c < C_1[p],$
9. **if** p in $Q, \text{Dequeue}(p);$
10. $C_1[p] \leftarrow c; L[p] \leftarrow L[v]; P[p] \leftarrow v;$
11. $\text{Enqueue}(p, C_1[p]);$
12. **if** $c = C_1[v], C_2[p] \leftarrow C_2[v] + 1;$
13. **else, if** $c = C_1[p]$ **and** $L[p] \neq L[v],$
14. **if** $c = C_1[v],$
15. **if** $C_2[p] = C_2[v] + 1, L[p] \leftarrow \text{TZ};$
16. **else** $L[p] \leftarrow \text{TZ};$

This algorithm is fast and has the same speed performance as the IFT-WS [FSL04]. The solution of TZWS is optimal because it is based on IFT, it keeps therefore the optimality of the shortest-path forest solution as demonstrated in [LF00, FSL04]. Besides, the other algorithms generally depend on arbitrary decisions in processing order (which pixel is removed first from a priority bucket of the queue?) that are not in the strict definition of WS and that introduce bias (see the different solutions of Fig. A.2(g)–(j)). Observe that the bias problem does not occur only on plateaux and may be unacceptable for some applications (e.g. precise measures on segmented objects).

The TZWS can also be obtained without using an ordered queue by processing the image data in raster-scan and anti raster-scan order alternatively until stability of the result (algorithm not presented here).

A.3 Bottlenecks

A.3.1 Watershed vs. watermerging

As said in section A.1, the watershed (WS) transform is compared to the flooding of a topography where dams are built to prevent distinct colored waters from merging (suppose that a color is assigned

to each marker/minimum). Now, let us illustrate the watermerging paradigm. For an intuitive comprehension, put the topography (representing the image) up-to-down (see an illustration in Fig.A.3). Imagine that each marker (former minimum that is now regional maximum) is a source of colored water. When colored waters meet together, no dam is built but the colored waters naturally merge into a water of blended color. Holes are punched in the regional minima (former maxima) for draining the water. Supposing that (abundant) waters propagate along all negative slopes (not only the steepest as occurs in reality), we get colored hills and possibly regions with blended colors. These regions correspond to the (multicolor) tie-zone. The up-to-down transformation is not used in practical implementations but only for an intuitive explanation of the watermerging paradigm. Watermerging is different from the *drop of water principle* or *rainfalling* algorithm [RM00].

As we defined earlier, the tie-zone is the region where pixels cannot be assigned to a specific label, i.e. there is no unique label that could be assigned to them. We can differentiate the tie-zone in labeled (or colored) tie-zones. A labeled tie-zone belongs necessarily to the tie-zone and assumes a specific tie-zone label: this label contains the information of which labels could be potentially assigned to the labeled tie-zone. In other words, the blended color of a colored tie-zone contains all the pigments of original colored waters that merged together. For example, when label a and label b tie together at a region, this region becomes a labeled tie-zone and will assume a *merged-label* $\{a, b\}$. We say that the original labels a and b are included in the merged-label $\{a, b\}$: $a \in \{a, b\}$; $b \in \{a, b\}$.

The Label Merging (LM) algorithm is a useful variation of the previous algorithm 1. When waters from different minima are merging, it assigns a new blended label to the region invaded by these waters. Substitute in algorithm 1 TZ label (lines 15-16) by `MergeLabels($\mathbb{L}[p]$, $\mathbb{L}[v]$)` and the simple label map L by a merged-label map \mathbb{L} . So, the final labeled image allows a traceability on tie zones: it informs exactly which and how many colored waters of different labels tied together at each node. By analyzing the pigment mix, one can deduce which original pigments are included in. There is no more one TZ label but so many as the distinct label mergings (4 in the example of Fig. A.2(c)).

A.3.2 Multiplicity-based Optimal Forests (MOF)

We define the *multiplicity* $m(p)$ of a pixel p as the number of original labels $\lambda_i \in \Lambda$ that could be assigned to it, i.e. the number of labels that tied together. Clearly, a pixel in a catchment basin has always a multiplicity equal to one and a pixel in a tie-zone has necessarily multiplicity greater than one (see Fig.A.2(f)). Formally, the labeling function λ and the multiplicity m are defined as follows:

$$\begin{aligned} \lambda : S &\rightarrow \Lambda \\ s &\mapsto \lambda(s) \end{aligned}$$

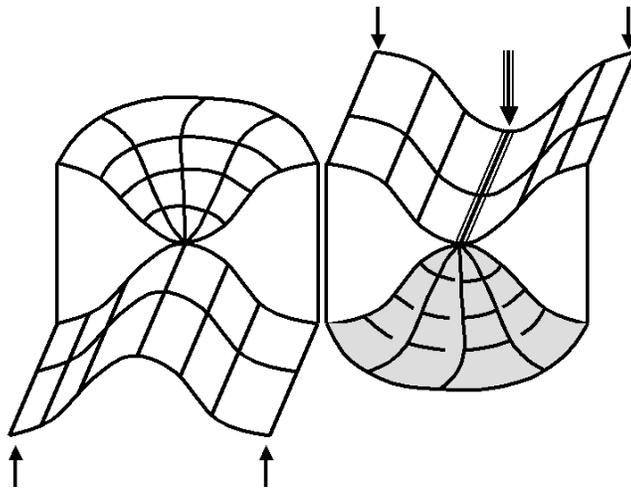


Fig. A.3: Watermerging paradigm. On the left (right), the arrows point the 2 minima (maxima) of the topography. A triple line (of constant altitude) shows the bottlenecks locus and constitutes with the gray zone the tie-zone where waters merged. Only the front bottleneck in contact with the gray zone (its influence zone) is non-trivial.

