



HABILITATION A DIRIGER DES RECHERCHES

en Informatique

Année Universitaire : 2020-2021

présentée et soutenue publiquement par

Julien MILLE

le 1er juillet 2021

Segmentation et classification d'objets

Object segmentation and classification

JURY (par ordre alphabétique) :

Mme. Raphaëlle CHAINE	Professeure des Universités	Université Claude Bernard Lyon 1
M. Thierry CHATEAU	Professeur des Universités	Université Clermont-Auvergne
Mme. Julie DELON	Professeure des Universités	Université de Paris
M. Jacques-Olivier LACHAUD	Professeur des Universités	Université Savoie Mont Blanc
M. Olivier LEZORAY	Professeur des Universités	Université de Caen-Normandie
M. Nicolas RAGOT	Maître de conférences, HDR	Université de Tours

Acknowledgements

Many thanks to :

- Thierry Chateau, Jacques-Olivier Lachaud, and Olivier Lezoray for reviewing this habilitation report.
- Raphaëlle Chaine and Julie Delon for her participation in the committee.
- Team-mate and advisor Nicolas Ragot for thorough reading, and many thoughtful comments.
- Former or current lab-mates for fruitful scientific interactions, by alphabetical order (hoping not to forget anyone): Nicolas Bonneel, Sébastien Bougleux, Laurent Cohen, Julie Digne, Moncef Hidane, Gabriel Peyré, Laure Tougne, Hugo Raguet, Romain Raveaux, Jean-Loïc Rose, Christian Wolf.
- other former or current colleagues at INSA Centre Val de Loire, Université de Tours, Université Lyon 1, Université Lyon 2, INSA Lyon (this set intersects the set of above-cited people)
- PhD students Imtiaz Ali, Aurélie Leborgne, Fabien Baradel and Luong Phat Nguyen.
- my kids Thomas and Manon, who will not get this at the time of writing. I hope you will cope well enough in English to read this sentence one time or another.
- Pauline, for improving my life!

Contents

1 Detailed résumé	6
1.1 Summary	6
1.2 Career history	6
1.3 Teaching activities	7
1.4 Summary of research activities	9
1.5 List of publications	14
2 Résumé en français	18
2.1 Introduction	18
2.2 L'objet pré-identifié: segmentation à l'aide de chemins	20
2.3 L'objet en mouvement : segmentation vidéo basée sur les modèles de fond	24
2.4 L'objet en entrée : appariement de formes basé sur le squelette	31
3 General introduction	38
4 The object as an identified target: segmentation with paths	44
4.1 Segmentation with prior interaction	44
4.2 Plane curves	45
4.3 Curves as minimizers of energies	46
4.4 Energies	49
4.5 Minimal paths	52
4.6 Building closed contours with minimal paths	55
4.7 The CDP method: combination of disjoint paths	57
4.8 The CSPP method: combination of saddle point-based paths	62
4.9 Combination of admissible paths using the simplicity energy	70
4.10 Experiments and discussion	71
4.11 Conclusion	77
5 Objects as moving entities: video object segmentation with background modeling	81
5.1 Video object segmentation	82
5.2 Non-parametric color density-based VOS	86
5.3 Segmentation as a Bayesian estimation problem	91
5.4 Short-sight modeling of background	93
5.5 Space-time spectral model for object detection in dynamic textured backgrounds	102
5.6 Conclusion	112
6 The object as a given input: skeleton-based shape matching	116
6.1 Ridge detection in Euclidean distance maps	117
6.2 Theoretical values for AOF and ridgeness	123
6.3 Matching with skeletons	143
6.4 Building a weighted skeleton	145

6.5	Matching hierarchical skeletons	158
6.6	Conclusion	176
7	What next?	179
7.1	Looking back over my shoulder	179
7.2	Long-term perspective	180
A	Minimal paths	183
A.1	Mathematical derivations of overlap and exteriory terms	183
A.2	Implementation details	185
B	Video object segmentation	194
B.1	Derivations involving divergence and entropy	194
C	Mathematical derivations on Euclidean distance-based skeletons	196
C.1	Derivations related to AOF and ridgeness	196
C.2	Deformable skeleton: calculus of variations with variable endpoints	218

Chapter 1

Detailed résumé

In this chapter, French designations are in *italics*. The following English/*French* translations are used:

- Associate professor: *Maître de conférences*
- Assistant professor: *Attaché Temporaire d'Education et de Recherche (ATER)*
- Supply teacher: *Vacataire*
- Referee: *Examinateur*
- Reviewer: *Rapporteur*

1.1 Summary

Julien MILLE

Associate professor in computer science

39 years old, divorced, 2 children

Mail: julien.mille@insa-cvl.fr

Webpage: <https://julien-mille.github.io>

1.2 Career history

2015-	Associate professor in computer science <u>Teaching institution</u> : <i>Institut National des Sciences Appliquées (INSA) Centre-Val de Loire</i> <u>Research institution</u> : <i>Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LI-FAT)</i> , formerly <i>Laboratoire d'Informatique (LI) de l'Université de Tours (EA 6300)</i>
2009-2015	Associate professor in computer science <u>Teaching institution</u> : <i>Université Claude Bernard Lyon 1, Faculté des Sciences et Technologies, Département Informatique</i> <u>Research Institution</u> : <i>Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS, UMR CNRS 5205)</i>
2008-2009	Postdoc position <i>Centre de Recherche en Mathématiques de la Décision (CEREMADE, UMR CNRS 7534), Université Paris Dauphine</i>

2007-2008	Assistant professor <u>Teaching institution</u> : <i>Ecole Polytechnique de l'Université de Tours (Polytech'Tours), Département Informatique (EPU-DI)</i> <u>Research Institution</u> : <i>Laboratoire d'Informatique (LI) de l'Université de Tours (EA 2101)</i>
2004-2007	PhD student Subject : <i>Modèles déformables pour la segmentation et le suivi en imagerie 2D et 3D</i> (Deformable models for segmentation and tracking in 2D and 3D imaging) defended the 3rd of December 2007 <i>Université François Rabelais de Tours</i> <i>Laboratoire Informatique (EA 2101)</i> Supervisors: Romuald BONE, Hubert CARDOT, Pascal MAKRIS <u>Committee</u> : Isabelle BLOCH, reviewer, <i>Telecom'Paris</i> Romuald BONE, supervisor, <i>Université de Tours</i> Hubert CARDOT, supervisor, <i>Université de Tours</i> Laurent COHEN, referee, <i>Université Paris Dauphine</i> Pascal MAKRIS, supervisor, <i>Université de Tours</i> Nikos PARAGIOS, reviewer, <i>Ecole Centrale de Paris</i> Marinette REVENU, referee, <i>ENSI Caen</i>

1.3 Teaching activities

The table below summarizes the amount of teaching activities, in terms of levels and time. Teaching from 2004 to 2008 was done as a supply teacher and assistant professor at the University of Tours. Appointed as an associate professor in 2009, I taught in the computer science department of the University of Lyon until 2015. Since then, teaching was done at *Institut National des Sciences Appliquées - Centre Val de Loire*, in Blois. The total time per year is expressed in tutorial-equivalent time (1h of lecture amounts to 1.5h of tutorial).

INSA-CVL = *Institut National des Sciences Appliquées - Centre Val de Loire*

UCBL = *Université Claude Bernard Lyon 1*

EPU-DI = *Ecole Polytechnique de l'Université de Tours, Département Informatique*

ENIVL = *Ecole Nationale d'Ingénieurs du Val de Loire*

L1-L3 = Bachelor of Science (BsC), year 1-3 (french *Licence*)

M1-M2 = Master, year 1 and 2

ENG = Engineering school

Year	Status	Institution	Level(s)	Total time
2020-2021	Assoc. Pr.	INSA-CVL	ENG (1st to 5th year)	220
2019-2020	"	"	"	280
2018-2019	"	"	"	280
2017-2018	"	"	"	260
2016-2017	"	"	"	230
2015-2016	"	"	ENG (1st and 3rd year)	225
2014-2015	"	UCBL	L1, L2, L3, M1, M2	280
2013-2014	"	"	L2, L3, M1, M2	232
2012-2013	"	"	L1, L2, L3, M1, M2	243
2011-2012	"	"	L2, L3	242
2010-2011	"	"	"	231
2009-2010	"	"	L1, L2	151
2007-2008	Assist. Pr.	EPU-DI	ENG (3st and 4th year)	102
2006-2007	Supply teacher	"	"	90
2005-2006	"	"	"	60
2004-2005	"	Univ. Tours / ENIVL	L1, ENG and 4th year)	67

1.3.1 Details on main teaching activities

Algorithmics and programming

Level(s): engineering school (1st and 2nd years), BsC (1st to 3rd year)

Institution(s): UCBL, INSA-CVL

Keywords: basic algorithmics, C programming language, functions, data structures, complexity.

In this paragraph, I gather several courses for students ranging from 1st year to 3rd after high school graduation (both BsC students and first cycle of engineering school). This represents, by far, the major part of my time spent in front of students. During the first year (or at least first semester), students play with conditional branchings, arrays, loops and function calls. Implementation is done in C. The classical difficulties related to low-level programming are hidden. The difficult aspects of C programming, such as addressing, pointers and memory handling are studied during second year (or second semester). More advanced aspects of algorithms are studied in third year:

- Abstract data types (stacks, queues, sets, ...) and how they can be implemented (arrays, linked lists)
- Complexity (in time and space), asymptotic orders and big-O notation. Array sorting algorithms are textbook cases for studying complexity.
- Advanced data structures: priority queues implemented using min/max-heaps, or balanced trees
- Basic graph traversal algorithms (depth-first search, breadth-first search), Dijkstra's shortest path algorithm and Ford-Fulkerson's augmenting path method for the max-flow problem

Object-oriented programming in C++

Level(s): engineering school (3rd and 4th years)

Institution(s) : INSA-CVL

Keywords: abstraction, class, encapsulation, inheritance, genericity, polymorphism, templates.

The prerequisite of this course is the 'Algorithmics and programming' described previously. After some reminder about the C language, object-oriented programming is studied as a programming paradigm, regardless of any language. This includes the separation of a project into meaningful modules, the modeling

of real-world entities as objects, the need for hiding the internals of objects, yielding the notion of private and public members. Then, implementation in C++ is studied, with classes, operator overloading, and inheritance.

The advanced object-oriented programming course deals with virtual member functions, abstract classes, interfaces, and polymorphism. Template programming is seen as a way of implementing genericity. The usage of generic containers of the standard C++ library is tackled. Students play with a Qt GUI application, in order to get familiar with event-based programming.

Image processing and computer vision

Level(s): engineering school (5th year)

Institution(s) : INSA-CVL

Keywords: partial differential equations, denoising, optical flow, keypoint detectors, Markov random fields, graph cuts.

My contribution to the course on image processing deals with Partial Differential Equations (PDEs). The different properties of PDEs (such as unicity of solution or convergence) and properties of discrete schemes for solving them numerically (consistency, convergence, stability) are studied. Discretization using finite differences and explicit (forward) Euler scheme are tackled in detail. In practical courses, PDEs are applied to three domains:

- denoising: heat diffusion, mean curvature motion, and Perona-Malik anisotropic diffusion.
- optical flow estimation, with the Horn & Schunck method.
- segmentation, namely the two-phase piecewise-constant segmentation, with level sets (Chan-Vese model).

I am in charge of the course on computer vision. In order to provide a wide scope of computer vision applications, while being able to dig into the core of algorithms, two different domains are studied in detail:

- keypoint detection for matching. Resting on the derivative operators for edge detection, Harris corners are studied as the most basic keypoints. The limitations of this detector highlight the need of invariance to rigid transformations and scaling. Multiscale modeling (Gaussian and Laplacian-of-Gaussian pyramids) is addressed, which finally leads to SIFT point detection.
- segmentation using graph cuts. Multiclass segmentation is formulated as an optimization problem on a Markov random field (MRF). The MRF is presented as a graphical model under a Bayesian framework, involving probabilities over fields of hidden and observed random variables. It is shown that minimizing the energy, derived from the probabilistic framework, is equivalent to finding a minimal cut in a graph built from the image. The Ford-Fulkerson algorithm is studied in detail to solve the equivalent max-flow problem.

For both keypoint detection and segmentation using graph cuts, practical courses are dedicated to implementing methods in C++ using the OpenCV library.

1.4 Summary of research activities

1.4.1 Topics

My activities are in the vast domains of image processing and computer vision. From the beginning of my PhD in 2004 until now, the major part of my contributions has been dedicated to object segmentation in images and videos. A smaller subset of my work deals with shape analysis and matching. These contributions were made while occupying different positions:

- from 2004 to 2008, as a PhD student and assistant professor in the *Laboratoire d’Informatique (LI)* in Tours.
- from 2008 to 2009, as a postdoc at the *CEntre de REcherche en MAthématiques de la DEcision (CEREMADE)* in Paris-Dauphine university.
- from 2009 to 2015, as an associate professor in the *Laboratoire d’Informatique en Image et Systèmes d’Information* in Lyon. I was a member of the Imagine team.
- from 2015, when I came back to the *Laboratoire d’Informatique (LI)* in Tours as an associate professor, until now. I am a member of the *Reconnaissance des Formes et Analyse d’Images (RFAI)* team (=Pattern Recognition and Image Analysis). In this period, the *LI* has been renamed as *Laboratoire d’Informatique Fondamentale et Appliquée de Tours (LIFAT)*.

These positions are shown in a timeline form in Fig. 1.1. Research topics will be introduced in detail in Chapter 3 and developed all along the manuscript. They are summarized in the current section so that the reader can relate them to the subject of PhD theses presented in Section 1.4.2 and publications in Section 1.5.

The first main part of my work in segmentation has its origin in my PhD thesis on active contours and continued during my postdoc on minimal paths. As the core feature of these methods, curves are determined as minimizers of energy functionals. Curves build the contour of the object of interest, and are constrained by user-provided points. This work was continued and led to publications [SSVM09, VISAPP12, BMVC13, IJCV15].

Segmentation is extended to the temporal domain in the second main part of my work. It includes object detection with statistical background models (PhD thesis of Imtiaz Ali [PAA14, PRL12, SCIA11]) and object space-time segmentation using color distributions [ICIP11, BMVC11].

The third and last main part of my work deals with shape analysis and matching, a process that would typically come after segmentation. This work is related to the PhD of Aurélie Leborgne, on skeleton-based shape matching [ISVC14, JVCIR15, ICIP16]. It was later extended to the study of the properties of Euclidean distance maps [JMIV19].

Circa 2017, I got more involved in machine learning for computer vision. As many researchers in image processing and computer vision, I dove into the world of convolutional neural networks. I get increasingly interested in modern machine learning methods, such as deep convolutional neural networks (co-supervision of PhD student Fabien Baradel on structured deep learning for video analysis [CVPR18, ECCV18, BVMC18] and Luong Phat Nguyen on video classification using space-time texture patterns [VISAPP20]). One may note that this part of my work is NOT addressed in detail in this habilitation report, which is justified in Chapter 3.

1.4.2 Supervision

PhD students

I have been involved in the co-supervision of 5 PhD theses in computer science. Three of them were defended, one is ongoing and one was interrupted. They are presented in this section by chronological order. Positions of colleagues involved in the supervisions are shortened to: AP for Associate Professor, HAP for Habilitated Associate Professor and PR for Full Professor.

Imtiaz ALI

PhD thesis in computer science, *Université Lumière Lyon 2 / LIRIS*

Object detection in moving backgrounds

Supervision: Laure TOUGNE (PR Lyon 2, 50%), Julien MILLE (50%, started in october 2009)

Period: october 2008 - march 2012

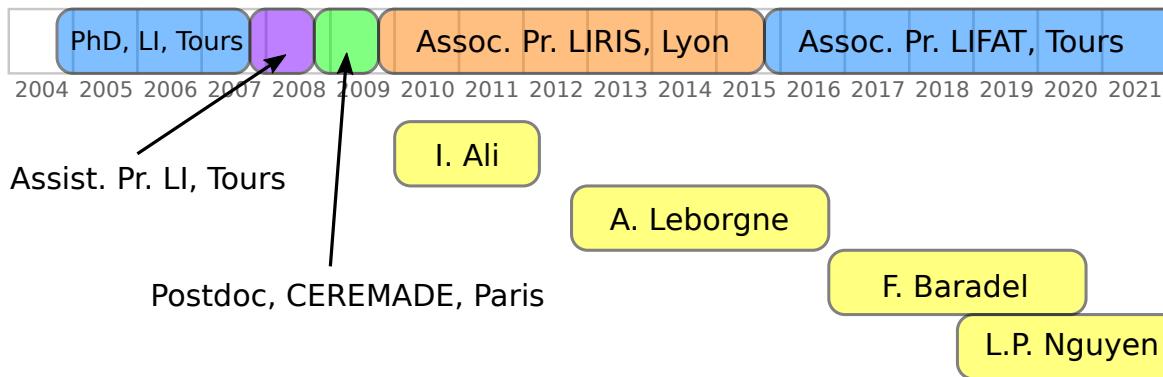


Figure 1.1: Timeline of held positions, with locations, and supervised PhD students (defended and ongoing)

Funded by a grant of the Ministry of Science and Technology of Pakistan

Defended in march 2012

NB: my involvement in the Imtiaz's PhD started after he had done his first year of under the sole supervision of Laure Tougne (see the timeline in Fig. 1.1).

Kitti PURITAT (interrupted)

PhD thesis in computer science, *Université Lumière Lyon 2 / LIRIS* (joint supervision with Chiang Mai University, Thailand)

Geometric and statistical methods for shape analysis

Supervision: Laure TOUGNE (PR Lyon 2, 50%), Julien MILLE (50%)

Period: october 2011 - june 2012

Funded by a grant of Chiang Mai University, Thailand

Aurélie LEBORGNE

PhD thesis in computer science, *INSA Lyon / LIRIS*

Skeleton-based shape matching

Supervision: Laure TOUGNE (PR Lyon 2, 50%), Julien MILLE (50%)

Period: october 2012 - july 2016

Funded by a grant of the French ministry of research

Defended in july 2016

Fabien BARADEL

PhD thesis in computer science, *INSA de Lyon / LIRIS*

Structured deep learning for video analysis

Supervision: Christian WOLF (HAP INSA Lyon, 80%), Julien MILLE (20%)

Period: octobre 2016 - june 2020

Funded by the DeepVision ANR project

Defended in june 2020

Luong Phat NGUYEN

PhD thesis in computer science, *Université de Tours / LIFAT*

Video classification using space-time texture patterns

Supervision: Donatello CONTE (HAP Univ. Tours), 20%, Dominique LI (AP Univ. Tours, 30%), Julien MILLE (30%), Nicolas RAGOT (HAP, Univ. Tours, 20%)

Period : october 2018 -

Funded by a grant of *Région Centre Val de Loire*

Postdoctoral students

Rizwan Ahmed KHAN

Postdoc in computer science, *INSA Lyon / LIRIS*

Supervision: Christian WOLF (HAP *INSA Lyon*), Julien MILLE

Detection and recognition of objects in urban scenes acquired with a mobile camera

Period: december 2013 - december 2014

Funded by the RIVIERE project

1.4.3 Involvement in national and international projects

Head of a French projet Jeune Chercheur du GdR ISIS

Young researcher program of the French research group 'Information, Signal, Image and Vision'

Project title: cooperation of minimal paths and polygonal active contours for object detection

Funding: CNRS (*GdR ISIS*)

Other project members: Sébastien BOUGLEUX (AP *Université de Caen-Basse Normandie, GREYC*), Laurent COHEN (CNRS research director, *Université Paris-Dauphine, CEREMADE*)

Period: september 2010 - december 2011 (16 months)

The project report is available online at <http://www.gdr-isis.fr/projets-de-recherche.html>

Member of French ANR project ReVeS

Project title: *REconnnaissance de VÉgétaux pour les interfaces Smartphones (ReVeS)* (Plant recognition for Smartphone devices)

Funding: ANR (French national research funding agency)

Academic partners: *LIRIS, LISTIC, EVS*

Period: october 2010 - september 2013 (36 months)

Head: Laure TOUGNE (PR *Lyon 2 / LIRIS*)

Website: <http://liris.univ-lyon2.fr/reves/>

Member of French project RiViERE

Project title: *Renaturer la Ville : facteur de RisquE ou de bien-être social aux yeux des habitants et des usagers ? L'apport d'une médiation paysagère (RiViERE)* (Restoring nature in cities: risk or social well-being for residents? What mediation on landscapes can bring)

Funding: LabEx *Intelligence des Mondes Urbains (IMU)*

Academic partners: *EVS, LIRIS, Ecologie Microbienne Lyon, LEHNA*

Period: october 2013 - september 2015 (24 months)

Head: Marylise COTTET (CNRS researcher, *EVS*)

Website: <http://imu.universite-lyon.fr/projet/pratic-riviere-urbieau/>

Member of international ANR/NSERC project DeepVision

Project title: DeepVision, Seeing and understanding humans with deep structured models

Funding: ANR / NSERC (Natural Sciences and Engineering Research Council of Canada)

Academic partners: *LIRIS, LIFAT, LIP6, University of Guelph (Canada), Simon Fraser University (Canada)*

Period: september 2016 - march 2019 (42 months)

Head: Christian WOLF (HAP *INSA Lyon*, France), Graham TAYLOR (AP, University of Guelph, Canada)

Website: <https://projet.liris.cnrs.fr/deepvis/>

NB: at submission time, I was at the *LIRIS* in Lyon, so my affiliation appears as such in reports and on the website.