$$\begin{aligned}
 m : V &\rightarrow \mathbb{N} \\
 p &\mapsto m(p) = \text{CARD}\{\mathbb{L}[p]\}
 \end{aligned}$$

And, assuming that Λ is the set of original labels and Λ_I the set of all possible subsets of Λ , the label map \mathbb{L} is now:

$$\begin{aligned}
 \mathbb{L} : V &\rightarrow \Lambda_I \\
 p &\mapsto \mathbb{L}[p] = \{\lambda_1, \lambda_2, \dots, \lambda_{m(p)}\}
 \end{aligned}$$

As we saw in sections A.2.1 and A.2.2, there can be many optimal forests corresponding to an image. In other words, there can be many possible predecessors for some nodes of the image-graph. We distinguish a special type of optimal forests: the Multiplicity-based Optimal Forests (MOF). In the set of possible optimal forests, the MOFs are those where each node points to the predecessor (father) of highest multiplicity among all possible predecessors. Note that there can exist many MOFs associated to a same image-graph. Figure A.2 shows examples of a non-MOF (d) and two MOFs (b) and (e).

A.3.3 Bottleneck characterization

The bottlenecks are particular pixels that play an important role in the comprehension of the image and its partition in catchment basins or in tie-zone. Indeed, they are the points from which tie-zones appear.

Intuitively, they represent a unique and thin access ('bottleneck') for the water to enter and invade another region ('bottle'). More generally, in our context, they are the first contacts between two (or more) wave fronts of different colored merged-labels (and whose merging will result in another merged-label). Once the wave fronts have merged, the resulting wave front can, in some cases, invade a region (the 'bottle' of the bottleneck). Clearly, a bottleneck is never in a CB as it is a merging point.

In the watermerging paradigm (LM), *bottlenecks* are defined as nodes (pixels) whose merged-label is different from the predecessor's merged-label, when considering any particular MOF of the image-graph. For example, the bottlenecks of the image of Fig. A.2(a) (in bold) can be detected from either MOF in (b) or (e) using the definition:

$$\mathbb{L}[p] \neq \mathbb{L}[P[p]] \Leftrightarrow p \text{ is bottleneck} \quad (\text{A.1})$$

Let us consider now the region whose access is permitted by the bottleneck: the influence zone. If all possible MOFs are considered, the *extended influence zone* $IZ_e(b)$ of a bottleneck b is the set of all possible descendants (direct and indirect children) of the bottleneck and itself. The number of nodes of a bottleneck's influence zone is called extended weight w_e of the bottleneck. $w_e(b) = \text{CARD}\{IZ_e(b)\}$. When considering only the descendant nodes with same merged-label as the bottleneck's one, we simply refer to the *influence zone* $IZ(b)$ of bottleneck b . And the number of nodes within is defined as the *weight* w of the bottleneck:

$$w(b) = \text{CARD}\{IZ(b)\}.$$

A bottleneck is *trivial* if and only if its weight is one. When the bottleneck has weight strictly greater than one, it is said *non-trivial* (see Fig. A.3).

An adapted version of the LM algorithm allows to identify the bottlenecks on-the-fly during the construction of the forest. This Bottleneck Identification algorithm processes the label merging like in the LM algorithm and also updates the predecessor map according to the multiplicity criterion even if it has no influence on the cost and label maps. Thus, a particular MOF is obtained. When a node is removed from the queue (see algorithm 1), its definitive label is compared to the label of its predecessor to decide whether or not it is a bottleneck according to relation (A.1).

A.4 On the robustness of a segmentation

Now the bottleneck (BN) concept has been defined and related to the tie-zone (TZ) concept, we will show that the analysis of both TZs and BNs can be associated with the robustness of a segmentation. In this paper we call segmentation's *robustness* the impartiality or stability associated to the result of a segmentation process utilizing the same inputs and method. In our case, if different WS algorithms and implementations are used for segmenting a given image with given parameters (like adjacency and seeds), result's variations will affect the robustness (reliability).

Observe that a robust segmentation is not necessarily a 'good' segmentation in a semantic point of view but it is constant in relation to arbitrary choices of implementation. Moreover, one can be sure that a non-robust segmentation is only a possible but unreliable segmentation among numerous other ones. In other words, a non-robust segmentation is biased by the implemented algorithm as the image, seeds and segmentation process should not define thin WS but a large TZ.