1.4.4 Reviewing

International journals

I am a regular reviewer for IEEE Transactions on Image Processing, Computer Vision and Image Understanding, and Pattern Recognition Letters.

More occasionally, I made reviews for Pattern Recognition, Journal of Mathematical Imaging and Vision, IEEE Transactions on Systems, Man and Cybernetics, Image and Vision Computing, Journal of Visual Communication and Image Representation, Neural Processing Letters, Computer Methods and Programs in Biomedicine.

International conferences

I made reviews for CVPR 2020, IJCAI 2020, IJCAI 2019, and EUSIPCO 2012.

Projects

I reviewed a submission for the French *Programme de Coopération Scientifique Inter-universitaire (PCSI)*, in the context of call of *Agence Universitaire de la Francophonie (AUF)*, in 2010.

I reviewed a submission for an ANR project, in 2012.

1.4.5 Local responsibilities

Since september 2019, I am the representative (= local head) of the LIFAT members teaching at *INSA-CVL*. As being the interface between researchers in computer science and directors of the institute, this responsibility implies, among other tasks, to get involved in the recruitment process of assistant and associate professors in computer science at *INSA* and centralize expenses related to research.

I am a member of the steering committee of *Réseau Thématique de Recherche (RTR)* "Données, Intelligence Artificielle, Modélisation et Simulation", which groups people of three local higher education institutes (university of Orleans, university of Tours and *INSA*) and other institutions (*INRAE*, *CNRS*, hospitals, etc) established in *Région Centre Val de Loire*. The *RTR* gathers local researchers around topics related to computer science. Its main purpose is to fund missions, Master internships and organization of meetings within small workgroups. <https://www.univ-orleans.fr/lifo/evenements/RTR-DIAMS>

I was a member of two committees for recruiting associate professors at *INSA / LIFAT*: the first one as an external member in 2014, while I was at the *LIRIS* in Lyon, and the second one as a local member in 2018.

1.4.6 Software

Since 2010, I have been making C++ libraries and applications available on source code repositories.

The first one implements the narrow band region-based active contour developed during my PhD [ICIP07] and subsequently improved [CVIU09, EMMCVPR09]. The library has been available on my personal webpage since 2010 and was updated for the last time in 2013. The code can be downloaded from <https://julien-mille.github.io/code.html>

The second one implements the interactive segmentation method using the piecewise-geodesic path model published in [BMVC13, IJCV15] and described in detail in Chapter 4. It has been available since may 2015 at <https://github.com/julien-mille/comb-pgeo-paths>.

More recently, in the context of the supervision of L.P. Nguyen, we got interested in the refinement of coarse optical flow fields estimated by means of convolutional neural networks. The original DeepFlow algorithm [Weinzaepfel et al., 2013, Revaud et al., 2016], available at <https://thoth.inrialpes.fr/src/deepflow>, was ported to GPU in C++ using CUDA and OpenCV libraries. This reimplementation is available since

september 2019 at <https://github.com/julien-mille/DeepFlowCUDA>.

Finally, I developed a simple image annotation tool in C++ with the Qt GUI library. It allows a pencil-based labeling of pixels, in order to make ground truth segmentations. It has been available since april 2020 at <https://github.com/julien-mille/PixelLabeling>

1.5 List of publications

Journal papers (international)

- [JMIV19] **J. Mille**, A. Leborgne and L. Tougne. Euclidean distance-based skeletons: a few notes on average outward flux and ridgeness. *Journal of Mathematical Imaging and Vision*, 61(3):310–330, 2019.
- [JVCIR15] A. Leborgne, **J. Mille**, and L. Tougne. Noise-resistant Digital Euclidean Connected Skeleton for graph-based shape matching. *Journal of Visual Communication and Image Representation*, 31:165–176, 2015.
- [IJCV15] **J. Mille**, S. Bougleux and L. Cohen. Combination of piecewise-geodesic paths for interactive segmentation. *International Journal of Computer Vision*, 112(1):1–22, 2015.
- [CVIU14] C. Wolf, E. Lombardi, **J. Mille**, O. Çeliktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandréa, C.-E. Bichot, C. Garcia, B. Sankur. Evaluation of video activity localizations integrating quality and quantity measurements. *Computer Vision and Image Understanding*, 127:14–30, 2014.
- [PAA14] I. Ali, **J. Mille**, L. Tougne. Adding a rigid motion model to foreground detection: application to moving object detection in rivers. *Pattern Analysis and Applications*, 17(3):567–585, 2014.
- [CVIU13] G. Cerutti, L. Tougne, **J. Mille**, A. Vacavant, D. Coquin. Understanding leaves in natural images - a model-based approach for tree species identification. *Computer Vision and Image Understanding*, 117(10):1482–1501, 2013.
- [PRL12] I. Ali, **J. Mille**, L. Tougne. Space-time spectral model for object detection in dynamic textured background. *Pattern Recognition Letters*, 33(13):1710–1716, 2012.
- [Media11] K. Hameeteman, M.A. Zuluaga, M. Freiman, L. Joskowicz, O. Cuisenaire, L. Florez-Valencia, M.A. Gülsün, K. Krissian, **J. Mille**, W.C.K Wong, M. Orkisz, H. Tek, M. Hernández Hoyos, F. Benmansour, A.C.S. Chung, S. Rozie, M. van Gils, L. van den Borne, J. Sosna, P. Berman, N. Cohen, P. Douek, I. Sánchez, M. Aissat, M. Schaap, C. Metz, G.P. Krestin, A. van der Lugt, W.J. Niessen, T. van Walsum. Evaluation framework for carotid bifurcation lumen segmentation and stenosis grading. *Medical Image Analysis*, 15(4):477–488, 2011.
- [CVIU09] **J. Mille**. Narrow band region-based active contours and surfaces for 2D and 3D segmentation. *Computer Vision and Image Understanding*, 113(9):946–965, 2009.
- [IJCVB08] J. Olivier, **J. Mille**, R. Boné, and J.-J. Rousselle. Dynamic neighborhoods in active surfaces for 3D segmentation. *International Journal for Computational Vision and Biomechanics*, 1(2):173–180, 2008.

Conference papers (international)

- [ICLR20] F. Baradel, N. Neverova, **J. Mille**, G. Mori, C. Wolf. COPHY: COunterfactual learning of PHYSical dynamics. *International Conference on Learning Representations (ICLR)*, 2020.
- [VISAPP20] L.P. Nguyen, **J. Mille**, N. Ragot, D. Conte, D. Li. Trajectory extraction and deep features for classification of liquid-gas flow under the context of forced oscillation. *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020.

- [IAMPS19] A. Leborgne, **J. Mille**, L. Tougne. Hierarchical structure of the veins of the tree leaves. *International Workshop on Image Analysis Methods for the Plant Sciences (IAMPS)*, 2019.
- [ECCV18] F. Baradel, N. Neverova, C. Wolf, **J. Mille** and G. Mori. Object level visual reasoning in videos. *European Conference in Computer Vision (ECCV)*, pages 106–122, 2018.
- [BVMC18] F. Baradel, C. Wolf, **J. Mille**, Human activity recognition with pose-driven attention to RGB. *British Machine Vision Conference (BMVC)*, 2018.
- [CVPR18] F. Baradel, C. Wolf, **J. Mille**, G. Taylor, Glimpse Clouds: human activity recognition from unstructured feature points. *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 469–478, 2018.
- [IAMPS18] A. Leborgne, **J. Mille**, L. Tougne, Automatic extraction of leaf network of veins. *International Workshop on Image Analysis Methods for the Plant Sciences (IAMPS)*, 2018.
- [ICCVW17] F. Baradel, C. Wolf, **J. Mille**, Human action recognition: pose-based attention draws focus to hands. *3rd International Workshop on Observing and Understanding Hands in Action (HANDS), in conjunction with ICCV*, 2017.
- [ICIP16] A. Leborgne, **J. Mille**, L. Tougne. Hierarchical skeleton for shape matching. *IEEE International Conference in Image Processing (ICIP)*, 2016.
- [ISVC14] A. Leborgne, **J. Mille**, L. Tougne. Extracting noise-resistant skeleton on digital shapes for graph matching. *International Symposium on Visual Computing (ISVC)*, volume 8887 of *LNCS*, pages 293–302, 2014.
- [CS14] **J. Mille**, S. Bougleux, L. Cohen. Combination of piecewise-geodesic curves for interactive image segmentation. *International Symposium on Curves and Surfaces*, 2014.
- [BVMC13] **J. Mille**, S. Bougleux, L. Cohen. Combination of paths for interactive segmentation. *British Machine Vision Conference (BMVC)*, 2013.
- [ICIP13] G. Cerutti, L. Tougne, **J. Mille**, A. Vacavant, D. Coquin. A model-based approach for compound leaves understanding and identification. *IEEE International Conference on Image Processing (ICIP)*, pages 1471–1475, 2013.
- [VISAPP12] **J. Mille**, S. Bougleux, L. Cohen. Minimally overlapping paths sets for closed contour extraction. *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 259–268, 2012.
- [BVMC11] **J. Mille**, J-L. Rose. Multitarget region tracking based on short-sight modeling of background and color distribution temporal variation. *British Machine Vision Conference (BMVC)*, 2011.
- [ICIP11] **J. Mille**, J-L. Rose. Region tracking with narrow perception of background. *IEEE International Conference on Image Processing (ICIP)*, pages 493–496, 2011.
- [CLEF11] G. Cerutti, L. Tougne, **J. Mille**, A. Vacavant, D. Coquin. Guiding active contours for tree leaf segmentation and identification. CLEF Workshop, 2011.
- [SCIA11] I. Ali, **J. Mille**, L. Tougne. Wood detection and tracking in videos of rivers. *Scandinavian Conference on Image Analysis (SCIA)*, pages 646–655, 2011.
- [ISBI10] **J. Mille**, L. Cohen. 3D CTA image segmentation with a generalized cylinder-based tree model. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, pages 1045–1048, 2010.
- [EMMCVPR09] **J. Mille**, L. Cohen. A local normal-based region term for active contours. *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 5681 of *LNCS*, 2009.

- [MMBIA09] **J. Mille** and L. Cohen. Deformable tree models for 2D and 3D branching structures extraction. *CVPR Workshop on Mathematical Methods in Biomedical Image Analysis*, 2009.
- [SSVM09] **J. Mille** and L. Cohen. Geodesically linked active contours: evolution strategy based on minimal paths. *International Conference on Scale-Space and Variational Methods (SSVM)*, volume 5567 of *LNCS* Springer, pages 163–174, 2009.
- [ECCV08] **J. Mille**, R. Boné, and L. Cohen. Region-based 2D deformable generalized cylinder for narrow structures segmentation. *European Conference on Computer Vision (ECCV)*, volume 5303 of *LNCS*, Springer, pages 392–404, Marseille, France, 2008.
- [ICIP07] **J. Mille**, R. Boné, P. Makris, and H. Cardot. 2D and 3D deformable models with narrow band region energy. *IEEE International Conference on Image Processing (ICIP)*, pages 57–60, San Antonio, USA, 2007.
- [ISPA07] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Narrow band region approach for active contours and surfaces. *IEEE International Symposium on Image and Signal Processing and Analysis (ISPA)*, Istanbul, Turquie, 2007.
- [CBMS07] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Segmentation and tracking of the left ventricle in 3D MRI sequences using an active surface model. *22th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, pages 257–262, Maribor, Slovénie, 2007.
- [ComplIMAGE06a] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Fast estimation of skeleton points on 3D deformable meshes. *Computational Modelling of Objects Represented in Images: fundamentals, methods and applications (ComplIMAGE)*, Coimbra, Portugal, 2006.
- [ComplIMAGE06b] J. Olivier, **J. Mille**, R. Boné and J.-J. Rousselle. Active surfaces acceleration methods. *Computational Modelling of Objects Represented in Images: fundamentals, methods and applications (ComplIMAGE)*, Coimbra, Portugal, 2006.
- [ICIP06] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Greedy algorithm and physics-based method for active contours and surfaces: a comparative study. *IEEE International Conference on Image Processing (ICIP)*, pages 1645–1648, Atlanta, USA, 2006.
- [3DPVT06] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Exploring boundary concavities in active contours and surfaces. *3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission*, Chapel Hill, USA, 2006.
- [VIIP05] **J. Mille**, R. Boné, P. Makris, and H. Cardot. 3D segmentation using active surface: a survey and a new model. *5th International Conference on Visualization, Imaging & Image Processing (VIIP)*, Benidorm, Espagne, 2005.

Conference papers (in french)

- [RFIA16] A. Leborgne, **J. Mille**, L. Tougne. Squelette pondéré de persistance pour l'appariement de formes. *Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, 2016.
- [CORESA14] A. Leborgne, **J. Mille**, L. Tougne. Squelette Euclidien Discret Connecté (DECS) résistant au bruit pour l'appariement de formes basé graphes. *Compression et Représentation des Signaux Audiovisuels (CORESA)*, 2014.
- [GRETSI09] **J. Mille**, L. Cohen. Reconstruction de structures arborescentes par chemins minimaux et bande déformable. GRETSI : Traitement du Signal et des Images, 2009.

[GRETSI07] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Segmentation et suivi de l'endocarde dans des séquences IRM 3D par surface active. GRETSI : Traitement du Signal et des Images, pages 81–84, Troyes, France, 2007.

[CORESA06a] **J. Mille**, R. Boné, P. Makris, and H. Cardot. Surface active pour la segmentation d'images 3D : comparaison de méthodes d'évolution. *Compression et Représentation des Signaux Audiovisuels (CORESA)*, pages 120–125, Caen, France, 2006.

[CORESA06b] J. Olivier, **J. Mille**, R. Boné and J.-J. Rousselle. Accélération de surfaces actives. *Compression et Représentation des Signaux Audiovisuels (CORESA)*, pages 225–229, Caen, France, 2006.

Thesis manuscript

[THESE07] **J. Mille**, Modèles déformables pour la segmentation et le suivi en imagerie 2D et 3D. *Thèse de l'Université François Rabelais de Tours*, décembre 2007.

Chapter 2

Résumé en français

2.1 Introduction

Segmenter des objets ? Comprendre le contenu sémantique d'une image ou d'une vidéo peut impliquer une étape de segmentation, qui consiste en un étiquetage, pixel à pixel, du fond et des objets présents dans la scène. Cette segmentation peut être une finalité en soi. En imagerie médicale par exemple, le praticien souhaite délimiter précisément le contour d'une structure anatomique. Le diagnostic ne nécessitera pas d'autres étapes automatisées, mis à part le calcul, assez simple, de descripteurs de la forme segmentée (périmètre, aire, etc). A l'inverse, la segmentation peut n'être qu'une étape d'un processus d'analyse de plus haut niveau. Par exemple, l'indexation de vidéos par le contenu peut bénéficier de la segmentation pour extraire les types d'objets et de mouvements.

Nous étudions la segmentation à un niveau sémantique où, au minimum, la notion d'objet et de fond est présente. Cela exclut la *sursegmentation*, obtenue par exemple avec des méthodes de type *superpixels* [Achanta et al., 2012]. Prenons par exemple l'image de microscopie cellulaire de la Fig. 2.1(a). Dans le cas illustré sur la Fig. 2.1(b), la segmentation passe par une reconstruction du contour d'une cellule donnée, avec une intervention manuelle au préalable pour placer des points de repère. La méthode des ciseaux intelligents [Mortensen and Barrett, 1998] produit par exemple ce type de résultats¹.

Si aucune information *a priori* n'est fournie sur la localisation approximative de l'objet d'intérêt, nous partons du principe que l'image doit être étiquetée dans sa totalité, comme sur la Fig. 2.1(c). Dans la forme la plus simple d'une segmentation "globale", chaque pixel est étiqueté soit "objet", soit "fond". Les objets sont alors

¹Ce type de segmentation ne génère pas directement un étiquetage individuel des pixels. Sous réserve que les chemins liant les points de repère forment un contour qui ne s'intersecte pas lui-même, un étiquetage peut être obtenu par remplissage.

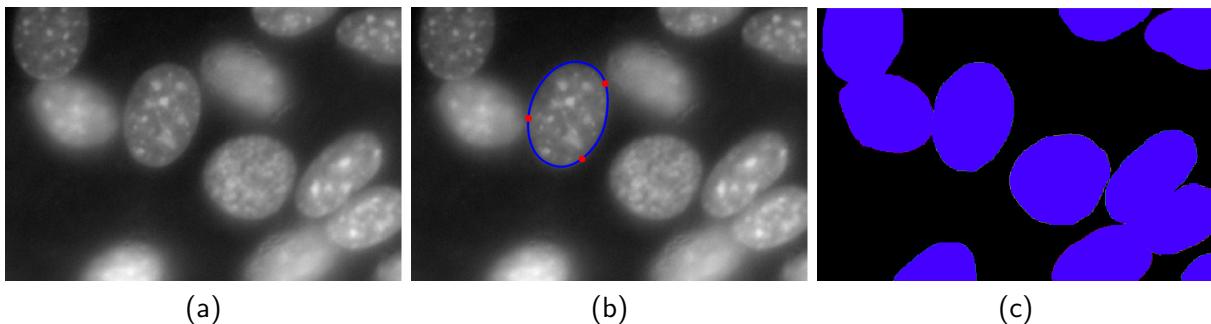


Figure 2.1: Différents types de segmentation en imagerie cellulaire microscopique : (a) Image initiale, (b) Segmentation d'un objet unique avec points de guidage fournis par l'utilisateur, (c) Segmentation sémantique

déterminés par extraction des composantes connexes de pixels marqués "objet", comme dans les approches de type soustraction de fond [Stauffer and Grimson, 1999]. Si l'on fait un pas sémantique supplémentaire, on souhaite déterminer non seulement si un pixel appartient à un objet, mais également la classe de l'objet auquel il appartient ("route", "pelouse", "ciel", etc). Auquel cas, les classes sont fixées au préalable, leur granularité dépendant du niveau sémantique requis par l'application (une classe "animal" est-elle suffisante ? Ou faut-il distinguer les classes "chat", "chien" et "vache" ?). La segmentation est un aspect central de mon habilitation. Une grande partie des travaux décrits dans ce rapport entrent dans le contexte de la segmentation d'objet unique avec interaction utilisateur, évoquée par la Fig. 2.1(b), ou de la segmentation d'objets multiples, illustrée sur la Fig. 2.1(c).

Apparier des objets ? Mettons de côté la segmentation en tant que finalité, et considérons la tâche de classification d'images, où l'on souhaite déterminer la catégorie à laquelle une image peut appartenir, en se basant sur le ou les objet(s) qu'elle contient. Dans une architecture "traditionnelle"² de reconnaissance, l'extraction des *caractéristiques* (ou *descripteurs*) des objets et la classification sont des étapes bien distinctes. Les travaux anciens cités dans l'étude bibliographique [Andreopoulos and Tsotsos, 2013, Section 2] sont de bons exemples de ce schéma classique.

Une fois qu'un objet est segmenté, il peut être associé à une catégorie connue, au moyen d'une extraction de caractéristiques sur sa forme et/ou son apparence³, et en fournissant ces caractéristiques à un classifieur. Dans ce paragraphe, nous nous focalisons sur la classification d'objets en nous basant uniquement sur leur forme, dans la mesure où l'apparence (couleur ou texture) présente dans l'image initiale n'est pas prise en compte. Il s'agit de reconnaissance d'objets binaires. Un jeu de données très utilisé pour cette tâche est la base Kimia216 [Sebastian et al., 2004], dont la Fig. 2.2 représente un extrait. Dans sa forme la plus simple, la classification d'objets binaires peut être réalisée en calculant un nombre fixé de caractéristiques, le vecteur réel obtenu étant donné en entrée à un classifieur. Autrement, la classification peut être abordée au moyen d'un appariement. Dans ce contexte, une forme requête (à classifier) est appariée à plusieurs formes représentatives, ou prototypes, de chaque classe. La forme requête est affectée à la classe conduisant aux meilleurs appariements.

En plus de la segmentation, l'appariement de formes binaires est le deuxième aspect fondamental des travaux décrits dans ce manuscrit. Mes collègues et moi avons été amenés à nous intéresser au *squelette* [Blum and Nagel, 1978], représentation ayant de nombreuses propriétés intéressantes pour l'appariement, notamment grâce à la robustesse qu'il induit par rapport aux déformations articulées.

Liens entre les contributions Deux grands objectifs cohabitent dans ce rapport : la segmentation, où l'objet est un résultat, et l'appariement, où l'objet est une entrée. Il existe un lien évident entre ces deux objectifs, l'appariement pouvant être réalisé après une segmentation dans une architecture classique de détection et reconnaissance d'objets. Je développe ici les aspects transversaux de mes contributions :

- Le point de départ de la segmentation ou de l'appariement est toujours un modèle continu, à la fois pour l'ensemble de départ (le plan) et pour l'ensemble d'arrivée de la fonction image. L'image (image fixe ou *frame* d'une vidéo) est représentée par une fonction continûment dérivable. Pour une image en niveau de gris, on a

$$f : \mathcal{D} \subset \mathbb{R}^2 \longrightarrow \mathbb{R}, \quad (2.1)$$

²L'appellation "traditionnelle" est ici utilisée par opposition aux architectures modernes à base de réseaux de neurones profonds, prenant directement l'image en entrée et produisant un ensemble de localisations et classes d'objets

³En vision par ordinateur, l'apparence désigne l'ensemble des propriétés image d'un objet, indépendamment de la géométrie et topologie de son contour (et de son mouvement dans le cas de vidéos). La plupart du temps, cela fait référence aux couleurs et/ou aux textures.

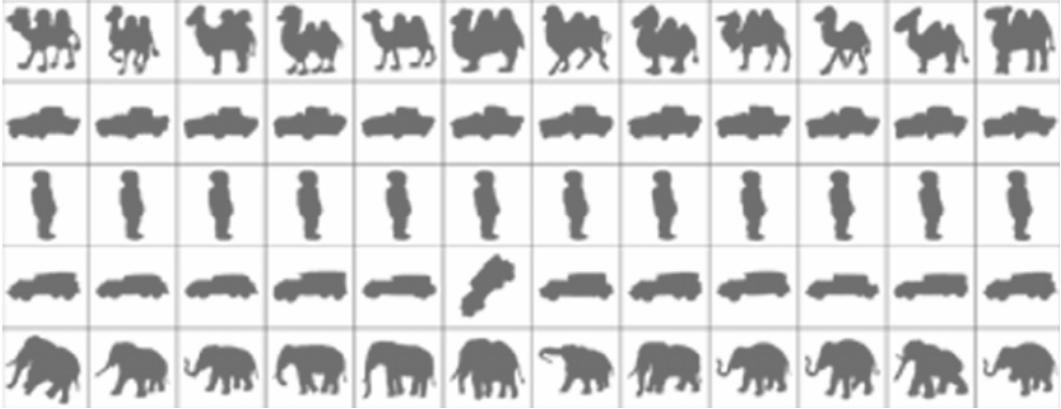


Figure 2.2: Un extrait du jeu de données Kimia216 [Sebastian et al., 2004]

où \mathcal{D} est le domaine spatial. Pour une image à valeur vectorielle (couleur),

$$\begin{aligned} \mathbf{f} & : \mathcal{D} \subset \mathbb{R}^2 \longrightarrow \mathcal{Q} \subset \mathbb{R}^3 \\ \mathbf{f}(\mathbf{x}) & = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ f_3(\mathbf{x})]^\top, \end{aligned} \quad (2.2)$$

où \mathcal{Q} est un espace colorimétrique à trois composantes (RGB, YUV, Lab, etc). Nous considérons un objet, noté Ω , ou plusieurs objets $\{\Omega_i\}_{i=1..n}$, tous sous-ensembles de \mathcal{D} . Bien entendu, tout modèle continu implique une discréétisation à un moment donné, pour permettre sa mise en œuvre algorithmique. Le domaine image \mathcal{D} sera discréétisé en une grille rectangulaire $\hat{\mathcal{D}}$, sous-ensemble de \mathbb{Z}^2 .

- Toutes les méthodes développées ici impliquent une minimisation d'énergie à une certaine étape, où la solution est une courbe ou une fonction d'étiquetage. L'énergie est la modélisation mathématique des propriétés requises de la solution. Par exemple, nous souhaitons que le contour des objets soit lisse, nous souhaitons que les distributions des couleurs à l'intérieur des objets soient différentes de celle du fond, etc. Dans les méthodes de segmentation présentées, la minimisation d'énergie (très exactement, le contour ou la fonction d'étiquetage qui minimise cette énergie) est l'objectif final, alors qu'elle n'est qu'une étape parmi d'autres dans l'approche d'appariement de formes. Les énergies à minimiser sont non-convexes. Nous utilisons des méthodes d'optimisation continue ou combinatoire, qui convergent vers un minimum local.

Les trois sections suivantes sont des résumés respectifs des contributions décrites dans les chapitres 4, 5 et 6.

2.2 L'objet pré-identifié: segmentation à l'aide de chemins

2.2.1 Introduction

Je résume dans cette section les méthodes développées pour la segmentation d'un objet unique avec interaction *a priori*. Les algorithmes décrits ici visent des applications où l'utilisateur est prêt pour un minimum de travail manuel (par exemple : segmentation d'images médicales par un praticien, édition d'images/vidéos, etc). L'utilisateur fournit plusieurs points de guidage le long du contour de l'objet. La segmentation consiste à reconstruire le contour en connectant les points de guidage consécutifs à l'aide de chemins, comme illustré sur la Fig. 2.1(b).

L'ensemble des méthodes décrites ici tirent leur inspiration de la méthode des chemins minimaux de [Cohen and Kimmel, 1997]. Etant donnés deux points, fixés, la méthode des chemins minimaux consiste à relier ces deux points par une courbe, empruntant les contours pertinents dans l'image. La courbe est déterminée comme le minimiseur global d'une longueur pondérée. Telle quelle, la méthode n'est pas prévue

pour reconstruire des contours fermés. Par ailleurs, créer une boucle en reliant successivement une série de points ne garantit pas d'obtenir un contour fermé qui soit *simple* (sans auto-intersection), et donc pertinent pour la segmentation. Ainsi, l'idée fondamentale des contributions décrites ici est de sélectionner une combinaison parmi un ensemble de chemins possibles, de manière à construire un contour pertinent. Nous obtenons une courbe fermée lisse par morceaux, sur laquelle une fonctionnelle d'énergie, intégrant des termes contour et région, est formulée. Au lieu de ne considérer que les plus courts chemins, nous envisageons un concept plus général, que nous appelons chemins *admissibles* [IJCV15]⁴.

Cette contribution est la partie la plus personnelle de mes travaux, dans la mesure où elle n'est pas le résultat d'un co-encadrement de doctorant. Elle est le fruit d'une collaboration avec Sébastien Bougleux du GREYC et Laurent Cohen du CEREMADE, démarrée lors de mon postdoc au CEREMADE, durant lequel j'avais travaillé sur les chemins minimaux. Les publications associées, listées en Section 1.5, sont [SSVM09, VISAPP12, BMVC13, CS14, IJCV15].

2.2.2 Existant : les chemins minimaux

Les chemins minimaux se placent dans le cadre, très général, de la minimisation d'une énergie par rapport à une courbe. Introduisons tout d'abord quelques bases sur les courbes planes⁵. Soit une courbe paramétrée \mathbf{c} ,

$$\begin{aligned}\mathbf{c} &: [0, 1] \longrightarrow \mathcal{D} \subset \mathbb{R}^2 \\ \mathbf{c}(u) &= [x(u) \ y(u)]^\top,\end{aligned}$$

où x et y sont au moins de classe C^2 . Par convention, toutes les courbes seront définies sur le domaine normalisé $[0, 1]$. La courbe \mathbf{c} est dite *régulière* si son vecteur vitesse \mathbf{c}' est non-nul partout, et *simple* si elle n'a pas de point multiple, c-à-d si elle ne s'auto-intersecte pas et n'est pas "auto-tangente".

Une *fonctionnelle* est une application d'un espace de fonctions vers son corps de scalaires. La segmentation est formulée comme la recherche d'une courbe minimisant une fonctionnelle, appelée ici *énergie*, composée généralement de termes de régularisation et de termes d'attache aux données. On se limite ici aux fonctionnelles à valeurs dans \mathbb{R} , dépendant de la courbe et de ses dérivées d'ordre 1 et 2, de forme générale

$$E[\mathbf{c}] = \int_0^1 h(\mathbf{c}(u), \mathbf{c}'(u), \mathbf{c}''(u)) du. \quad (2.3)$$

Dans l'approche des chemins minimaux, l'énergie à minimiser peut être vue comme une longueur, au sens d'une métrique hétérogène et isotrope définie via une fonction *potentiel* $P: \mathcal{D} \rightarrow \mathbb{R}^{*+}$,

$$C[\mathbf{c}] = \int_0^1 P(\mathbf{c}(u)) \|\mathbf{c}'(u)\| du. \quad (2.4)$$

La courbe doit passer par les contours de l'image. Le potentiel doit donc pénaliser un gradient d'image faible. Pour cela, on choisit

$$P(\mathbf{x}) = \frac{1}{1 + \|(f * \nabla G_\sigma)(\mathbf{x})\|} + w, \quad (2.5)$$

où G_σ est la Gaussienne 2D de moyenne nulle et de covariance $\sigma^2 \mathbf{I}$, $*$ est le produit de convolution, et $w \in \mathbb{R}^{*+}$ est le poids appliqué à la composante Euclidienne du potentiel, agissant en tant que régularisation. Etant données deux extrémités fixes \mathbf{a} et \mathbf{b} , la courbe ouverte partant de \mathbf{a} et se terminant en \mathbf{b} , minimisant l'énergie (2.4), est appelée *géodésique* (l'équivalent discret, sur un graphe orienté, est le *plus court chemin*),

$$\gamma_{\mathbf{a}, \mathbf{b}} = \operatorname{argmin}_{\mathbf{c}} \{C[\mathbf{c}]\} \quad \text{s.t.} \quad \begin{cases} \mathbf{c}(0) = \mathbf{a} \\ \mathbf{c}(1) = \mathbf{b}. \end{cases} \quad (2.6)$$

⁴Dans ce résumé et dans l'ensemble du manuscrit, des conventions différentes sont utilisées pour les références, selon qu'elles correspondent à des publications dont je suis co-auteur ou non. Les papiers dont je suis co-auteur sont référencés avec l'acronyme du journal ou de la conférence, et renvoient à la Section 1.5, tandis que les références externes utilisent le style apalike de L^AT_EX

⁵Pour un approfondissement sur les notions de géométrie différentielle, le lecteur pourra se référer aux chapitres de livre [do Carmo, 1976, Chapter 1] et [Raussen, 2008, Chapter 2]

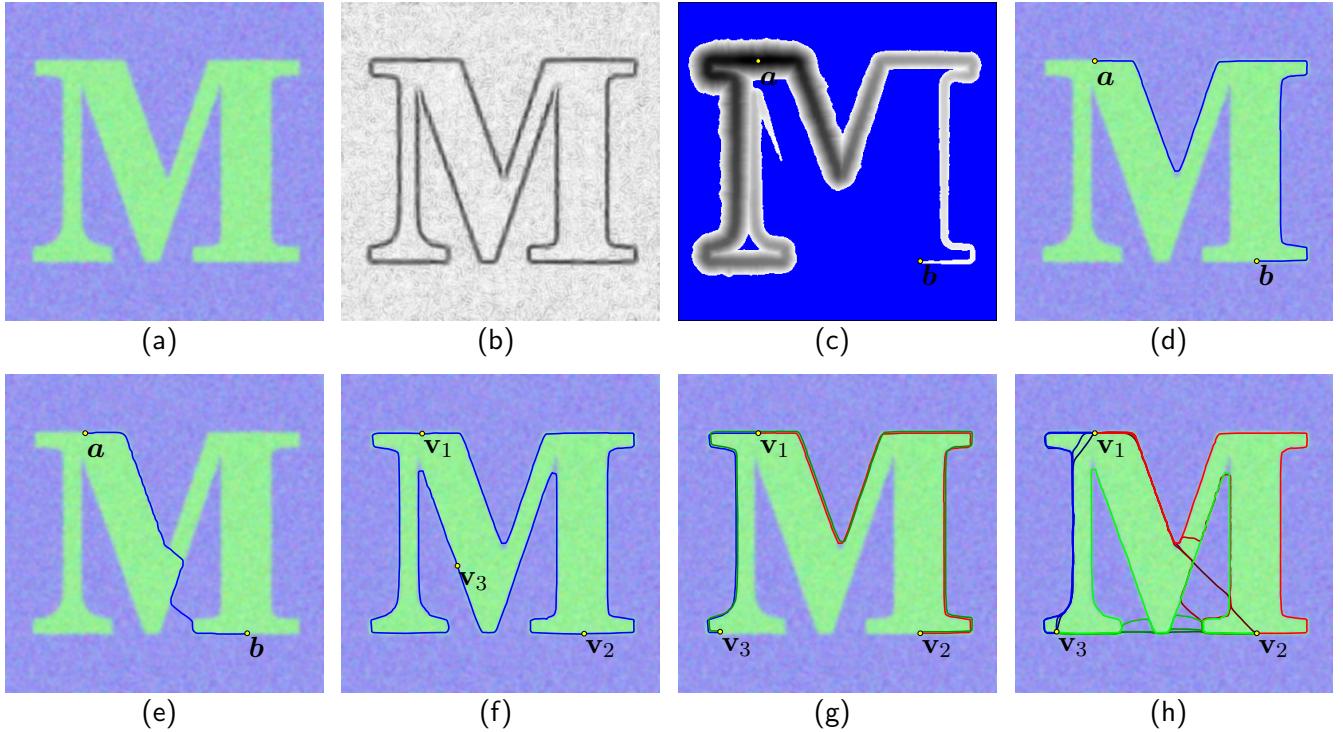


Figure 2.3: (a) Image initiale. (b) Potentiel P . (c) Action U_a (d) Avec un paramètre de régularisation w suffisamment petit, la géodésique d'un point à l'autre passe par le contour de l'objet. (e) Dû à une régularisation excessive, la géodésique emprunte un raccourci non-souhaité. (f) Courbe géodésique par morceaux pertinente. (g) Courbe géodésique par morceaux pathologique. (h) Plusieurs chemins "intéressants" par couple de germes consécutifs.

Le cheminement classique, mis en œuvre dans les approches de type contour actif [Kass et al., 1988, Chan and Vese, 2001], consisterait à appliquer le *calcul des variations* [Forsyth, 1960, Chapter 1] sur $C[c]$, afin d'en tirer une équation aux dérivées partielles, représentant la condition selon laquelle c est un extremum local de $E[c]$. Un schéma numérique de descente de gradient serait ensuite mis en place pour résoudre itérativement cette équation. Cependant, dans le cadre des chemins minimaux, le minimum global du problème (2.6) peut être obtenu en considérant la *carte de distance géodésique* [Crandall et al., 1992], également appelée *action*, définie comme suit,

$$U_a(x) = \inf_c \{C[c]\} \quad \text{s.t.} \quad \begin{cases} c(0) = a, \\ c(1) = x. \end{cases} \quad (2.7)$$

Cette action est l'unique solution de l'équation Eikonale, qui peut être résolue en $O(N \log N)$ opérations grâce à l'algorithme du Fast Marching [Tsitsiklis, 1995, Sethian, 1996], N étant le nombre de pixels sur la grille. Un exemple jouet, son potentiel et l'action issue d'un point extrémité sont illustrés respectivement sur les Fig. 2.3(a), 2.3(b) et 2.3(c). Une fois l'action U_a calculée sur le domaine \mathcal{D} , la géodésique est construite par descente de gradient de U_a , en partant de b jusqu'à atteindre a . Deux exemples de chemins apparaissent sur les Fig. 2.3(d) et 2.3(e), où l'influence du poids w est mise en évidence.

2.2.3 Construction de contours fermés

Comme décrit dans [Miranda et al., 2012], de nombreux objets nécessitent de placer plusieurs points de guidage, appelés *germes*, le long de leur contour, ceci afin de les segmenter précisément. Parmi les raisons, on peut citer :

- un manque de contraste sur certaines portions du contour, créant des contours mal définis, avec un potentiel P élevé.

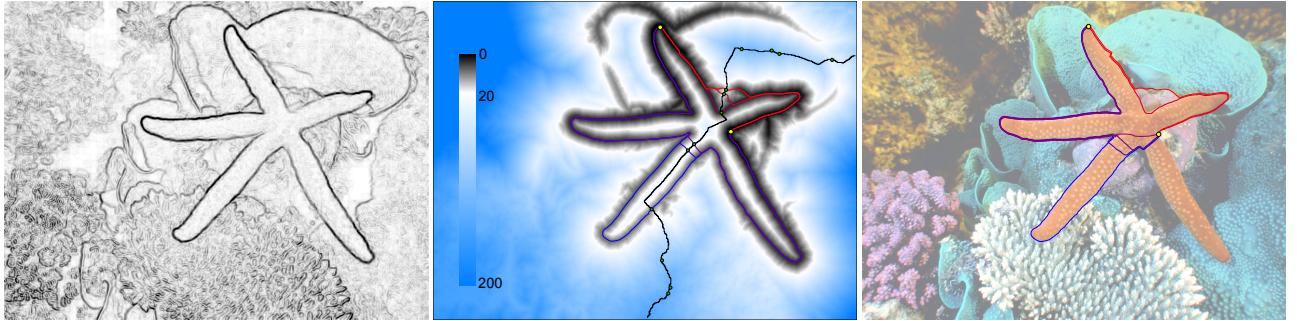


Figure 2.4: Chemins admissibles basés sur les points-selles. (Gauche) Potentiel P . (Centre) Courbe médiane (en noir) et points-selles (en vert) avec chemins admissibles correspondants, tracés sur la carte d'action combinée. (Droite) Les mêmes chemins admissibles superposés à l'image initiale

- des formes complexes, créant des "goulots", qui favorisent les raccourcis (les chemins traversent l'objet plutôt que de parcourir son contour).
- des fonds complexes, contenant d'autres objets adjacents à l'objet d'intérêt, avec des contours mieux définis.