A.4.1 Robustness based on tie-zone analysis

First, the area of the TZ can be a robustness measure. Larger the TZ is, less robust the segmentation is. For example, Fig.A.4(a) shows WS lines obtained from a non-filtered airplane image gradient by the Vincent and Soille's algorithm. Clearly, anyone can observe an oversegmentation as the *airplane* and the buildings are the expected objects of interest. This result is not only 'bad' in a semantic point of view but also very partial. The large and numerous tie-zones of the TZWS in Fig.A.4(d) point out the high uncertainty that affects any thin solution. In Fig.A.4(b) and (c), a filtering was applied on gradient's minima : basins (minima) with height smaller than $h = 10$ and $h = 35$ were respectively removed. The segmentation is much better in both semantic and robustness points of view (Fig. A.4(e)(f)). It is necessary to normalize the TZ area with the image size to have an objective and size-independent robustness measure. The first (non)robustness measure is consequently:

$$R_1 = \frac{CARD\{T\}}{CARD\{V\}}.$$

We can also observe that in many cases as in Fig.A.4(f), the TZ is reduced to a thin WS line contouring the object of interest. As the output of the segmentation can be separating lines, we may consider that the TZ area of this line does not represent a lack of robustness. Furthermore, when a lot of objects of interest are expected (cells separation for example), the previous measure R_1 will bias the reality. We cannot consider the lines contouring the objects as a real handicap for robustness. Thus, we apply an erosion ϵ on the TZ to eliminate the contouring lines and, thereby, not to have a measure depending on the number of objects. The *residue* R of this erosion represents now the

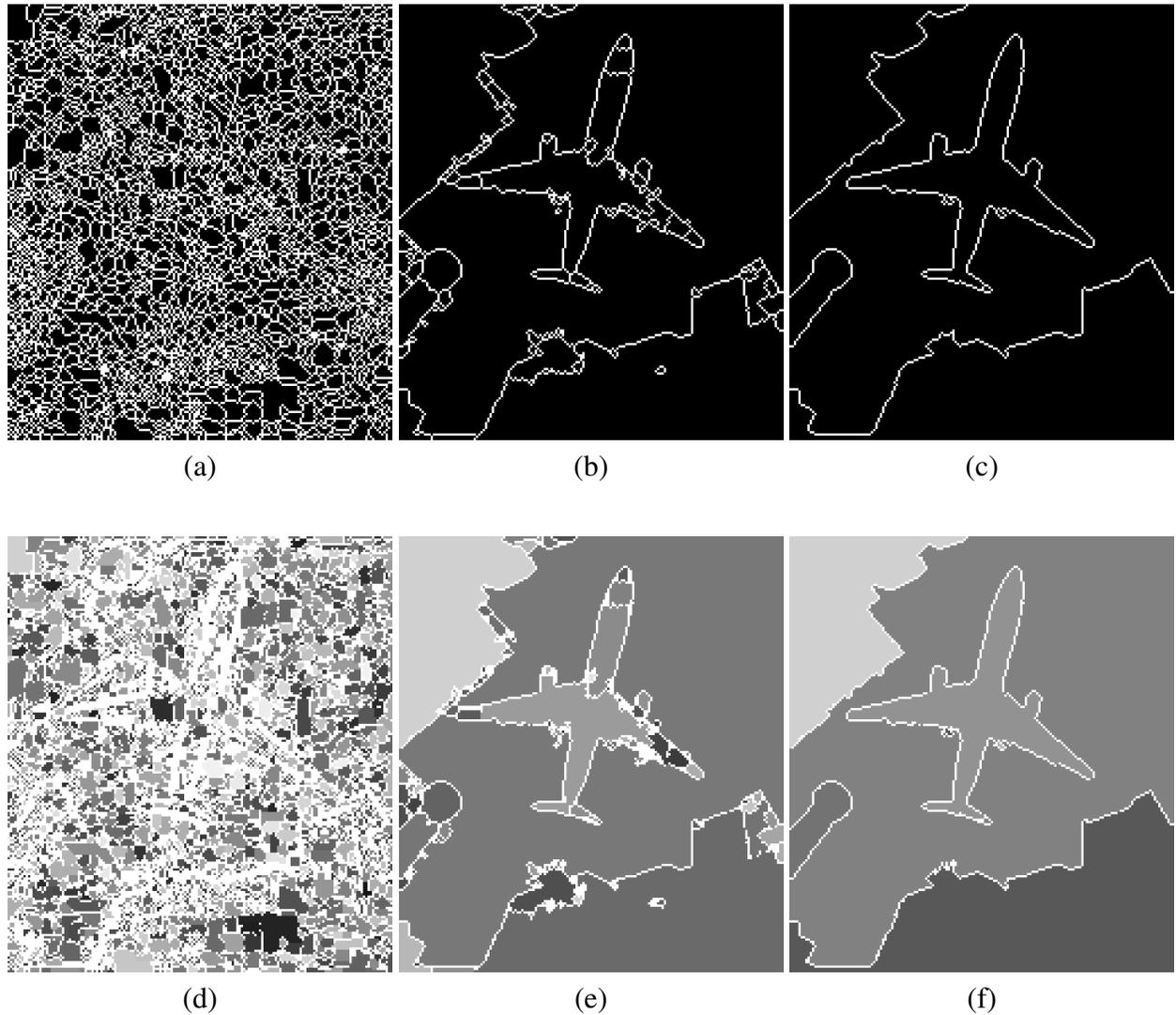


Fig. A.4: *Airplane* image: Watershed lines (by Vincent-Soille's algorithm) of the gradient (a) and the filtered gradient (b)(c) (with $h = 10; 35$ resp.). (d)–(f) Respective TZ in white.