Dans ces situations, le processus de segmentation doit tirer parti de l'information de localisation *a priori*. On suppose que l'utilisateur fournit une séquence de germes $(\mathbf{v}_i)_{1 \leq i \leq n}$, avec $n \geq 3$, ordonnés dans le sens horaire le long du contour à segmenter.

Une première approche, naïve, que nous avons développée dans [SSVM09], considère les couples de germes consécutifs $(\mathbf{v}_i, \mathbf{v}_{i+1})$, indépendamment les uns des autres. Le contour Γ est simplement construit par concaténation des géodésiques,

$$\Gamma = \text{concat}(\gamma_{\mathbf{v}_1, \mathbf{v}_2}, \gamma_{\mathbf{v}_2, \mathbf{v}_3}, \dots, \gamma_{\mathbf{v}_{n-1}, \mathbf{v}_n}, \gamma_{\mathbf{v}_n, \mathbf{v}_1}). \quad (2.8)$$

Une concaténation de géodésique n'étant pas une géodésique elle-même, cette construction est appelée *Contour Géodésique par Morceaux* (CGM). Cette approche a l'inconvénient de ne pas garantir un contour simple. Ce cas se produit particulièrement lorsque les germes sont mal répartis le long du contour, comme illustré sur la Fig. 2.3(g) (où $\gamma_{\mathbf{v}_2, \mathbf{v}_3}$ emprunte le sens contraire de celui attendu, et recouvre les chemins $\gamma_{\mathbf{v}_1, \mathbf{v}_2}$ et $\gamma_{\mathbf{v}_3, \mathbf{v}_1}$).

Pour palier ce problème, pour chaque géodésique $\gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ construite, on pourrait imaginer interdire à la géodésique suivante $\gamma_{\mathbf{v}_{i+1}, \mathbf{v}_{i+2}}$ de passer par les mêmes points que $\gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}$. Cependant, imposer aux chemins d'être disjoints a plusieurs inconvénients : cela rend la construction du contour dépendante du premier germe considéré. Par ailleurs, les contours de l'image créent des zones de potentiel faible qui peuvent être épaisses. Lorsqu'un chemin $\gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ emprunte ce type de zones, les géodésiques suivantes peuvent longer $\gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}$, et créer les mêmes cas pathologiques que celui illustré sur la Fig. 2.3(g). Enfin, le CGM ne permet pas d'intégrer des termes région [Chan and Vese, 2001, Kim et al., 2005, Michailovich et al., 2007] dans l'énergie.

Nous souhaitons avoir une reconstruction de contour robuste au placement des germes, et indépendante du germe de départ considéré (l'ordre dans le sens horaire étant toutefois respecté). De plus, la génération du contour final doit pouvoir prendre en compte un critère région (homogénéité, distributions de couleurs différentes entre l'objet et le fond, etc). Nous avons pour cela développé une approche basée sur les chemins, dits *admissibles*.

2.2.4 Combinaison de chemins admissibles basés sur les points-selles

Cette contribution a donné lieu aux publications [BMVC13, IJCV15]. Pour chaque couple de germes consécutifs $(\mathbf{v}_i, \mathbf{v}_{i+1})$, plutôt que de n'extraire que le plus court chemin $\gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}$, nous construisons un ensem-

ble \mathcal{A}_i de K_i chemins pertinents ou *admissibles*,

$$\mathcal{A}_i = \{\gamma_{i,j}\}_{1 \leq j \leq K_i}.$$

On remarquera que les chemins γ sont désormais indicés différemment de ceux de l'Eq. (2.6), dans la mesure où $\gamma_{i,j}$ représente le $j^{\text{ème}}$ chemin partant du germe \mathbf{v}_i . Le chemin $\gamma_{i,1}$ est la géodésique, tandis que les chemins suivants sont construits à l'aide d'une approche originale, qui exploite les *points-selles* de la carte de distance géodésique.

L'Eq (2.7) formalise l'action propagée par Fast Marching, à partir de la source a . Lorsque l'action est propagée simultanément à partir de deux sources a et b , les deux fronts se rencontrent aux points-selles de la *carte d'action combinée* $U_{a,b}$. En faisant l'analogie entre l'action combinée et l'altitude en zone montagneuse, les points-selles sont les points les plus hauts des différentes routes allant de a à b (les cols). Si l'on décrit l'ensemble des points situés à égale distance géodésique de a et de b ,

$$\{x \mid U_a(x) = U_b(x)\},$$

par une courbe $m_{a,b}$, dite *courbe médiane*, les points-selles sont les minima locaux de $U_{a,b}$ le long de $m_{a,b}$. En extrayant les K_i points-selles les plus bas, et en effectuant une descente de gradient de part et d'autre de chacun de ces points-selles, nous générerons les K_i chemins admissibles allant de \mathbf{v}_i à \mathbf{v}_{i+1} . Un exemple de carte d'action combinée, le potentiel dont elle est issue et les chemins admissibles construits à partir de cette carte sont illustrés sur la Fig. 2.4.

Une fois les ensembles de chemins admissibles \mathcal{A}_i générés pour tous les germes, le contour final est construit en sélectionnant la meilleure combinaison (en choisissant un chemin dans chaque ensemble \mathcal{A}_i). La meilleure combinaison est celle minimisant l'énergie, somme pondérée d'un terme garantissant que le contour est simple, d'un terme pénalisant le potentiel le long du contour et d'un terme basé sur les distributions des couleurs dans les régions à l'intérieur et à l'extérieur du contour,

$$E[\Gamma] = E_{\text{simplicity}}[\Gamma] + \omega_{\text{edge}} E_{\text{edge}}[\Gamma] + \omega_{\text{region}} E_{\text{region}}[\Gamma]. \quad (2.9)$$

où Γ est une concaténation des chemins choisis, comme dans l'Eq. (2.8). Une recherche exhaustive de la meilleure combinaison aurait une complexité exponentielle en $O(K_{\max}^n)$, avec $K_{\max} = \max_{1 \leq i \leq n} K_i$. Nous proposons une heuristique, basée sur des recherches locales, de complexité $O(n^2 K_{\max})$, permettant d'obtenir des résultats comparables avec une complexité moindre.

2.3 L'objet en mouvement : segmentation vidéo basée sur les modèles de fond

2.3.1 Introduction

Nous prenons ici en compte la dimension temporelle, et abordons la segmentation de plusieurs objets dans des vidéos. Parmi les applications possibles, on peut citer la vidéo-surveillance ou l'estimation de la circulation routière. Dans ce contexte, un objet est n'importe quelle entité en mouvement. La forme et l'apparence des objets varient en général beaucoup au cours d'une vidéo, ceci étant dû aux déformations non-rigides, changements d'éclairage ou occultations, ce qui pose des défis importants en segmentation vidéo.

Nous résumons ici deux contributions, basées chacunes sur leur propre modélisation du fond. La première résulte d'une collaboration avec Jean-Loïc Rose, à l'époque ATER au LIRIS. Elle étend, entre autres, des aspects développés pendant ma thèse, publiés à l'origine dans [CVIU09]. La tâche de segmentation spatiotemporelle est réalisée sur toute la vidéo, dans laquelle le premier *frame* est segmenté. Ces travaux font abstraction de la détection, dans la mesure où un nouvel objet qui apparaîtrait en cours de vidéo, absent du

premier *frame*, ne serait pas pris en compte. Etant donnée une segmentation à l'instant t , l'objectif est de déterminer la segmentation à l'instant $t + 1$.

La deuxième contribution est liée à l'encadrement de la thèse d'Imtiaz Ali, soutenue en 2012. Elle porte sur une méthode de soustraction de fond, qui intègre la détection et la segmentation. Comme dans la plupart des algorithmes de soustraction de fond [Stauffer and Grimson, 2000, Elgammal et al., 2002], la détection est réalisée en mesurant l'écart entre l'apparence locale et un modèle de fond de référence. Le pixel est étiqueté "objet" si l'écart dépasse un certain seuil. Les objets sont construits par extraction des composantes connexes de pixels objet. Il n'y a donc pas de suivi proprement dit, contrairement à la première contribution. Le modèle de fond développé est dédié aux fonds présentant des mouvements périodiques locaux, qui forment des textures spatiotemporelles. Ces motifs sont caractérisés à l'aide de la transformée de Fourier locale.

Avant de résumer les deux contributions, nous en formalisons les aspects communs. La variable temps est ajoutée à la donnée en entrée définie auparavant dans l'Eq. (2.2). La vidéo est une séquence d'images couleur⁶ $(\mathbf{f}_t)_{t=1 \dots T}$, avec T *frames*, telle qu'à chaque instant $t = 1 \dots T$,

$$\begin{aligned} \mathbf{f}_t : \mathcal{D} \subset \mathbb{R}^2 &\longrightarrow \mathcal{Q} \subset \mathbb{R}^3 \\ \mathbf{f}_t(\mathbf{x}) &= [f_{t,1}(\mathbf{x}) \ f_{t,2}(\mathbf{x}) \ f_{t,3}(\mathbf{x})]^\top. \end{aligned}$$

Nous formulons le problème de segmentation vidéo comme la recherche d'un étiquetage de chaque *frame* t en, au plus, $n + 1$ régions, où l'étiquette 0 correspond au fond, et les $i = 1 \dots n$ autres étiquettes correspondent aux n objets. L'étiquetage est écrit comme une séquence de fonctions $(\phi_t)_{t=1 \dots T}$ à valeurs entières,

$$\phi_t : \mathcal{D} \longrightarrow \{0, \dots, n\}. \quad (2.10)$$

Le fond ou le $i^{\text{ème}}$ objet à l'instant t est $\Omega_t^i = \{\mathbf{x} | \phi_t(\mathbf{x}) = i\} \subset \mathcal{D}$. Dans la forme la plus générale de la tâche de segmentation vidéo, n doit être appréhendé comme le nombre total d'objets distincts dans toute la vidéo. La région Ω_t^i peut être vide (pour $i > 0$), car le $i^{\text{ème}}$ objet peut ne pas être présent à l'instant t . Par ailleurs, elle n'est pas nécessairement simplement connexe, car les objets (très exactement, leurs projections sur le plan image) sont de topologie arbitraire.

2.3.2 Segmentation spatiotemporelle basée sur une perception locale du fond

La méthode décrite ici est une version étendue de travaux publiés dans [BMVC11, ICIP11]. Son point de départ est un modèle probabiliste, causal en temps. Ainsi, pour déterminer l'étiquetage ϕ_{t+1} , on dispose des observations précédentes et courante $(\mathbf{f}_s)_{1 \leq s \leq t+1}$, ainsi que des étiquetages précédents $(\phi_s)_{1 \leq s \leq t}$. Nous nous limitons à un modèle du premier ordre en temps, l'étiquetage ϕ_{t+1} ne dépendant que de l'étiquetage précédent, et des images courante et précédente. En terme probabiliste, cette dépendance s'écrit

$$\Pr(\phi_{t+1} | \mathbf{f}_{1:t+1}, \phi_{1:t}) = \Pr(\phi_{t+1} | \mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t),$$

où l'indice $1 : t$ correspond à une séquence de 1 jusqu'à t . Dans le cadre Bayésien, on peut montrer que la probabilité *a posteriori* $\Pr(\phi_{t+1} | \mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)$ se factorise en

$$\Pr(\phi_{t+1} | \mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t) = \frac{\Pr(\mathbf{f}_{t+1} | \mathbf{f}_t, \phi_t, \phi_{t+1}) \Pr(\phi_{t+1} | \mathbf{f}_t, \phi_t) \Pr(\mathbf{f}_t, \phi_t)}{\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)},$$

où

- $\Pr(\mathbf{f}_{t+1} | \mathbf{f}_t, \phi_t, \phi_{t+1})$ est la vraisemblance de l'image \mathbf{f}_{t+1} étant donnée l'image précédente \mathbf{f}_t et les étiquetages ϕ_t et ϕ_{t+1}

⁶Les modèles et les raisonnements mathématiques sont décrits avec des images à valeurs vectorielles, la simplification pour les images en niveaux de gris étant triviale

- $\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)$ est l'*a priori* sur l'étiquetage à $t+1$
- $\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)$ est l'*a priori* sur les données observées

Une factorisation similaire se trouve, par exemple, dans [Mansouri, 2002]. La segmentation est formulée comme la recherche de l'étiquetage ϕ_{t+1} conduisant au *Maximum A Posteriori* (MAP). Les facteurs indépendants de ϕ_{t+1} peuvent être ignorés,

$$\begin{aligned}\phi_{t+1}^* &= \operatorname{argmax}_{\phi_{t+1}} \frac{\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)\Pr(\mathbf{f}_t, \phi_t)}{\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)} \\ &= \operatorname{argmax}_{\phi_{t+1}} \Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t).\end{aligned}$$

Le terme d'*a priori* $\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)$ incarne les contraintes de régularité de forme et de mouvement. Nous appliquons dessus une transformation existante [Mansouri, 2002], via laquelle nous obtenons un terme de régularisation classique. Une grande partie du raisonnement mathématique élaboré dans nos travaux se concentre sur le terme de vraisemblance $\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})$. Plusieurs hypothèses permettent de le transformer en une expression calculable, et maximisable.

Cette transformation fait intervenir les distributions de couleurs des différentes régions Ω_t^i . Nous introduisons donc à ce stade une estimation des densités de probabilité des couleurs dans ces régions. Nous utilisons pour cela un estimateur à noyau Gaussien G_σ . La densité dans la $i^{\text{ème}}$ région à l'instant t est estimée par

$$\tilde{p}_t^i(\mathbf{q}) = \frac{1}{|\Omega_t^i|} \int_{\Omega_t^i} G_\sigma(\mathbf{q} - \mathbf{f}_t(\mathbf{x})) \, d\mathbf{x}, \quad (2.11)$$

où \mathcal{Q} est l'espace colorimétrique introduit dans l'Eq. (2.2). Afin d'obtenir une forme calculable du terme de vraisemblance $\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})$, la première étape est l'hypothèse d'indépendance conditionnelle de $\mathbf{f}_{t+1}(\mathbf{x})$ et $\mathbf{f}_{t+1}(\mathbf{y})$, pour tout couple de points (\mathbf{x}, \mathbf{y}) tel que $\mathbf{x} \neq \mathbf{y}$, étant donné les étiquetages à t et $t+1$. Par ailleurs, une hypothèse raisonnable est de considérer que la vraisemblance d'observer la couleur $\mathbf{f}_{t+1}(\mathbf{x})$ dépend de \mathbf{f}_t , de ϕ_t et de l'étiquetage recherché au point \mathbf{x} seulement, ce qui donne

$$\begin{aligned}\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) &= \prod_{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1})d\mathbf{x} \\ &= \prod_{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x}))d\mathbf{x}.\end{aligned} \quad (2.12)$$

Cette vraisemblance incarne nos hypothèses sur la cohérence temporelle qui peut exister entre *frames* consécutifs. La distribution des couleurs au sein des objets, et dans le fond, varie de façon lisse au cours du temps. Ainsi, le terme $\Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x}))$ est lié à la distribution des couleurs à l'instant t dans la région étiquetée $\phi_{t+1}(\mathbf{x})$. En utilisant la densité $\tilde{p}_t^{\phi_{t+1}(\mathbf{x})}$ définie dans l'Eq (2.11), le terme de vraisemblance (2.12) devient

$$\begin{aligned}\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) &= \prod_{n}^{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x})) \\ &= \prod_{i=0}^n \prod_{\{\phi_{t+1}=i\}} \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x}))d\mathbf{x}\end{aligned}$$

La maximisation de la probabilité *a posteriori* est convertie en problème de minimisation,

$$\begin{aligned}\phi_{t+1}^* &= \operatorname{argmin}_{\phi_{t+1}} E_{\text{data}}[\phi_{t+1}] + E_{\text{reg}}[\phi_{t+1}] \\ &= \operatorname{argmin}_{\phi_{t+1}} -\log \Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) - \log \Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t).\end{aligned} \quad (2.13)$$

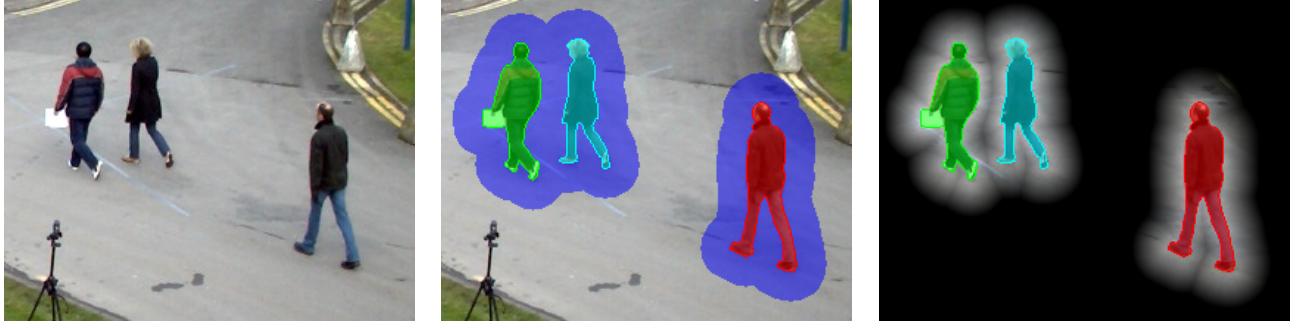


Figure 2.5: Les objets ont leur propre perception locale du fond. (Gauche) image originale, (centre) objets segmentés avec bandes, et (droite) fond "fondu au noir" en fonction de sa contribution aux perceptions locales

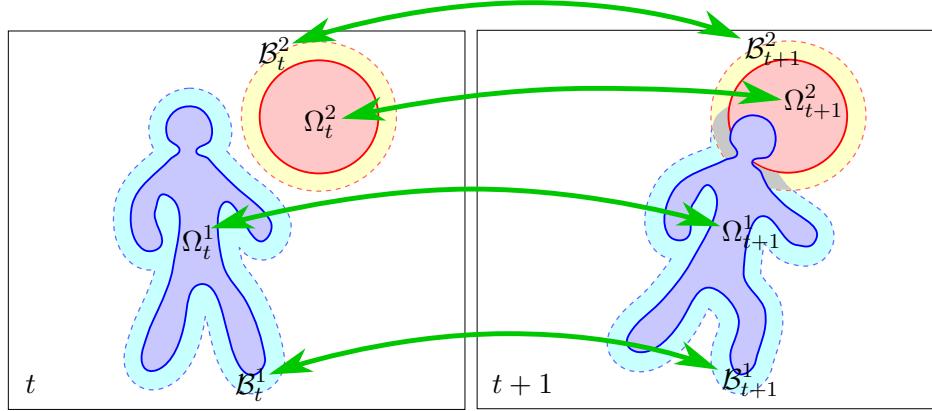


Figure 2.6: Le terme de perception locale lie les distributions de couleur au cours du temps, pour chaque objet et la bande qui l'entoure

En appliquant le logarithme sur la vraisemblance, nous obtenons le terme d'attache aux données $E_{\text{data}}[\phi_{t+1}]$,

$$\begin{aligned} E_{\text{data}}[\phi_{t+1}] &= -\log \Pr(\mathbf{f}_{t+1} | \mathbf{f}_t, \phi_t, \phi_{t+1}) \\ &= -\sum_{i=0}^n \int_{\{\phi_{t+1}=i\}} \log(\tilde{p}_t^i(\mathbf{f}_{t+1}(x))) dx. \end{aligned} \quad (2.14)$$

Dans nos travaux, nous prouvons que rechercher l'étiquetage ϕ_{t+1} qui minimise E_{data} a pour effet de pénaliser l'écart entre les distributions \tilde{p}_{t+1}^i et \tilde{p}_t^i . Par ailleurs, minimiser E_{data} force la distribution \tilde{p}_{t+1}^i à avoir une entropie faible, c-à-d une faible dispersion.

Exploiter la distribution des couleurs, calculée sur le fond dans son entiereté, a un inconvénient majeur. Le problème qui se pose est la grande dispersion de cette distribution, qui survient dans les cas où le fond est encombré de nombreux objets non-suivis, avec des apparences (couleurs et/ou textures) très variées. Dans la plupart des vidéos que l'on est amené à traiter, le fond n'est pas statique. Tout objet non-suivi qui apparaît dans le champ de la caméra est intégré dans le fond. Ces objets "indésirables" modifient la distribution des couleurs du fond, même s'ils sont loin des objets suivis $\{\Omega_t^i\}_{1 \leq i \leq n}$. Ainsi, dû à la dispersion, la densité $\tilde{p}_t^0(q)$ peut devenir faible quel que soit q . On considère un étiquetage à $t+1$ dans lequel un point x appartient au fond, $\phi_{t+1}(x) = 0$, mais proche de l'objet Ω_{t+1}^i ($i \neq 0$). L'étiquette de x sera changée en i à partir du moment où $\tilde{p}_t^i(\mathbf{f}_{t+1}(x)) > \tilde{p}_t^0(\mathbf{f}_{t+1}(x))$. Les régions Ω_{t+1}^i ($i \neq 0$) sont donc très susceptibles d'inclure des pixels de fond et de "fuire" en dehors des objets réels.

Pour remédier à cet inconvénient et extraire des distributions plus représentatives du fond, plus dis-

criminantes pour le suivi, nous introduisons une modélisation du fond basée sur une *perception locale*⁷ du fond. Dans une certaine mesure, nous adaptons la philosophie des contours actifs basés région locaux [Lankton and Tannenbaum, 2008, Brox and Cremers, 2009], à la segmentation spatiotemporelle. On peut également faire le parallèle avec le contexte spatial de [Nguyen et al., 2007].

Dans le modèle de base, les points de fond contribuent de manière égale à la distribution. A l'inverse, dans le modèle de perception locale, nous leur accordons une importance croissante en fonction de leur proximité aux objets, et modifions E_{data} en conséquence. Pour ce faire, nous introduisons tout d'abord d_t^i , la fonction qui associe, à chaque point de fond \mathbf{x} , la distance entre \mathbf{x} et le point le plus proche sur le contour de Ω_t^i ,

$$\begin{aligned} d_t^i & : \Omega_t^0 \longrightarrow \mathbb{R}^+ \\ d_t^i(\mathbf{x}) & = \min_{\mathbf{y} \in \partial \Omega_t^i} \|\mathbf{x} - \mathbf{y}\|. \end{aligned}$$

Soit $\mathcal{B}_t^i \subset \Omega_t^0$ l'ensemble des points de fond dont la distance à Ω_t^i est inférieure ou égale à un seuil fixé B . Pour chaque $i = 1 \dots n$,

$$\mathcal{B}_t^i = \{\mathbf{x} \in \Omega_t^0 \mid d_t^i(\mathbf{x}) \leq B\}.$$

On peut se représenter le domaine \mathcal{B}_t^i comme une bande autour de Ω_t^i , dont l'épaisseur est au plus B . Nous faisons diminuer la contribution des points de fond au fur et à mesure qu'ils s'éloignent des objets, grâce à une fonction de pondération $\psi : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, continue, à support compact (nulle sur $[B, +\infty]$) et non-croissante en fonction de la distance d . Ainsi, nous considérons que les distributions de couleur pertinentes pour la séparation fond/objet sont celles sur les bandes \mathcal{B}_t^i , et nous faisons donc abstraction des données couleur disponibles dans le fond "lointain" $\Omega_t^0 \setminus \bigcup_{i=1}^n \mathcal{B}_t^i$. Les points situés à une distance supérieure à B de tous les objets ne contribuent pas à l'estimation par noyau des densités de couleur, ni au nouveau terme d'attache aux données. Une bande \mathcal{B}_t^i étant défini pour chaque Ω_t^i , chaque objet a sa propre perception locale du fond qui l'entoure, comme illustré sur la Fig 2.5. L'estimation par noyau des densités est pondérée,

$$\tilde{b}_t^i(\mathbf{q}) = \frac{1}{\int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) d\mathbf{x}} \int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) G_\sigma(\mathbf{q} - \mathbf{f}_t(\mathbf{x})) d\mathbf{x}. \quad (2.15)$$

L'utilisation de la fonction de pondération ψ permet les changements graduels des distributions sur les bandes au cours du temps. Lorsque l'objet Ω^i se déplace, les contributions des points de fond "entrent" progressivement dans la distribution \tilde{b}^i .

Le terme de perception locale est construit à partir de E_{data} , défini dans l'Eq. (2.14). On remplace le terme de fond par la somme des log-vraisemblances sur les bandes. Les log-vraisemblances elle-mêmes sont pondérées en fonction de la distance à l'objet considéré. Rappelons qu'à l'instant $t+1$, les bandes \mathcal{B}_{t+1}^i et les distances aux régions d_{t+1}^i dépendent de l'étiquetage à déterminer ϕ_{t+1} . Dans l'écriture du nouveau terme d'attache aux données, nous rendons cette dépendance explicite en utilisant les notations $\mathcal{B}_{t+1}^i(\phi_{t+1})$ et $d_{t+1}^i[\phi_{t+1}; \mathbf{x}]$, ce qui donne

$$\begin{aligned} E_{\text{data}}^{\text{SS}}[\phi_{t+1}] &= \\ \sum_{i=1}^n \left\{ - \int_{\{\phi_{t+1}=i\}} \log \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} - \int_{\mathcal{B}_{t+1}^i(\phi_{t+1})} \psi(d_{t+1}^i[\phi_{t+1}; \mathbf{x}]) \log \tilde{b}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} \right\}. \end{aligned} \quad (2.16)$$

Ce terme d'attache aux données est le *terme de perception locale*, par opposition à celui de l'Eq. (2.14), que nous désignons par terme d'attache aux données global (*full-background*).

Pour calculer effectivement ϕ_{t+1} , en minimisant l'énergie munie du terme (2.16), nous la réécrivons sous forme discrète en espace,

$$E^{\text{SS}}[\phi] = \sum_{\mathbf{x} \in \hat{\mathcal{D}}} U_x^{\text{SS}}[\phi] + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})), \quad (2.17)$$

⁷Dans le manuscrit, nous utilisons l'appellation *short-sight*, que l'on traduirait par "myope". C'est ici un apport, plus qu'un défaut, c'est pourquoi nous lui préférons la dénomination "perception locale"

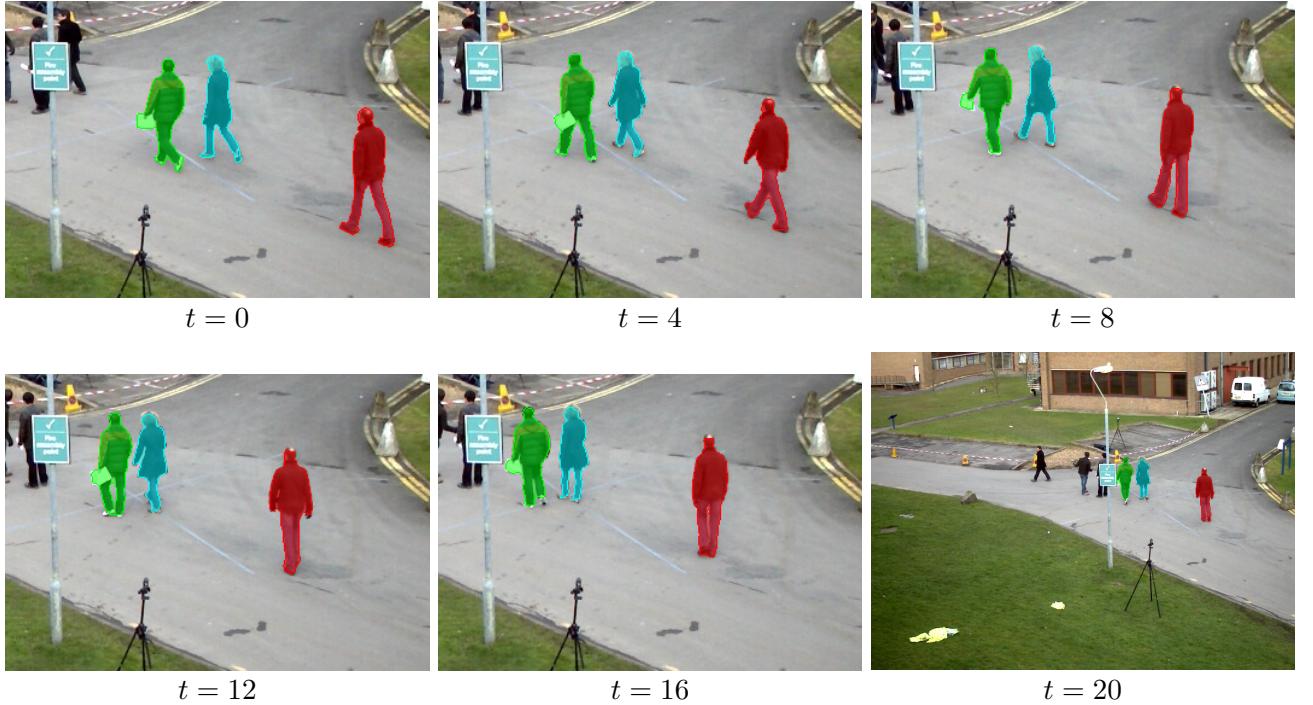


Figure 2.7: Segmentation vidéo multi-objets sur un extrait de la base PETS 2009

où $\hat{\mathcal{D}}$ est la grille de discréétisation. Le coût d'attache aux donnée (perception locale) en un point x de la grille est

$$U_x^{\text{SS}}[\phi] = \begin{cases} \sum_{i=1}^n -\psi(d^i[\phi; x]) \log \tilde{b}_t^i(\mathbf{f}(x)) & \text{si } \phi(x) = 0 \\ -\log \tilde{p}_t^{\phi(x)}(\mathbf{f}(x)) & \text{sinon.} \end{cases} \quad (2.18)$$

Le terme binaire $V(\phi(x), \phi(y))$ est standard. Il est rencontré dans les approches classiques de type *graph-cut*. Le coût U_x^{SS} est différent de celui rencontré dans ce cadre classique. En effet, il dépend de l'étiquetage de toute l'image, et non pas seulement de x . Ainsi, afin de minimiser l'énergie (2.17), nous avons développé une adaptation de l'algorithme $\alpha - \beta$ -swap, initialement décrit dans [Boykov et al., 2001, Section 4]. Un exemple de résultats obtenus avec cet algorithme adapté est illustré sur la Fig 2.7.

2.3.3 Modèle de texture spatiotemporelle pour la détection d'objets

La méthode décrite ici, publiée dans [PRL12], est dédiée à la détection d'objets dans des fonds en mouvement, et plus particulièrement dans des fonds présentant des textures périodiques et stationnaires (vagues à la surface de l'eau, par exemple). Ce choix était justifié par le cadre applicatif de la thèse d'Imtiaz Ali, qui était la détection et le suivi d'objets flottants en rivières. Contrairement aux travaux décrits dans la section précédente, le problème est ici un étiquetage fond/forme, les objets n'étant pas identifiés et suivis individuellement,

$$\phi_t : \mathcal{D} \longrightarrow \{0, 1\}.$$

Nous réalisons la détection des pixels objet par soustraction de fond. Le modèle de fond développé, dit *modèle spatiotemporel spectral*, a deux propriétés majeures. Il est :

- au niveau *pixel*, dans le sens où un modèle de fond de référence est construit en chaque pixel x . Cependant, contrairement aux modèles de fond existants, basés sur les distributions des couleurs [Stauffer and Grimson, 2000, Elgammal et al., 2002], il est construit à partir d'un voisinage spatiotemporel autour de x . On suppose qu'un ou plusieurs mouvement(s) périodique(s) a/ont lieu au

sein de ce voisinage. Les voisinages se recouvrant, il est attendu que les modèles de fond de pixels proches soient similaires.

- *stationnaire.* Considérons un intervalle de temps $[t_0, t_0 + P]$, où la période P est suffisamment grande pour englober tous les mouvements périodiques locaux. Dans le voisinage spatiotemporel centré en un pixel \mathbf{x} , on fait l'hypothèse que les motifs spatiotemporels, formés par les mouvements locaux du fond, sont les mêmes, indépendamment du temps initial t_0 considéré. Ainsi, une fois le modèle de fond appris, il n'y a pas lieu de le mettre à jour au cours de la vidéo. Les changements d'éclairage globaux, qui se traduisent par des translations dans l'espace colorimétrique, peuvent survenir. Il est attendu que la caractérisation de la texture, sans être entièrement invariante à ces translations, soit peu influencée par elles.

On considère en entrée une séquence d'images en niveau de gris⁸ $(f_t)_{t=1 \dots T}$. Chaque image f_t est définie sur la grille discrète $\hat{\mathcal{D}} \subset \mathbb{Z}^2$. Dans cette section, il est plus commode de manipuler le temps au même titre que les dimensions spatiales. Ainsi, étant donnée une position spatiotemporelle $\mathbf{p} = (\mathbf{x}, t)$, nous utilisons la notation $f(\mathbf{p})$ en lieu et place de $f_t(\mathbf{x})$.

Nous proposons de représenter les textures spatiotemporelles, formées par les mouvements périodiques locaux, à l'aide de la Transformée de Fourier Locale Discrète (TFLD) 2D+T. Une fonction de pondération w , dite de *fenêtrage*, est multipliée à l'image avant la transformée, pour éviter les effets de bords [Harris, 1978]. On introduit une largeur spatiale N_s et une largeur temporelle N_t , qui définissent la taille du voisinage sur lequel est calculée la transformée. Ce sont deux entiers impairs, définis par l'utilisateur, qui devront être choisis selon la plus grande période de texture spatiotemporelle à capturer. On suppose que la fonction de pondération w est choisie paire par rapport à chacune de ses variables, et nulle ou négligeable au-delà de $\pm \frac{N_s}{2}$ selon x et y , et de $\pm \frac{N_t}{2}$ selon t .

En un point spatiotemporel donné $\mathbf{p} = (x, y, t)$, la caractérisation locale est calculée sur l'hyperrectangle

$$\mathcal{N}(\mathbf{p}) = \left\{ x - \left\lfloor \frac{N_s}{2} \right\rfloor, \dots, x + \left\lfloor \frac{N_s}{2} \right\rfloor \right\} \times \left\{ y - \left\lfloor \frac{N_s}{2} \right\rfloor, \dots, y + \left\lfloor \frac{N_s}{2} \right\rfloor \right\} \times \left\{ t - \left\lfloor \frac{N_t}{2} \right\rfloor, \dots, t + \left\lfloor \frac{N_t}{2} \right\rfloor \right\}.$$

La TFLD, centrée en \mathbf{p} , pour une fréquence spatiotemporelle \mathbf{u} , est calculée sur $\mathcal{N}(\mathbf{p})$, comme suit:

$$\text{DLFT}\{f\}(\mathbf{p}, \mathbf{u}) = \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} f(\mathbf{p}') w(\mathbf{p}' - \mathbf{p}) \exp(-2\pi i(\mathbf{p}' - \mathbf{p}) \cdot \mathbf{u}). \quad (2.19)$$

Comme fonction de pondération, on choisit la normale de moyenne nulle, et de covariance diagonale, avec comme éléments diagonaux σ_s^{-2} , σ_s^{-2} et σ_t^{-2} . La résolution de la TFLD est telle que, dans le cas unidimensionnel, si le signal d'entrée contient N échantillons, seuls les N premières composantes fréquentielles sont significatives [Harris, 1978]. La TFLD est donc calculée pour chaque vecteur de fréquences $\mathbf{u} = [u_1 \ u_2 \ u_3]^\top \in \mathcal{U}$, avec

$$\mathcal{U} = \{0, \dots, N_s - 1\} \times \{0, \dots, N_s - 1\} \times \{0, \dots, N_t - 1\}.$$

Le *spectre* en un point \mathbf{p} , qui correspond à la séquence des amplitudes des composantes fréquentielles de son voisinage,

$$\mathcal{S}(\mathbf{p}) = (|\text{DLFT}\{f\}(\mathbf{p}, \mathbf{u})|)_{\mathbf{u} \in \mathcal{U}} \in \mathbb{R}^{|\mathcal{U}|},$$

constitue la caractérisation de la texture spatiotemporelle centrée en \mathbf{p} . L'écart entre deux textures spatiotemporelles, centrées respectivement en \mathbf{p}_1 et \mathbf{p}_2 , est mesuré fréquence par fréquence au sens de la distance L^2 ,

$$\|\mathcal{S}(\mathbf{p}_1) - \mathcal{S}(\mathbf{p}_2)\|_2^2 = \sum_{\mathbf{u} \in \mathcal{U}} (|\text{DLFT}\{f\}(\mathbf{p}_1, \mathbf{u})| - |\text{DLFT}\{f\}(\mathbf{p}_2, \mathbf{u})|)^2. \quad (2.20)$$

⁸Les expérimentations menées plaident en faveur d'un rôle négligeable joué par la couleur dans la modélisation de la texture

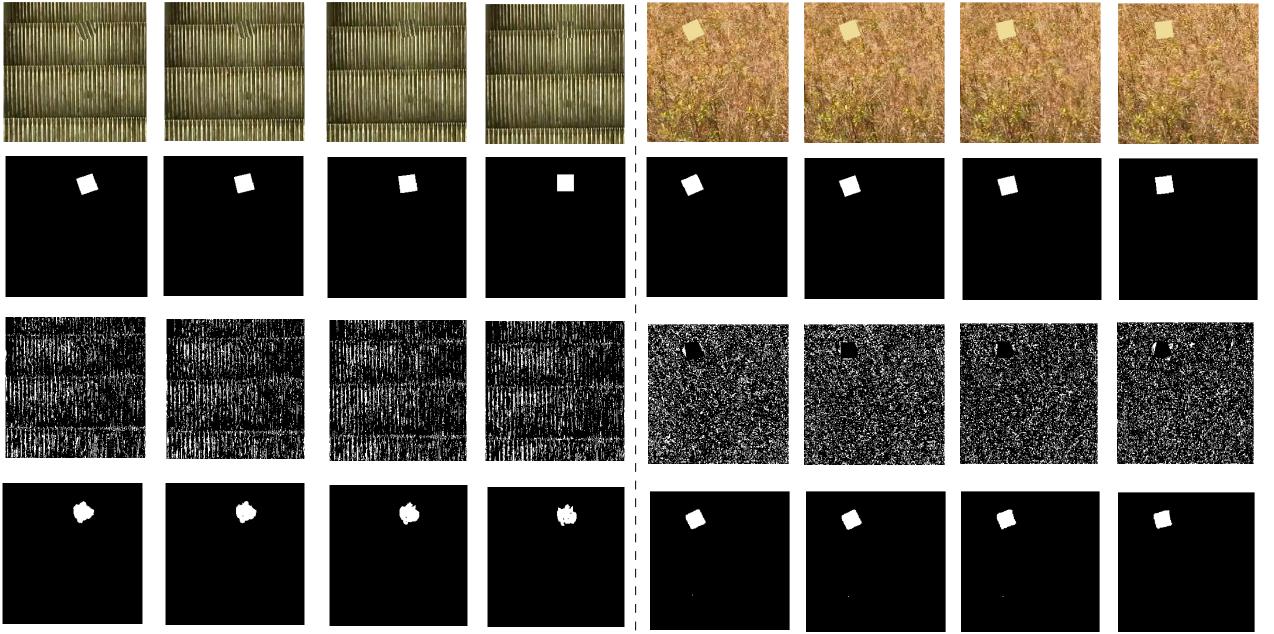


Figure 2.8: *Ligne 1* : Séquences d’images synthétiques (objet texturé, en translation+rotation, sur fond texturé). *Ligne 2* : vérités terrain. *Ligne 3* : Détections obtenues avec [Stauffer and Grimson, 2000]. *Ligne 4* : Détections obtenues avec le modèle de texture spatiotemporelle.