thickness of the TZ without depending on the number of objects. Therefore, a second objective measure of (non)robustness is the normalized area of the residue:

$$R_2 = \frac{CARD\{\epsilon(T)\}}{CARD\{V\}} = \frac{CARD\{R\}}{CARD\{V\}} \quad (\text{A.2})$$

For an application, one may consider that R_2 must be less than 1% to ensure a robust segmentation. After processing the TZWS for an image and its filtered versions, the variation of R_2 is computed and segmentations with the required robustness are automatically selected. Figure A.5 shows this variation for 8 images and the 1%-threshold. Observe that the artificial *weavetile* segmentation is very robust for $h \geq 0$ whereas natural *mirage* gradient image (Fig. A.6) becomes robust for $h \geq 6$ and *leaf-grass* for $h \geq 37$ (Fig. A.7). This is due to the grass background and natural light and shade effects. Clearly, when the image is too much filtered, the number of objects decreases and can be less than the expected number (no object at worst!): the image is undersegmented. Thus, there is a compromise between having a robust segmentation and a semantically good segmentation. If this compromise is impossible, another segmentation strategy (filtering, marker choice, etc.) must be applied. Semantic criteria together with R_2 can help the user to choose a reasonable filtering.

The pseudo-cone of Fig. A.8 is an interesting case of very large TZ. Any thin WS solution is totally unreliable and inconsistent because many lines have to converge to a same central pixel by passing through same pixels when closer to the top. This phenomenon is due to the discretization of the space and the non-null thickness of the lines. Observe that only the TZWS (b) has a 4-axis-symmetry like the input image (a). The other algorithms (c)(d) are biased by the processing order.

A.4.2 Robustness based on bottlenecks analysis

In section A.3, we saw that bottlenecks are the roots of the patches that constitute the TZ. They are thereby the origin of the segmentation's non-robustness. The bottlenecks analysis allows to know how the sources of partiality are distributed in the segmented image and how large their influence zones are. Figure A.9 shows for example a detail of the non-filtered *mirage* gradient image segmented by Label Merging (LM) algorithm. CBs (solid gray) meet and bottlenecks (BN) appear (white contour). Some of them have a large influence zone (IZ). Others have a little one or are trivial.

We compute therefore the histogram H of the BNs according to their weight w . It informs which types of BNs are most present in the segmented image. For example, for a same R_2 or TZ area, there can be a lot of trivial bottlenecks in the entire image which proves that the potential bias is not concentrated in a particular region; or maybe, there can be few BNs with important weight. It depends on the essence of the image. This BNs distribution by weight can be a useful information for the choice of filtering strategy when the user attempts to increase the segmentation's robustness.

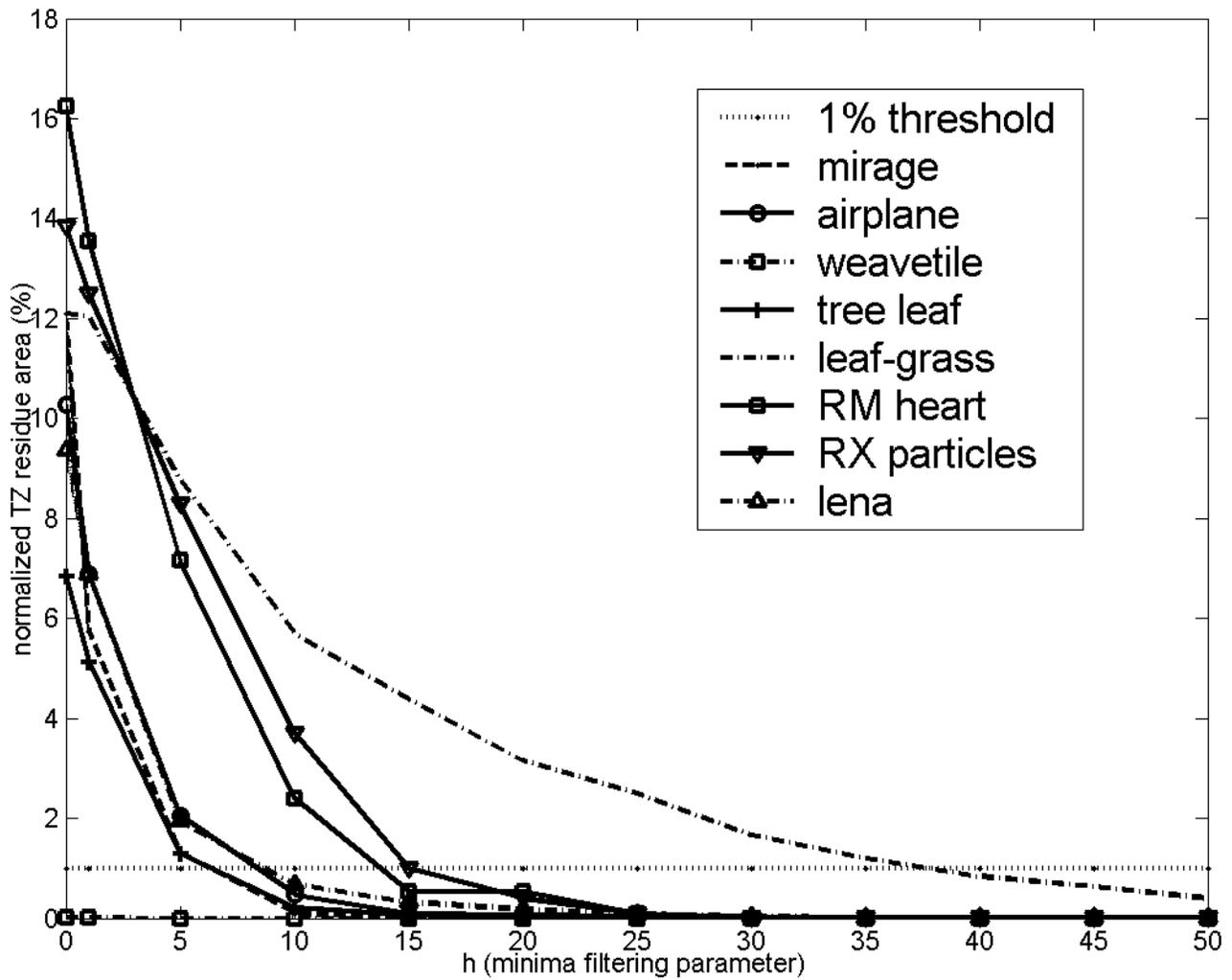


Fig. A.5: R_2 in function of h for *weavetile*, *tree-leaf*, *mirage*, *airplane*, *lena*, *MR heart*, *XR particles* and *leaf-grass* images. $R_2 < 1\%$ for $h \geq 0, 3, 3, 8, 8, 14, 15, 37$ respectively.

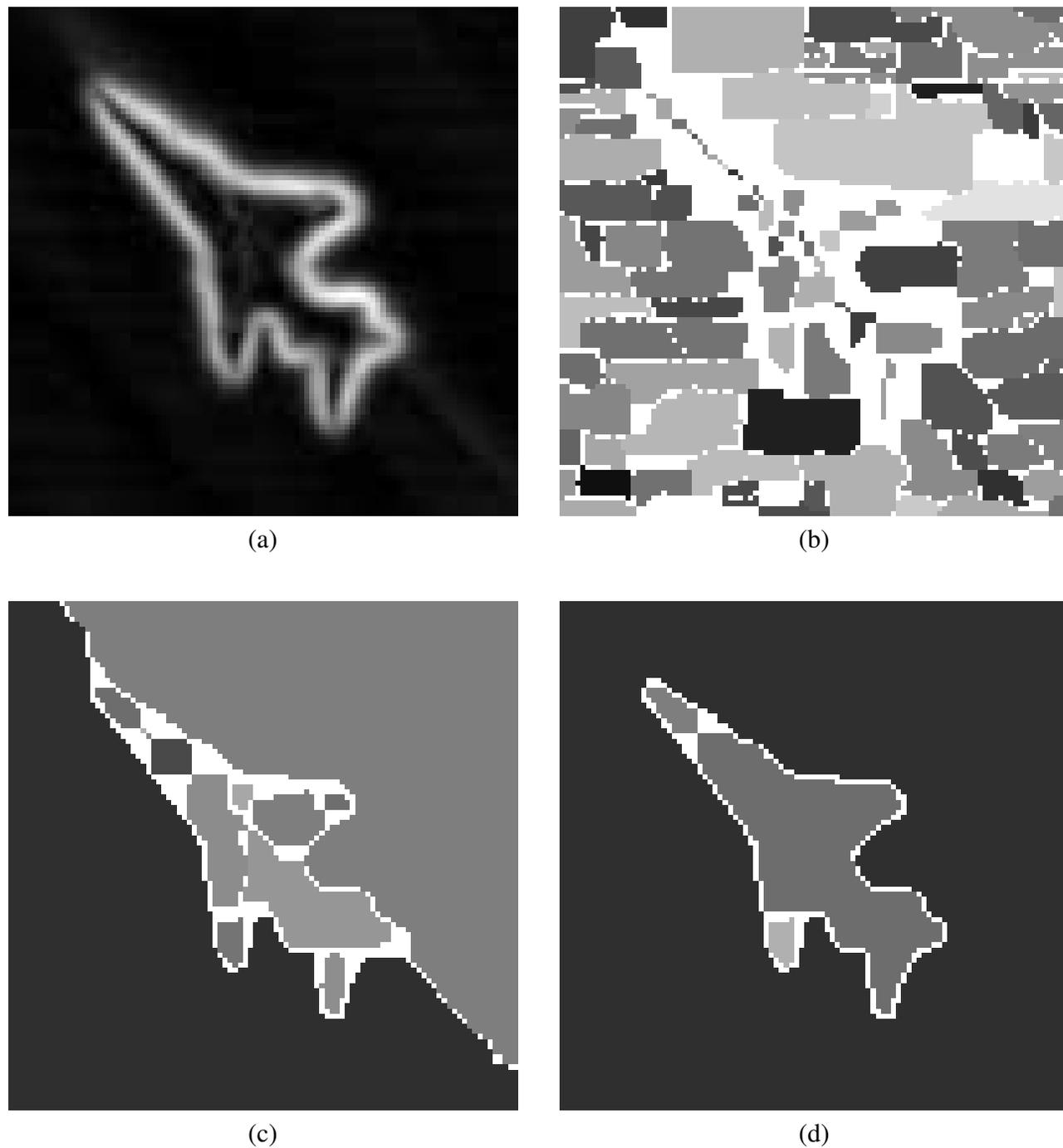


Fig. A.6: (a) Input *mirage* gradient image. (b)–(d) TZWS (white) after filtering the gradient minima with $h = 0; 5; 10$ (116, 13, 4 min.) resp.