Pour effectuer la détection d’objet dans une vidéo, celle-ci est divisée en deux portions. Les τ premiers *frames*, censés ne contenir aucun objet, sont utilisés pour apprendre le modèle de fond. L’apprentissage consiste simplement à mémoriser K spectres par pixels, sur l’intervalle de temps $[0, \dots, \tau]$,

$$\text{BM}(\mathbf{x}) = \{\mathcal{S}(\mathbf{x}, t_j)\}_{j=1 \dots K}.$$

Dans la deuxième portion de la vidéo ($t > \tau$), les objets sont détectés. Un spectre de test $\mathcal{S}(\mathbf{x}, t)$ est comparé à chaque spectre appris $\text{BM}_j(\mathbf{x}) = \mathcal{S}(\mathbf{x}, t_j)$. Si la distance L^2 (2.20) avec le spectre le plus proche excède un seuil,

$$\min_{j=1 \dots K} \|\mathcal{S}(\mathbf{x}, t) - \text{BM}_j(\mathbf{x})\|_2 > \text{th},$$

le pixel spatiotemporel (\mathbf{x}, t) est étiqueté comme "objet". Notre méthode est comparée à une méthode de sous-traction de fond standard [Stauffer and Grimson, 2000], basée sur une distribution paramétrique des couleurs, le mélange de Gaussiennes (GMM). La Fig. 2.8 illustre la qualité du modèle de texture spatiotemporelle par rapport à un modèle basée couleur.

2.4 L’objet en entrée : appariement de formes basé sur le squelette

2.4.1 Introduction

Une fois l’objet segmenté, il peut être affecté à une catégorie connue, au moyen d’une extraction de descripteurs sur sa forme et/ou son apparence, et d’une classification. Dans cette dernière partie du rapport, nous étudions la classification de formes planes, par appariement entre une forme *requête* (à classifier) et un ensemble de formes prototypes de chaque classe. Le coût de l’appariement correspondant à la dissimilarité entre deux formes, la forme requête est finalement affectée à la classes contenant les K prototypes les moins dissimilaires, au sens des K -plus proches voisins. Une caractérisation pertinente pour l’appariement est difficile à mettre en place, les objets réels devant être reconnus à partir de leur projection 2D sur le plan image. Cette projection varie en fonction de propriétés intrinsèques à l’objet, comme les positions et orientations relatives des parties articulées, mais également en fonction de paramètres extrinsèques, comme la position et l’orientation de la

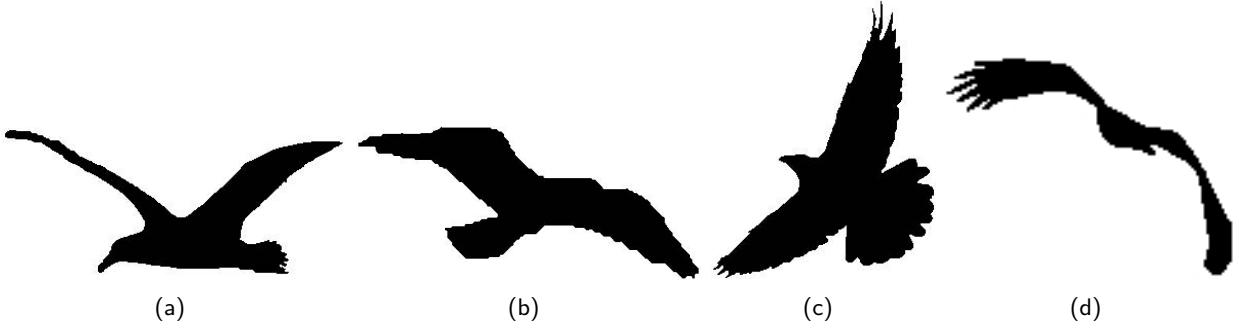


Figure 2.9: Les formes varient selon les déformations articulées et le point de vue [Yang et al., 2016].

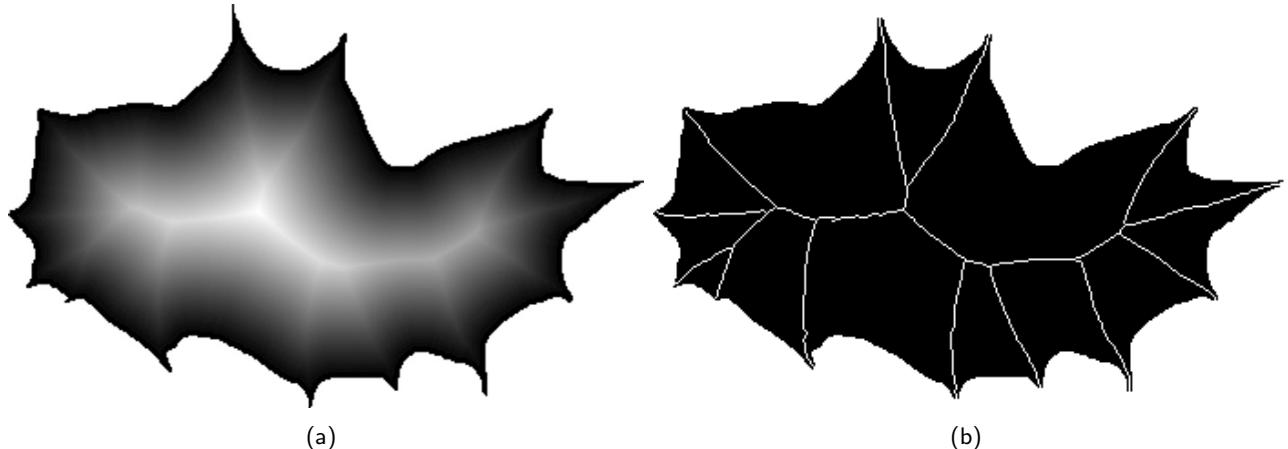


Figure 2.10: (a) Forme plane avec carte de distance Euclidienne et (b) le squelette extrait à partir de cette carte

caméra (voir Fig. 2.9). Une description idéale de forme devrait être, entre autres, discriminante, compacte, invariante, ou tout du moins peu sensible, aux transformations affines (translation, rotation, mise à l'échelle, symétrie, etc) et non-rigides (bruit, mouvements articulés, occultations, etc).

Au regard de ces propriétés, le *squelette*, ou *axe médian* [Blum and Nagel, 1978], est intéressant, car il se prête à la construction de descripteurs avec un certain degré d'invariance aux déformations articulées et à la réorganisation des parties de la forme [Bai and Latecki, 2008, Sebastian and Kimia, 2005]. Les deux contributions résumées dans cette section ont pour origine le co-encadrement de thèse d'Aurélie Leborgne, sur l'appariement de formes basé squelette. Comme point de départ de la première contribution, les branches du squelettes sont localisées sur les crêtes de la carte de distance Euclidienne, comme illustré sur la Fig 2.10. L'extraction de ces crêtes, lorsque la carte de distance est discréétisée sur la grille, nécessite l'emploi de détecteurs. Nous menons une étude théorique de deux détecteurs intervenant dans le calcul du squelette. Nous établissons plusieurs relations mathématiques impliquant ces deux détecteurs, sur plusieurs configurations "type" que l'on peut rencontrer dans les formes planes. Ce travail a été mené en parallèle de la thèse d'Aurélie, et a donné lieu à la publication [JMIV19].

Dans la deuxième contribution, nous développons notre méthode d'appariement de formes, basée sur le *squelette hiérarchique*. Celui-ci est construit via une représentation multi-échelle, générée par des lissages successifs du contour de la forme initiale. Dans le squelette hiérarchique, l'importance des branches est quantifiée et prise en compte dans la méthode d'appariement, de sorte que les correspondances entre branches d'importance proche soient privilégiées. L'appariement exploite, d'une part, les caractéristiques intrinsèques (géométriques) des branches, et, d'autre part, le contexte, incarné par leur position relative aux autres branches. L'appariement est formulé comme un problème d'affectation dans un graphe biparti, résolu à l'aide de l'algorithme hongrois [Kuhn, 1955, Munkres, 1957]. Les publications associées référencées dans la

Section 1.5 sont [ICIP16, RFIA16].

2.4.2 Etude théorique des détecteurs de crête

Soit $\Omega \subset \mathcal{D}$ une forme plane. Sa carte de distance Euclidienne⁹ associe, à tout point $x \in \mathcal{D}$, la distance Euclidienne le séparant du point le plus proche sur le contour $\partial\Omega$. On considère la distance signée, telle que

$$D(x) = \begin{cases} \min_{y \in \partial\Omega} \|y - x\| & \text{si } x \in \Omega \\ -\min_{y \in \partial\Omega} \|y - x\| & \text{si } x \notin \Omega. \end{cases}$$

Le squelette Euclidien S , ou axe médian, est l'ensemble des centres des boules maximales contenues dans Ω [Jalba et al., 2016]. En tout point s du squelette, le rayon de la boule maximale est $D(s)$. Ainsi,

$$S = \{s \in \Omega \mid \exists p, q \in \partial\Omega, p \neq q, \|s - p\| = \|s - q\| = D(s)\}$$

De manière équivalente, le squelette est l'ensemble des points critiques, positifs, de la carte de distance D (le gradient ∇D y est nul ou non-défini). Les branches du squelette correspondent aux crêtes de D . L'extraction de ces crêtes, qui est le point central de cette contribution, a été étudié dans [Siddiqi et al., 2002], au moyen du Flux Sortant Moyen (nous utiliserons l'acronyme AOF, pour *Average Outward Flux*). Dans un domaine $B \subset \mathcal{D}$, le flux sortant d'un champ de vecteur v est la tendance des vecteurs de v à pointer vers l'extérieur de B . Le flux sortant de ∇D sera quasi-nul sur les pentes linéaires de D et fortement négatif sur les crêtes de D . En tout point $x \in \Omega$, [Siddiqi et al., 2002] choisissent de calculer l'AOF sur un disque de rayon r centré en x , notée $B_r(x)$:

$$\begin{aligned} \text{aof}(x, r) &= \frac{1}{|\partial B_r(x)|} \int_{\partial B_r(x)} \nabla D \cdot \mathbf{n} \, ds \\ &= \frac{1}{2\pi} \int_0^{2\pi} \nabla D \left(x + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \, d\theta, \end{aligned}$$

où s est l'abscisse curviligne et \mathbf{n} la normale extérieure, le long du cercle $\partial B_r(x)$.

Par ailleurs, dans [JVCIR15], nous avons introduit la mesure de *ridgeness*¹⁰ basé sur le Laplacien de Gaussienne (LoG) 2D,

$$\Delta G_\sigma(x) = \frac{1}{2\pi\sigma^4} \left(\frac{\|x\|^2}{\sigma^2} - 2 \right) \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right).$$

La *ridgeness* en un point x est définie par convolution de la distance avec la fonction LoG,

$$\text{rdg}(x, \sigma) = -(D * \Delta G_\sigma)(x) = - \int_{\mathbb{R}^2} D(y) \Delta G_\sigma(x - y) dy.$$

La *ridgeness* est quasi-nulle sur les pentes linéaires de D , et élevée sur les crêtes. En comparaison avec l'AOF, la *ridgeness* intègre un lissage, dont l'étendue est contrôlée par l'écart-type σ .

Cette modélisation continue étant établie, la donnée en entrée est une forme discrète $\hat{\Omega} = \Omega \cap \mathbb{Z}^2$, sous-ensemble de la grille $\hat{\mathcal{D}}$. La carte de distance D est calculée grâce à l'algorithme décrit dans [Meijster et al., 2000, Coeurjolly and Montanvert, 2007]. Exploitant le fait que la distance Euclidienne au carré est séparable, l'algorithme est de complexité linéaire en fonction du nombre de pixels $|\hat{\mathcal{D}}|$. Pour extraire le squelette discret, nous utilisons l'algorithme d'amincissement itératif décrit dans [Siddiqi et al., 2002, Algorithm 1]. Il est basé sur la notion de point *simple* [Bertrand and Malandain, 1994], qui désigne un pixel dont la suppression ne modifie pas la topologie de l'objet¹¹. En partant du bord de la forme, cet

⁹ou transformée en distance Euclidienne

¹⁰Nous conservons l'appellation anglo-saxonne dans ce résumé, le néologisme *crêtitude* étant peu appétissant

¹¹On parle alors d'amincissement homotopique, qui conserve la topologie

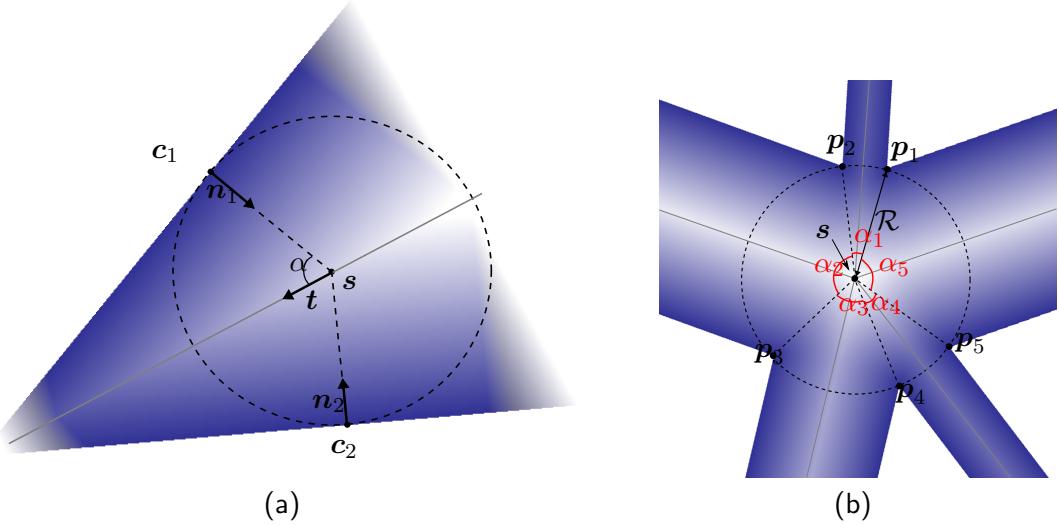


Figure 2.11: Deux configurations "type" de forme parmi celles étudiées : (a) Point de squelette régulier (sur une branche), et (b) Point de jonction

algorithme supprime itérativement les points simples et les points dont l'AOF est supérieur à un seuil négatif th_{aof} . Un exemple de squelette obtenu avec cet algorithme est illustré dans la Fig. 2.10(b). Nous utilisons également une variante de cet algorithme, dans laquelle nous remplaçons le critère basé sur l'AOF par un critère équivalent basé sur la *ridgeness* (les points supprimés sont alors ceux dont la *ridgeness* est inférieure à un seuil positif th_{rdg}). Moyennant un choix approprié du rayon r intervenant dans l'AOF et l'écart-type σ intervenant dans la *ridgeness*, l'algorithme original et sa variante donnent des résultats similaires.

Notre première contribution est motivée par les manques de justification théorique, dans [Siddiqi et al., 2002], sur le choix du seuil th_{aof} . De même, dans [JVCIR15], nous n'avions fourni qu'une approche empirique pour choisir le seuil th_{rdg} . Voici comment nous positionnons notre première contribution : les seuils th_{aof} ou th_{rdg} peuvent être choisis en prenant en compte les valeurs théoriques de aof et rdg sur des configurations de formes spécifiques, qui peuvent être calculées analytiquement en fonction des paramètres r et σ .

Un résultat important de cette contribution est le lien entre aof et rdg . Nous démontrons la relation

$$\text{rdg}(\mathbf{x}, \sigma) = -\frac{1}{\sigma^4} \int_0^\infty \rho^2 \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \text{aof}(\mathbf{x}, \rho) \, d\rho,$$

la *ridgeness* pouvant donc être considérée comme une version de l'AOF, lissée sur plusieurs échelles. Parmi les configurations géométriques "type" que l'on peut rencontrer dans une forme plane, et par conséquent dans le squelette généré, nous illustrons dans ce résumé le cas des points de squelette réguliers, et les jonctions. Le premier cas, représenté dans la Fig. 2.11(a), est créé par deux portions opposées du contour, que l'on approxime localement par des droites. Le deuxième cas correspond à l'intersection de plusieurs branches, illustré dans la Fig. 2.11(b). Entre autres, nous démontrons que l'AOF au voisinage d'un point de squelette régulier s est

$$\text{aof}_{\text{regular}}(\mathbf{x}) = \begin{cases} -\frac{2 \sin \alpha}{\pi r} \sqrt{r^2 - ((\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp)^2} & \text{si } |(\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp| < r \\ 0 & \text{sinon,} \end{cases}$$

où \mathbf{t} est la direction de la branche, et α est l'angle formé par \mathbf{t} et la normale au contour. La *ridgeness* au

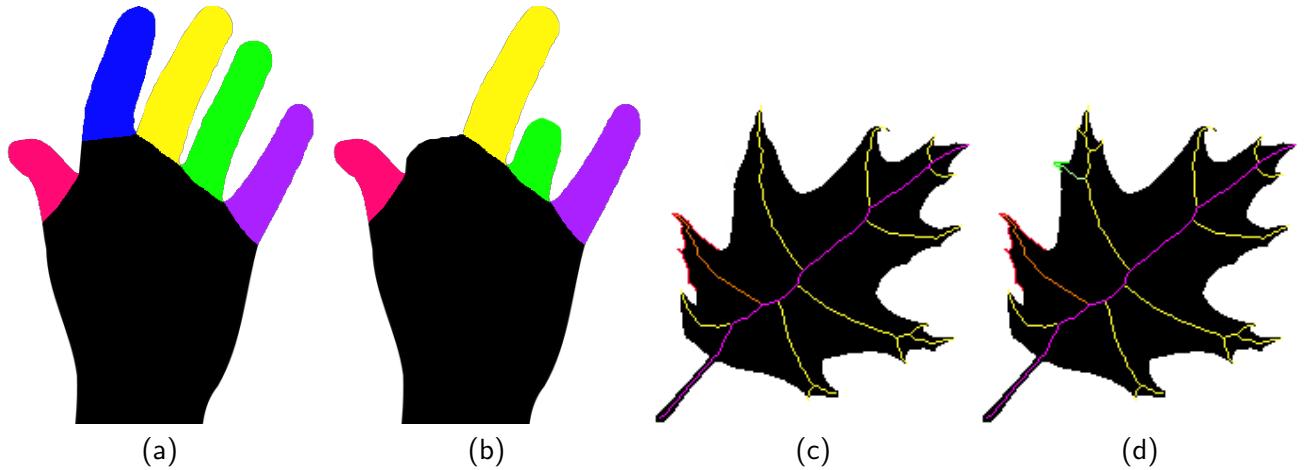


Figure 2.12: Appariement et séquence de branches. (a-b) Les parties de couleur identique seront appariées. (c-d) A chaque partie de la forme correspond une séquence de branches

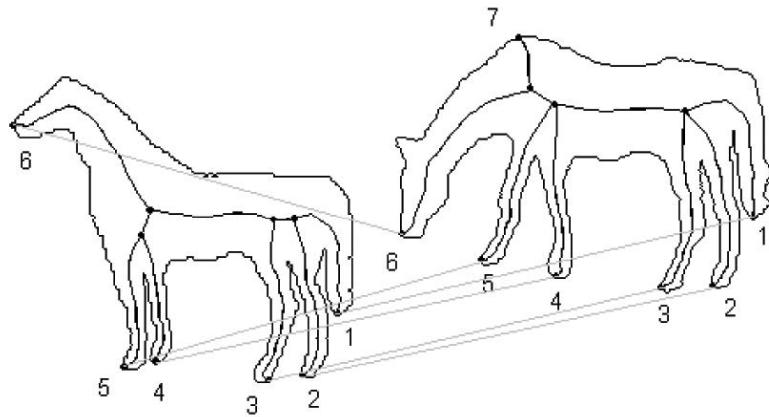


Figure 2.13: Méthode de [Bai and Latecki, 2008] : mise en correspondance des points terminaux du squelette

voisinage de ce même point est

$$\text{rdg}_{\text{regular}}(\mathbf{x}) = \frac{\sqrt{2\pi} \sin \alpha}{\pi \sigma} \exp \left(-\frac{((\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp)^2}{2\sigma^2} \right).$$

Nous établissons des formes analytiques similaires pour les points de jonction, ainsi que d'autres configurations types. A partir de la forme analytique de l'AOF et de la *ridgeness* sur les points de squelette réguliers, nous établissons une relation permettant de choisir les seuils th_{aof} ou th_{rdg} . Ces relations apportent une justification théorique à des choix qui étaient empiriques jusqu'ici.