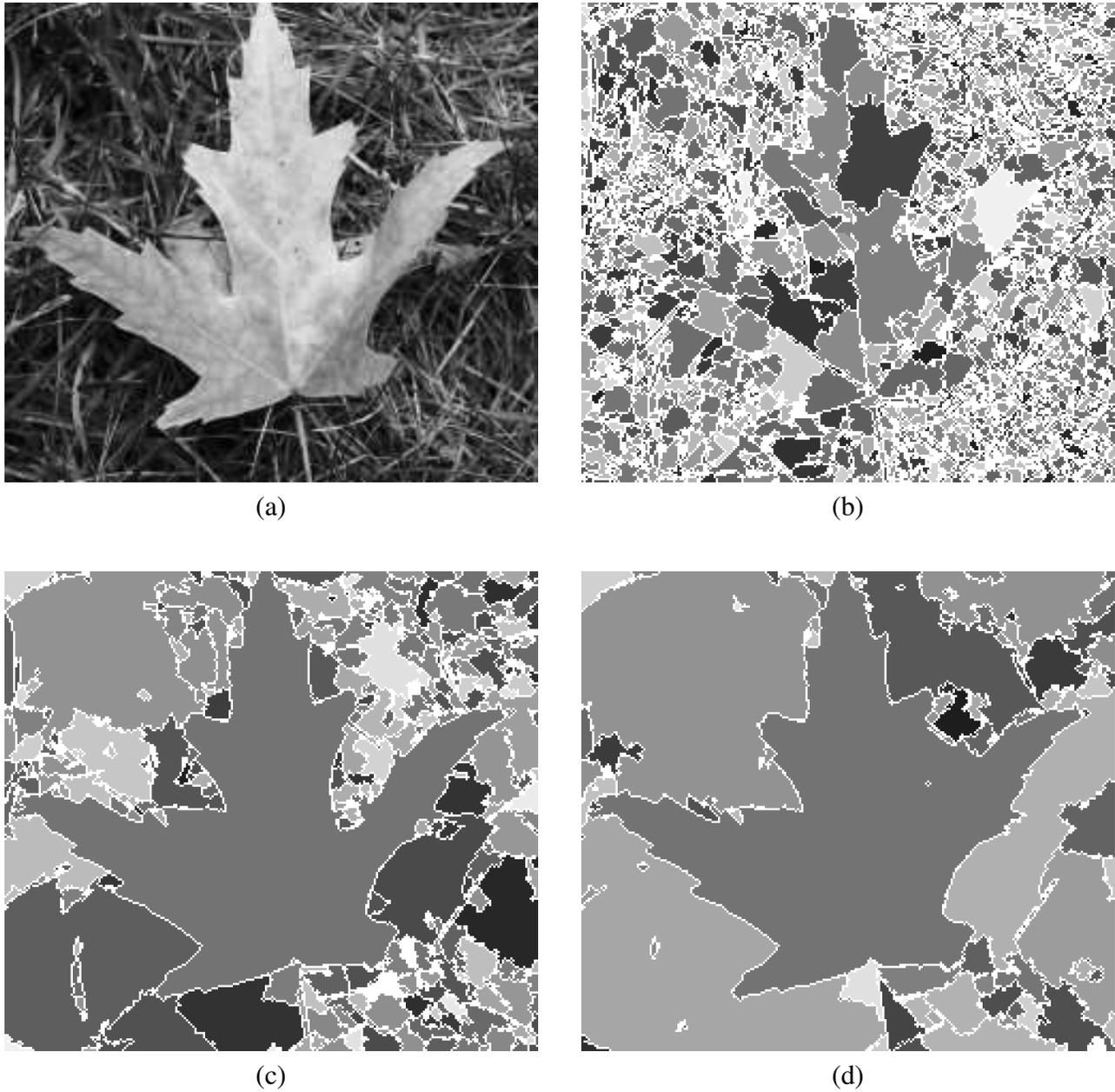
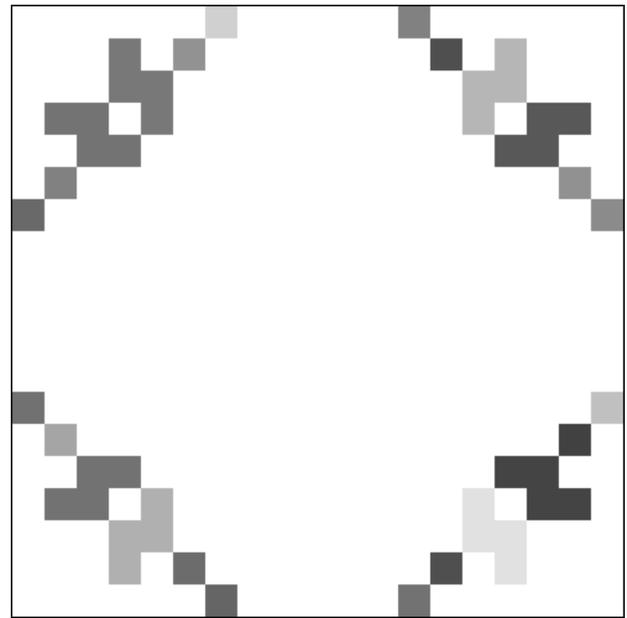