2.4.3 Appariement de squelettes hiérarchiques

L'objectif de la deuxième contribution est la classification de formes. Nous quantifions la dissimilarité de deux formes par le coût de leur appariement. On souhaite que l'appariement mette en correspondance les parties de formes similaires, comme illustré dans la Fig. 2.12(a-b). L'avantage d'utiliser le squelette, pour l'appariement, est la correspondance naturelle entre ses branches et les parties de la forme.

A l'origine de notre méthode d'appariement se trouve celle développée dans [Bai and Latecki, 2008], basée sur une mise en correspondance des chemins entre points terminaux du squelette, représentée sur la Fig. 2.13. Les plus courts chemins entre tous les points terminaux sont extraits dans les deux squelettes $\hat{S}^{(1)}$ et $\hat{S}^{(2)}$.

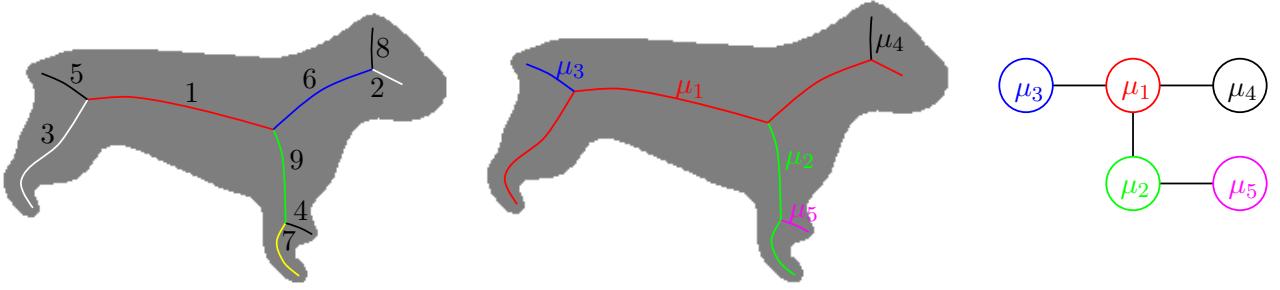


Figure 2.14: Du graphe de squelette au squelette hiérarchique : (gauche) branches, (centre) séquences de branches et (droite) squelette hiérarchique

Ensuite, pour chaque paire de points terminaux $(\mathbf{t}^{(1)}, \mathbf{t}^{(2)}) \in \hat{S}^{(1)} \times \hat{S}^{(2)}$, une dissimilarité, dite *path distance*, est calculée entre chaque chemin partant de $\mathbf{t}^{(1)}$ et chaque chemin partant de $\mathbf{t}^{(2)}$. Cette *path distance* est fonction de la différence, point à point, entre les valeurs de cartes de distance $D^{(1)}$ et $D^{(2)}$ le long des chemins. Ces dissimilarités sont agrégées pour calculer un coût d'appariement entre points terminaux. L'appariement entre les points terminaux de $\hat{S}^{(1)}$ et ceux de $\hat{S}^{(2)}$ est ensuite formulé comme un problème d'affectation dans un graphe biparti, résolu par l'algorithme Hongrois [Kuhn, 1955, Munkres, 1957].

Chaque composante protubérante du contour, même petite, crée une ou plusieurs branche(s), comme illustré dans la Fig. 2.12(c-d). Certaines branches correspondent à des parties importantes de la forme, comme la nervure centrale, alors que d'autres sont issues de détails du contour, qui, dans certains cas, peuvent être considérés comme du bruit. Un appariement robuste doit donc, autant que possible, mettre en correspondance des branches d'importances proches. En développant notre méthode d'appariement de squelette, nous avons souhaité étendre celle de [Bai and Latecki, 2008], en prenant explicitement en compte la hiérarchie des branches.

Le degré d'importance des parties du squelette est plus naturellement formulé au niveau des *séquences de branches* (SB), plutôt que sur les branches. En effet, si l'on construit une représentation multi-échelle de la forme, en appliquant des lissages successifs au contour, les branches issues des protubérances de la forme disparaissent progressivement. Plus la protubérance est petite, plus sa branche correspondante disparaît tôt. Considérons par exemple les branches 1, 3 et 5 de la Fig. 2.14 (gauche). Vraisemblablement, la protubérance de la queue, qui donne lieu à la branche 5, disparaîtra avant la branche 3 au fur et à mesure des lissages. Lorsque la branche 5 disparaît, les branches 1 et 3 n'en forment plus qu'une, et cette nouvelle branche formée est, hiérarchiquement, plus importante que 5. Ces branches, formées au fur et à mesure des lissages, sont les SB. Une définition rigoureuse est la suivante : une branche relie une extrémité ou jonction à une autre extrémité ou jonction, sans jonction intermédiaire. Une SB relie une extrémité à une autre extrémité ou jonction, et peut contenir des jonctions intermédiaires. Notons qu'une SB peut être élémentaire, auquel cas elle n'est issue que d'une seule branche.

Nous utilisons le lissage avec conservation d'aire [Gage, 1986]. D'un point de vue continu, l'évolution conjointe, au cours du temps, du contour de la forme \mathbf{c} , de la carte de distance D et du squelette S peut être formalisée par le système suivant, dans lequel la première équation est une EDP non-linéaire :

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{c}(u, t)}{\partial t} = \left(\kappa(u, t) - \frac{2\pi}{L[\mathbf{c}(\cdot, t)]} \right) \mathbf{n}(u, t), \\ D(\mathbf{x}, t) = \begin{cases} \min_{u \in [0,1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \\ -\min_{u \in [0,1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{out}(\mathbf{c}(\cdot, t)) \end{cases} \\ S(t) = \{ \mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \mid \exists u_1, u_2 \in [0, 1], u_1 \neq u_2, \|\mathbf{x} - \mathbf{c}(u_1, t)\| = \|\mathbf{x} - \mathbf{c}(u_2, t)\| = D(\mathbf{x}, t) \}. \end{array} \right.$$

Nous développons une méthode, dite de squelette déformable, pour mettre à jour S au fur et à mesure des lissages de c . Nous utilisons, comme critère d'arrêt, une condition basée sur le quotient isopérimétrique. Le processus de lissage est stoppé lorsque le contour est quasiment égal à un cercle. Au cours des mises à jour du squelette, les SB sont construites, en gardant en mémoire, pour chacune d'elle, l'itération à laquelle elle disparaît. Nous construisons ainsi le squelette hiérarchique, dans lequel chaque SB μ_a a un degré d'importance $wb(\mu_a) \in]0, 1]$,

$$wb(\mu_a) = \frac{t_a}{t_{\text{end}}},$$

où t_a est l'itération de lissage à laquelle μ_a a été supprimée, et t_{end} l'itération à laquelle le critère d'arrêt a été vérifié. Le squelette hiérarchique est le graphe d'adjacence des SB, illustré sur la Fig. 2.14 (droite), dans lequel chaque SB μ_a a pour attribut une courbe discrète, qui est la séquence des points du squelette initial le long de cette SB, notée $(v_{a,i})_i$.

Considérons maintenant l'objectif final, qui est le calcul d'une dissimilarité entre deux formes planes $\Omega^{(1)}$ et $\Omega^{(2)}$ (l'une étant la forme requête et l'autre le prototype d'une classe). Une fois construits les squelettes hiérarchiques SH⁽¹⁾ et SH⁽²⁾ des deux formes respectives, nous calculons un coût d'appariement entre chaque SB de SH⁽¹⁾ et de SH⁽²⁾. Pour deux SB $\mu_a^{(1)}$ et $\mu_b^{(2)}$, ce coût d'appariement brut, noté $Cb_{a,b}$, fait intervenir entre autres un terme géométrique intrinsèque, dépendant des écarts entre les valeurs de $D^{(1)}$ et $D^{(2)}$ le long des courbes $(v_{a,j}^{(1)})_j$ et $(v_{b,j}^{(2)})_j$. Ces distances sont normalisées par un facteur \aleph , dépendant de l'aire de la forme, pour assurer une certaine invariance à la taille des formes. Le terme géométrique se calcule comme suit,

$$\Xi_r(\mu^{(1)}, \mu^{(2)}) = \frac{1}{|\mu^{(1)}|} \min \left\{ \sum_{i=1}^{|\mu^{(1)}|} \left(\frac{D^{(1)}(v_i^{(1)})}{\aleph^{(1)}} - \frac{D^{(2)}(v_j^{(2)})}{\aleph^{(2)}} \right)^2, \sum_{i=1}^{|\mu^{(1)}|} \left(\frac{D^{(1)}(v_i^{(1)})}{\aleph^{(1)}} - \frac{D^{(2)}(v_k^{(2)})}{\aleph^{(2)}} \right)^2 \right\}$$

avec $j = (i-1) \frac{|\mu^{(2)}|-1}{|\mu^{(1)}|-1} + 1$ et $k = (|\mu^{(1)}|-i-1) \frac{|\mu^{(2)}|-1}{|\mu^{(1)}|-1} + 1$.

L'expression de cette dissimilarité géométrique prend en compte le fait que les longueurs des courbes, notées $|\mu^{(1)}|$ et $|\mu^{(2)}|$, peuvent être différentes. Par ailleurs, les deux sens de parcours sont considérés (les indices j et k avançant dans des directions opposées). Le coût d'appariement brut $Cb_{a,b}$ est également fonction d'un terme de contexte, extrinsèque aux SB $\mu_a^{(1)}$ et $\mu_b^{(2)}$. Ce terme de contexte est calculé en fonction des distances relatives d'une SB aux autres SB du même squelette. Ainsi, deux SB auront un coût brut faible si elles sont similaires géométriquement et si elles occupent des places similaires dans leurs squelettes respectifs. Pour exploiter l'importance wb, nous calculons ensuite un coût pénalisé $C_{a,b}$, qui augmente le coût en fonction de l'écart d'importance des SB, et qui le diminue simultanément si les SB sont importantes,

$$C_{a,b} = Cb_{a,b} \left(\exp \left(2|wb(\mu_a^{(1)}) - wb(\mu_b^{(2)})| \right) + (2 - wb(\mu_a^{(1)}) - wb(\mu_b^{(2)})) \right),$$

afin de privilégier la mise en correspondance des SB importantes, représentatives de la forme générale de $\Omega^{(1)}$ et $\Omega^{(2)}$. Comme dans [Bai and Latecki, 2008], l'appariement est formulé comme un problème d'affectation linéaire dans un graphe biparti, résolu par l'algorithme Hongrois. En terme de coût algorithmique, notre méthode est de complexité $O(V_{\max}^2)$, où V_{\max} est le maximum entre les nombres de SB dans les deux squelettes. En comparaison, la complexité de [Bai and Latecki, 2008] est $O(T_{\max}^4)$, où T_{\max} est le maximum entre les nombres de points terminaux dans les deux squelettes. Notre méthode est donc clairement moins coûteuse. Elle n'atteint malheureusement pas les performances de [Bai and Latecki, 2008] sur la base Kimia216, dont un extrait est représenté sur la Fig. 2.2. [Bai and Latecki, 2008] obtiennent 100% de classifications correctes au rang 1 : la forme la moins dissimilaire est toujours de la même classe que la forme requête. En utilisant le même protocole, nous obtenons 96% de classifications correctes. La mise en balance de la complexité inférieure et de performances moindres est une question en suspens.

Chapter 3

General introduction

Segmenting objects Understanding the semantic content of an image or video may imply a segmentation step, *i.e.* a pixelwise labeling of objects in the scene. In the computer vision/image processing field, an object refers to any entity of interest. On the one hand, segmentation can be a purpose in itself, *e.g.* in medical imaging, where a practitioner wishes to delineate precisely the contour of an anatomical structure. The diagnosis does not require any other computer-based processing, apart from straightforward calculation of descriptors on the segmented shape (length of contour, area, *etc*). On the other hand, segmentation can be part of a higher-level analysis process. In this case, segmentation is one step in a sequence of processings. As an example, the trendy Visual Question Answering (VQA) task can benefit from segmentation, as in [Gan et al., 2017].

Before going into the various types of segmentation, one should also note that high-level image or video understanding tasks can be performed without segmentation. Accurate pixelwise extraction of objects is not always necessary. Considering VQA again, questions can be answered using bounding box-based detections only [Zhang et al., 2018]. As an additional example, several editions of the Pascal VOC challenges [Everingham et al., 2010] have distinct segmentation and detection tasks which computer vision researchers can compete in. Segmentation is not a *de facto* mandatory step. Nevertheless, it cannot be overlooked.

Regardless of the community we are dealing with (image processing, computer vision, biomedical imaging, *etc*), what is known as 'segmentation' may correspond to tasks of various levels of difficulty.

In its most low-level form, segmentation can be understood as aggregating neighboring pixels. Groups are not expected to correspond to visual entities. They are rather small patches of arbitrary shape, containing pixels that share common individual properties (grayscale or color) or neighborhood properties (texture). This processing is often referred to as *oversegmentation*, which is typically the output of superpixel methods [Achanta et al., 2012], watersheds [Couprie et al., 2011], or the graph-based method of [Felzenszwalb and Huttenlocher, 2004].

We let aside oversegmentation and focus on types of segmentation methods targetting a higher semantic level, where, at least, the notion of object and background is present. Suppose we are interested in segmenting cells in the image shown in Fig. 3.1(a). In the case depicted in Fig. 3.1(b), segmentation consists in reconstructing

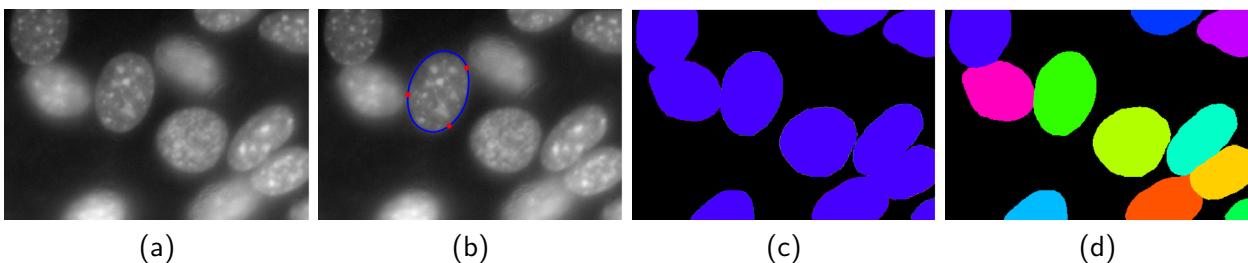


Figure 3.1: Different types of segmentation of a microscopic cell image: (a) initial image, (b) single object with user-provided landmark points, (c) semantic segmentation, (d) instance segmentation

the contour of a chosen cell, with a bit of prior manual intervention. Along the object contour, the user provides a few landmark points, which are automatically linked. Such contour reconstruction could be obtained with the *intelligent scissors* method [Mortensen and Barrett, 1998]¹.

In case there is no user-provided input about the approximate location of a target object, it is assumed that one is interested in all objects in the scene. Thus, all pixels should be labeled, as in Fig. 3.1(c). In the simplest setting, every pixel is labeled either as *object* (or *foreground*) or *background*. Objects can then be extracted as connected components of pixels marked as *foreground*, as in background subtraction methods [Stauffer and Grimson, 1999]. If we go one step further in the semantics, one does not wish to know if a given pixel is within object, but also which category of object it belongs to ('road', 'grass', 'sky', etc). Classes depend on the desired semantic level required by the target application (e.g, is a large 'animal' class sufficient? or do 'cat', 'dog', and 'cow' need to be distinct classes?). In computer vision, this is known as *semantic segmentation*, equally referred to as *scene labeling* or *scene parsing* [Farabet et al., 2013].

Semantic segmentation focuses on object categories, not individual objects. If one is interested in determining which object of which class a pixel belongs to, one is not in presence of a semantic segmentation problem anymore, but rather in front of an *instance segmentation* problem, as depicted in Fig. 3.1(d). Looking at famous benchmarks, the Pascal VOC [Everingham et al., 2010] and MS-COCO [Lin et al., 2014] datasets have separate semantic and instance ground truth annotations. Separating instances is usually considered as a task harder than separating categories. Instance segmentation adds the extra difficulty that regions close to each other, with similar appearances, may belong to different objects. It is especially challenging when an object partially occludes an object of the same category. Although instance segmentation is a particularly difficult task, state-of-the-art methods such as Mask R-CNN [He et al., 2020] achieve impressive results on the popular datasets.

Pixelwise representation of objects is a central concern in this manuscript. Indeed, a large part of the work detailed in subsequent chapters falls within the context of single-object segmentation with user input, evoked by Fig. 3.1(b), or multiple-object segmentation, illustrated in Fig. 3.1(d).

A brief and incomplete survey of trends in segmentation Like other important image processing tasks (classification, detection, tracking, matching, etc), segmentation has been subject to trends over the last decades. These trends are particularly visible if one considers main international venues. For conferences, let us mention CVPR, ICCV, ECCV, BMVC and ICIP. For journals, let us cite *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *International Journal of Computer Vision*, *IEEE Trans. on Image Processing* and *Pattern Recognition*. This paragraph does not pretend to make an exhaustive coverage of object segmentation, but rather to locate my work in a timeline.