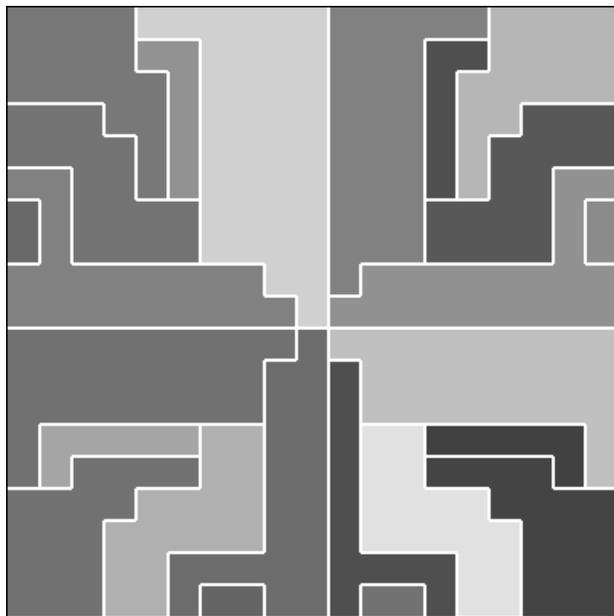
Fig. A.7: (a) Original *leaf-grass* image. (b)–(d) TZWS (white) after filtering the gradient minima with $h = 10; 35; 50$ (2936, 411, 123 min.) resp.

4	3	2	2	1	1	0	2	2	2	2	2	0	1	1	2	2	3	4
3	2	2	1	1	0	1	1	1	1	1	1	1	0	1	1	2	2	3
2	2	1	0	0	1	1	2	2	2	2	2	1	1	0	0	1	2	2
2	1	0	1	1	2	2	3	3	3	3	3	2	2	1	1	0	1	2
1	1	0	1	2	3	3	4	4	4	4	4	3	3	2	1	0	1	1
1	0	1	2	3	3	4	5	5	5	5	5	4	3	3	2	1	0	1
0	1	1	2	3	4	5	5	6	6	6	5	5	4	3	2	1	1	0
2	1	2	3	4	5	5	6	7	7	7	6	5	5	4	3	2	1	2
2	1	2	3	4	5	6	7	8	8	8	7	6	5	4	3	2	1	2
2	1	2	3	4	5	6	7	8	8	8	7	6	5	4	3	2	1	2
2	1	2	3	4	5	6	7	8	9	8	7	6	5	4	3	2	1	2
2	1	2	3	4	5	6	7	8	8	8	7	6	5	4	3	2	1	2
2	1	2	3	4	5	5	6	7	7	7	6	5	5	4	3	2	1	2
0	1	1	2	3	4	5	5	6	6	6	5	5	4	3	2	1	1	0
1	0	1	2	3	3	4	5	5	5	5	5	4	3	3	2	1	0	1
1	1	0	1	2	3	3	4	4	4	4	4	3	3	2	1	0	1	1
2	1	0	1	1	2	2	3	3	3	3	3	2	2	1	1	0	1	2
2	2	1	0	0	1	1	2	2	2	2	2	1	1	0	0	1	2	2
3	2	2	1	1	0	1	1	1	1	1	1	1	0	1	1	2	2	3
4	3	2	2	1	1	0	2	2	2	2	2	0	1	1	2	2	3	4

(a) *I*



(b) TZWS



(c) D



(d) VS

Fig. A.8: Pseudo-cone. (a) Input image with 24 minima at 0-level (4-adjacency) and 101 bottlenecks (in bold). (b) TZWS (white). (c)–(d) WS by Dijkstra’s and Vincent-Soille’s algorithms (raster-scan order).



Fig. A.9: LM algorithm on *mirage* (detail): bottlenecks (white contour) and IZs (shaded).

More useful is to know what contribution each type of BNs has. To this end, the weighted histogram H_w is computed. For each weight w corresponds the sum of the weights (IZ areas) of all BN with weight w . So we have the contribution of each BN-type to the total TZ area. Look for example at the cumulated weighted histogram H_w^c in Fig. A.10(a) for the non-filtered *mirage* case. We can see that the TZ area is greater than 2500 and there exists BN with $w \approx 120$ or 80 and nearly 500 trivial BNs. To have more compact informations, the quartiles are calculated (Fig. A.10(b)). For the *mirage* image segmentation without filtering, the non-robustness is due to BNs with weight $w \leq 2$ (for 25%), BNs with $2 < w \leq 9$ (for other 25%), BNs with $9 < w \leq 32$ (for 25% too) and BNs with $32 < w \leq 123$ (for the last 25%). Interestingly, for the *leaf-grass* image, the filtering modifies this distribution of contribution in an unexpected way. It creates BNs with larger IZs. Note that the total TZ area increases with this filtering too. Be careful that it does not appear in the quartiles diagrams and that the ranges of BN weights contribute *equally* to the TZ area.

In conclusion, the bottlenecks analysis gives information on how a segmentation can be biased whereas the TZ analysis does on how much it can be.

A.5 Conclusions and future work

In this work, we recalled the tie-zone watershed (TZWS) transform and, for understanding the TZ's essence, defined a new concept based on the watermerging paradigm: the bottleneck (BN). Each BN is responsible for a part of the TZ and, therefore, part of the possible bias of a WS transform. Theoretical examples and real images were shown to demonstrate that some WS segmentations have no sense when the TZ is too large: they are not representative. Furthermore, the robustness of a segmentation can be characterized quantitatively by the (eroded) TZ's area and qualitatively by the BN's distribution and contribution.

In future work, we will show the central role of BNs in a thinning procedure of the TZWS for segmentation purpose. We also intend to investigate the relation of the bottlenecks with pass-value and saliency concepts used by the topological watershed.

Acknowledgments

This work is supported by CAPES.

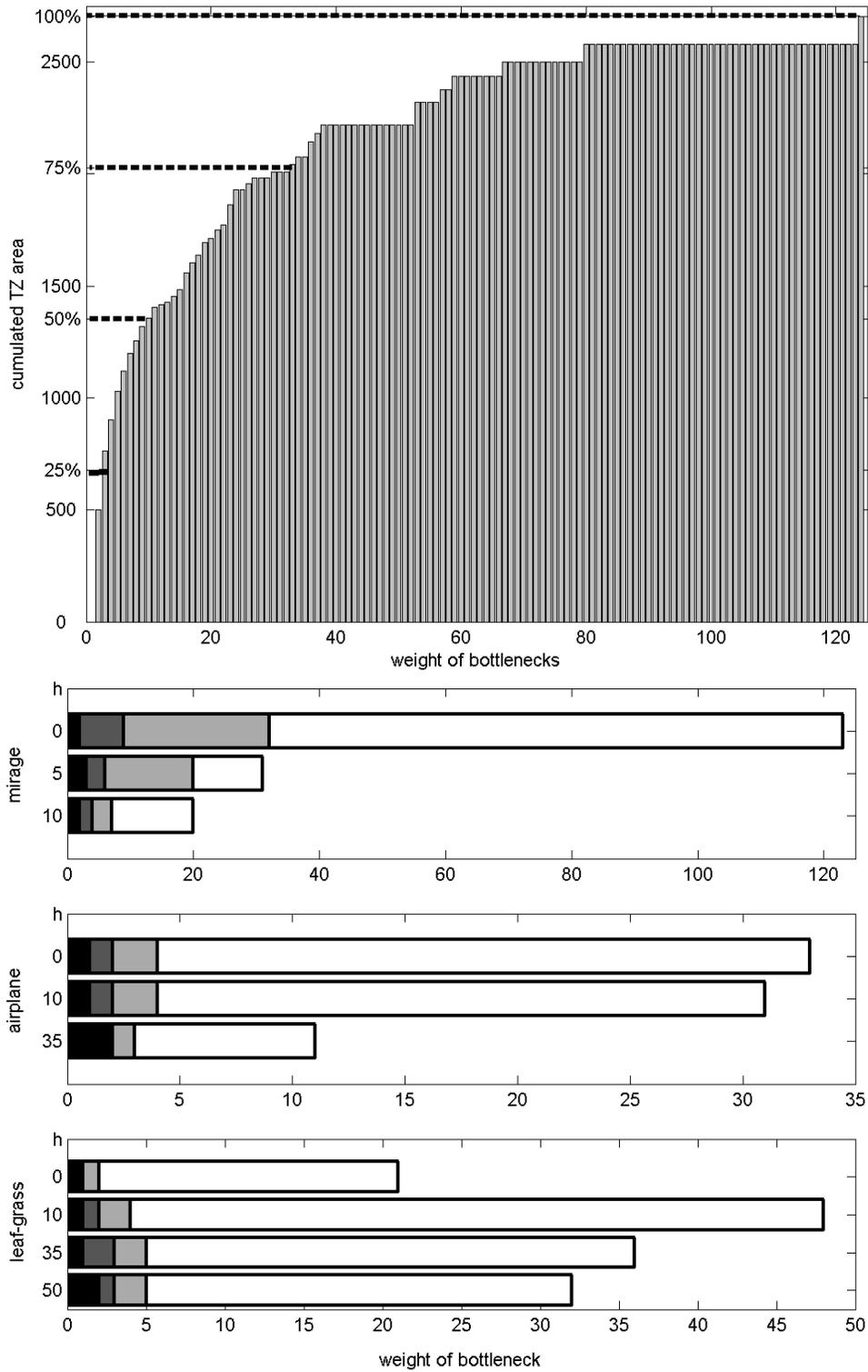


Fig. A.10: (a) Cumulated weighted histogram. (b) Quartiles for 3 segmented images varying h : w-ranges of the BNs that contribute for each quarter of the total TZ area.