While the world of segmentation in the 1970s and 1980s was populated with split-and-merge and region growing methods, the late 1980s, and 1990s, were the age of a large mathematization of the field. The purpose of segmentation began to be formulated as the solution of a minimization problem, in which the objective function and associated constraints were mathematical translations of desired properties of segmented objects (smoothness of contour and appearance, good fit to image data, etc). Following seminal work like the active contour of Kass et al [Kass et al., 1988] the Mumford-Shah piecewise-smooth model [Mumford and Shah, 1989], or the level set framework [Osher and Sethian, 1988], variational methods and Partial Differential Equations (PDEs) were extensively explored for segmentation².

An important trend of the 2000s is the graph cut framework, following the influential work of Boykov et al [Boykov et al., 2001, Boykov and Kolmogorov, 2004, Boykov and Funka-Lea, 2006]. Unlike variational methods which have an initial continuous-space formulation, graph-cut based methods rest upon Markov Random Fields (MRF), which are discrete-space graphical models. Like their continuous counterparts, graph-cut based methods seek a labeling minimizing an energy, which is usually a combination of one or several data-fitting term(s) and regularization term(s).

¹In itself, this type of segmentation does not provide a labeling of all pixels within the target object. However, a labeling is easily obtained by filling the interior of the contour, provided it is closed and does not self-intersect.

²This is true for segmentation only. Variational methods in image processing did not start with active contours. Mainstream variational methods date as far back as 1981, with the optical flow estimation [Horn and Schunck, 1981].

The 2010s era saw the massive arrival of machine learning techniques in image processing and computer vision. Semantic segmentation in a supervised setting (either weakly or fully), which had been initiated at the end of the previous decade, became ubiquitous. Conditional Random Fields (CRFs) [Farabet et al., 2013, Chandra and Kokkinos, 2016] were very popular in the beginning of this decade, extending the bases that had been settled before [He et al., 2004, Shotton et al., 2006]. Structured output, using e.g. random forests [Kontschieder et al., 2014] or CRFs [Vezhnevets et al., 2012], was also a frequently used property. Since 2015, deep convolutional nets have been massively adopted for semantic segmentation. The U-Net architecture [Ronneberger et al., 2015], the Fully Convolutional Network [Long et al., 2015, Shelhamer et al., 2017] or DeepLab [Chen et al., 2018] are notable examples³.

I began to work on segmentation using active contours and level sets in 2004, when I started my PhD. Depending on the type of data-fitting energy term they use, these variational segmentation methods can be classified as edge-based [Kass et al., 1988, Malladi et al., 1995, Xu and Prince, 1998] or region-based [Ronfard, 1994, Chan and Vese, 2001, Li et al., 2008, Brox and Cremers, 2009] (the distinction between edge-based and region-based models will be explained in detail in the introduction of Chapter 4). In retrospect, my PhD started when edge-based variational methods were leaving the front of the stage, in favor of region-based models. Graph-cut methods were also becoming increasingly popular. In this context, the main contributions coming from my PhD work [ICIP07, CVIU09] are within the domain of region-based active contours. An important part of the contributions described in this report, that collaborators and I made in the last decade, draw some inspiration from this work.

Shape matching Let us set aside segmentation as a final purpose for a while and consider the image classification task, where one wishes to determine the category of one or several object(s) in a scene. In a traditional recognition system⁴, segmentation, feature extraction over objects and classification are distinct successive steps. Early papers cited in the survey [Andreopoulos and Tsotsos, 2013, Section 2] are good examples of this classical pipeline.

Once an object is segmented, it can be assigned to a known object category, by means of feature extraction over its shape and/or appearance, and feeding the feature representation to a classifier. In this paragraph, we focus on classifying objects based solely on their shapes, in the sense that the appearance data (color or texture) present in the initial image is not taken into account. It can be referred to as binary shape classification. A famous benchmark for this task is the Kimia216 dataset [Sebastian et al., 2004], from which a subset is shown in Fig. 3.2. In its most basic form, shape classification can be performed by representing the shape with a fixed-size vector of real values, than can be fed as input to any classifier. Otherwise, classification can be addressed through matching. In this context, a query shape is matched to several shapes representative of each class. The query shape is finally assigned to the class containing the best matched shapes.

It is expected that a shape classification method has some invariance - or, at least, robustness - to rigid or affine transformations (rotation, scaling, skewing, etc) and non-rigid deformations that can arise on articulated parts (see the elephant's trunk in Fig. 3.2). Finding representations fulfilling these requirements has been an active topic over the last five decades, from early work such as Fourier descriptors [Zahn and Roskies, 1972], Zernike moments [Teague, 1980], the curvature scale-space [Mokhtarian and Mackworth, 1992] or shape contexts [Belongie et al., 2002] to more recent variable-size graph-based representations [Bai and Latecki, 2008, Macrini et al., 2011b]. For detailed surveys on feature-based shape recognition, the reader may refer to [Loncaric, 1998, Yang et al., 2008].

Shape matching is a side aspect of the work described in this manuscript. With collaborators, we were brought to focus on the skeleton [Blum and Nagel, 1978], which has interesting properties for matching with robustness to articulated deformations.

Common aspects of contributions Two distinct contexts of contributions coexist in this report:

³The invasion of ConvNets in segmentation arrived a few years after the one that had happened in image classification in 2012, when AlexNet [Krizhevsky et al., 2012, Krizhevsky et al., 2017] won the ImageNet ILSVRC 2012 challenge

⁴as opposed to a modern deep architecture taking an image as input and outputting a set of object locations and classes

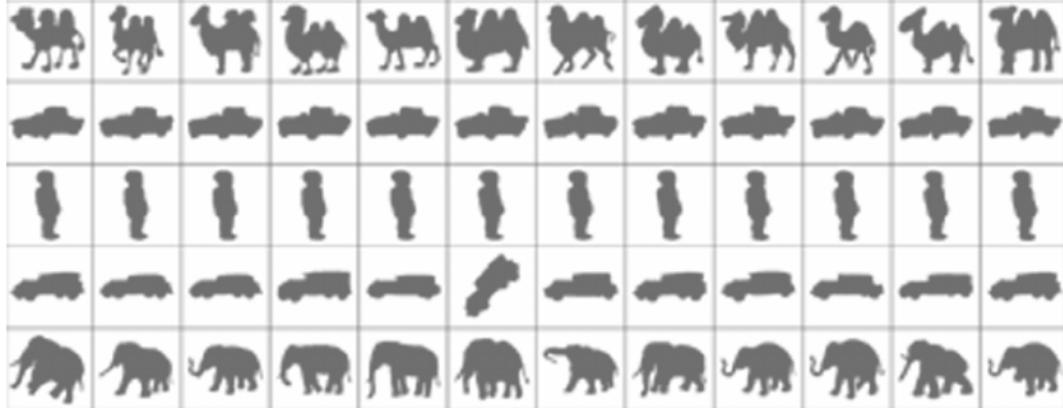


Figure 3.2: A subset of the Kimia216 shape dataset [Sebastian et al., 2004]

- segmentation, where the object is the output
- shape matching, where the object is the input

In addition to the obvious link between the two aspects, which consists in the fact that shape matching would come after segmentation in a traditional recognition system, the transversal aspects of the work in segmentation and shape matching should be highlighted:

- The starting point is always a continuous model, both in space domain and image range. The image is assumed to be a continuously differentiable function. For a grayscale image, we have

$$f : \mathcal{D} \subset \mathbb{R}^2 \longrightarrow \mathbb{R}, \quad (3.1)$$

where \mathcal{D} is the spatial domain. For a vector-valued (color) image,

$$\begin{aligned} f &: \mathcal{D} \subset \mathbb{R}^2 \longrightarrow \mathcal{Q} \subset \mathbb{R}^3 \\ f(\mathbf{x}) &= [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \ f_3(\mathbf{x})]^\top. \end{aligned} \quad (3.2)$$

where \mathcal{Q} is any three-component color space (RGB, YUV, Lab, etc). We deal with one object, denoted by Ω , or several objects $\{\Omega_i\}_{i=1..n}$, which are all subsets of \mathcal{D} . Given one shape Ω , the object boundary $\partial\Omega$ can be described by a set of curves, which are assumed to be continuously differentiable up to some order (at least 2). Energies to minimize and constraints will involve derivatives of f and c . Of course, continuous methods involve a discretization at some stage. The image domain \mathcal{D} will be discretized as a rectangular grid $\hat{\mathcal{D}}$, that can be considered as a subset of \mathbb{Z}^2 . We will handle two types of discretizations for objects:

- A discrete shape containing all pixels (grid nodes) inside the object, making a connected subset of $\hat{\mathcal{D}}$.
- A sampling of the object contour, *i.e.* a polygonal representation, where vertices are elements of \mathbb{R}^2
- All methods perform energy minimization at some stage, where the target variable is a curve or a labeling function. The energy is the mathematical embodiment of the required properties of the variable, *e.g.* we wish the boundary to be smooth, we wish the color distributions inside the object be different than the one of the background, *etc*. In the segmentation methods that are presented, energy minimization - and the object contour or labeling function that is the minimizer of this energy - is the final purpose, whereas it is an intermediate step in the shape matching approach. The energies to minimize are non-convex. We use either simple continuous optimization methods or combinatorial heuristics, that always converge to local minima.

The object as an identified target: segmentation with paths In Chapter 4, single-object segmentation with prior interaction is addressed. The work described here targets applications where the user feels comfortable with a bit of manual work (medical image segmentation by a practitioner, image or video editing, etc). The user provides a set of landmark points along the object boundary. Segmentation consists in reconstructing the object contour by linking successive points using paths, as depicted in Fig. 3.1(b).

The set of methods described here has its roots in the minimal path approach of [Cohen and Kimmel, 1997]. Given two fixed endpoints, the minimal path method consists in linking the two points by a curve, traveling along relevant contours in the image. The curve is found as the global minimizer of a weighted length. As is, the method is not designed to recover closed contours. In addition, linking a series of points consecutively does not guarantee to build a closed contour which is simple, consistent with respect to the purpose of segmentation. Hence, the key idea is to select a combination among a set of possible paths, such that the resulting structure represents a relevant closed curve. This makes up a closed piecewise-smooth curve, over which an edge or region energy functional can be formulated. Instead of considering minimal paths only, we switch to a more general formulation, which we refer to as *admissible* paths [IJCV15]⁵.

This contribution is the most personal part of my work, in the sense that it is not guided by the co-supervision of a PhD student. It is the result of an intense collaboration with Sébastien Bougleux from the GREYC in Caen, and Laurent Cohen from the CEREMADE. Ideas developed in Chapter 4 originates from my postdoc at the CEREMADE, which was about minimal paths.

Objects as moving entities: video object segmentation with background modeling In Chapter 5, we add a time component and tackle multiple object spatiotemporal segmentation⁶. There are two main contributions here, which are both based on a specific modeling of the background.

The first part results from a collaboration with Jean-Loic Rose, who was assistant professor at the LIRIS. It extends elements developed in my PhD, on region-based active contours [CVIU09]. We address the problem of multitarget region tracking within image sequences. The detection part is not addressed in this work, in the sense that a pixelwise segmentation of objects in frame t is available, the purpose being to determine the segmentation at time $t+1$. Starting from a Bayesian interpretation of the segmentation problem - the observed variable being the image sequence and the hidden variable the labeling - we develop an energy-minimization based approach involving color probability distributions of objects and background. The minimized energy models temporal consistency and discrepancy between objects and background.

The second part of Chapter 5 is related to the supervision of PhD student Imtiaz Ali. It integrates both detection and segmentation problems. As in most of background subtraction techniques [Stauffer and Grimson, 1999], detection is performed by measuring the discrepancy between a pixelwise appearance model and a reference background model. The pixel is classified as object if the discrepancy is above a given threshold. Semantic objects are built as connected components of pixels marked as 'object'. The PhD of Imtiaz Ali focused on tracking objects within backgrounds exhibiting local periodic motions, e.g. water waves. Thus, we proposed a novel background modeling method based on frequency, well suited to spatiotemporal texture patterns. The local Fourier Transform was applied to construct a pixel-wise representation of local frequency components.

The object as a given input: skeleton-based shape matching I was involved in the supervision of PhD student Aurélie Leborgne, which led me towards shape matching and skeletonization algorithms. Again, this chapter has two distinct contributions.

Our shape matching algorithm is based on the well-known Euclidean skeleton, or medial axis [Blum, 1967, Blum and Nagel, 1978]. While extracting the skeleton, we were faced to the problem of obtaining a structure which was stable, thin and homotopic to the shape. Skeleton branches correspond to ridges of the Euclidean distance map, i.e. continuous lines of points that are local maxima of

⁵Throughout the manuscript, different conventions are used for references, whether I am a co-author of the paper or not. Co-authored papers are referenced with the journal or conference acronym, and sent back to Section 1.5. External references use the first author's surname and are listed at the end of the manuscript

⁶as opposed, for example, to tracking with bounding boxes

the distance in at least one direction. Extracting these ridges is a non-trivial task on a discrete grid. In order to detect ridges, we studied the average outward flux, used in the Hamilton-Jacobi skeleton [Siddiqi et al., 2002], and also proposed the ridgeness measure [JVCIR15]. In a work parallel to the main research track of the PhD, we established the mathematical relation between these detectors and, extending the work in [Dimitrov et al., 2003], we study various local shape configurations, on which closed-form expressions or approximations of the average outward flux and ridgeness can be derived. This work is described in the first part of Chapter 6.

The second contribution is related to the purpose of the PhD of Aurélie. It tackles shape matching using a hierarchical representation of planar shapes. The novelty is the quantification of importance of skeleton branches, and the use of this importance in the matching process. The intuition guiding the calculation of importance is the need for distinguishing significant object parts from details. Naturally, branches corresponding to significant parts should have a greater impact in the matching than branches arising from smaller parts. This work is described in the second part of Chapter 6.

Work not addressed in this manuscript In Chapter 1, Section 1.4.2 reports the supervision of two PhD students working in machine learning and computer vision: Fabien Baradel and Luong Phat Nguyen. I chose NOT to include the contributions related to their PhDs in the report.

Fabien Baradel worked on deep learning methods for video analysis, with applications in human action recognition and video prediction of physical systems. The PhD was funded by a French/Canadian project, of which I was involved in the submission while at the LIRIS in Lyon. I moved to the LIFAT before the PhD started, so that most of my involvements in the supervision were remote. The major part of scientific supervision was done by Christian Wolf at the LIRIS, who has strong expertise in deep learning and especially in convolutional and recurrent neural networks. From my side, I dove into deep learning during the PhD of Fabien, and did not have the knowledge or distance to provide a leading scientific direction. In addition, Fabien was an incredibly mature and autonomous student, and many ideas developed in his work come from himself. For the sake of scientific honesty, I will not repeat in this report what already appears in his PhD manuscript⁷.

Luong Phat Nguyen is currently investigating deep learning and pattern mining algorithms for video classification. This is an ongoing work, which explores the use of sequential pattern mining for classifying videos containing highly-periodic motion components, like swarms or crowds seen at a large scale, fluids, etc. As opposed to convolutional or recurrent neural networks, which learn feature maps that can be hard to read into, discovering frequent patterns could help interpretability and explainability of the classification. As it is a fresh research track, rather far from the aspects described previously, it does not lend itself to be integrated as a full chapter in this report. The link between this work and my perspective will be made in Chapter 7.

⁷Available at <https://fabienbaradel.github.io>

Chapter 4

The object as an identified target: segmentation with paths

We address single-object segmentation with prior interaction. It is not fully automatic, as the user is expected to provide some manual input for segmenting a given image. This targets applications such as medical image segmentation by a practitioner or image/video editing, for example. Methods described here draw their inspiration from the minimal path approach of [Cohen and Kimmel, 1997], which determines the curve linking two fixed endpoints, as the global minimizer of a cost function.

In this chapter, segmentation is contour-oriented, as it is performed by linking a sequence of user-provided points to reconstruct the object boundary. The minimal path method of [Cohen and Kimmel, 1997], in its native form, was not designed to recover closed contours. Given a series of $n \geq 2$ landmark points, linking these points consecutively with minimal paths generates a closed curve which is not necessarily simple or consistent with respect to the purpose of segmentation. The main idea in the current work is to generate several relevant paths per pair of successive landmark points. We do not consider minimal paths only, and switch to the more general notion of *admissible* paths. We then select the best combination of admissible paths to build a closed curve, which is simple and segmentation-compliant. The combination of admissible path makes up a closed piecewise-smooth curve, over which an edge or region energy functional can be formulated.

The work described in this chapter is the result of a collaboration with Sébastien Bougleux from the GREYC in Caen, and Laurent Cohen from the CEREMADE, started during my postdoc at the CEREMADE. Related publications in Section 1.5 are [SSVM09, VISAPP12, BMVC13, CS14, IJCV15]. Our contributions are based on the minimal paths method, and borrow several elements from the active contour model as well. Sections 4.1 to 4.5 present the background on these curve optimization methods. Our contributions start at Section 4.6.2.

4.1 Segmentation with prior interaction

Methods addressing the problem of object/background segmentation based on energy minimization techniques provide a solid mathematical background, and have proven to find suitable solutions in many practical situations. Among them, a significant amount of methods are interactive, in the sense they require user-defined information, provided prior to the optimization process or during the different steps of segmentation. For instance, in the case of local optimization methods, such as active contours or level sets driven by gradient descent of the energy [Chan and Vese, 2001], the interaction serves for providing an initialization that one hopes to be relatively close to the final solution. We especially focus on prior interactions, which may be roughly classified into two types. They may be region-based, in which case the user quickly draws scribbles that will be used as region seeds. Particular graph cut-based methods [Boykov and Funka-Lea, 2006, Vicente et al., 2008], random walks [Grady, 2006] or the lazy snapping

algorithm [Li et al., 2006] fall into this category. Interactions can also be boundary-based, in which case the user draws a curve, e.g for parametric active contours [Ivins and Porrill, 1995, Kass et al., 1988], or provides a set of landmark points, or *boundary seeds*, that will be linked by curves satisfying, among others, smoothness conditions and fitting to the image data. Significant examples include the intelligent scissors (or *live-wire*) [Mortensen and Barrett, 1998] and their *on-the-fly* extension [Falcão et al., 2000], or the Riverbed algorithm [Miranda et al., 2012] based on the image foresting transform [Falcão et al., 2004].

In this chapter, we focus on segmentation methods with boundary-based interaction, which is provided only before the segmentation process. The sequence of user-provided boundary seeds is a fixed input. Interactive segmentation methods constrained by user-provided seeds very often rely on minimum cost paths, which may be expressed within an either discrete or continuous framework.

In the discrete setting, the image domain is structured as a graph following the grid connectivity. Depending on the model, nodes can be pixels [Mortensen and Barrett, 1998] or pixel seeds [Falcão et al., 2000] - also referred to as *pointels*, for point element - so that arcs are edges between pairs of pixels or pixel boundaries - *linels*, for line element - respectively. In any case, relevant image contours are sought as paths, i.e. ordered loopless sequence of adjacent nodes, minimizing some cost functional. The most natural case is the additive cost function, allowing boundaries to be computed thanks to Dijkstra's shortest path algorithm [Cormen et al., 2009, Chapter 24], like in [Falcão et al., 2000, Mortensen and Barrett, 1998], but more general cost functions were also studied [Miranda et al., 2012].

In the continuous setting, a minimal path is a curve minimizing an energy, which is formulated as an integral of an edge-dependent cost function along the curve [Cohen and Kimmel, 1997]. Cost can be interpolated between grid nodes, and the resulting paths can travel between these nodes, with an arbitrary sampling. As the proposed methods build upon the continuous minimal path method, it is reviewed in detail in Section 4.5.

4.2 Plane curves

In the context of active contours or minimal paths, which are at the core of our contributions, an energy is minimized with respect to a curve. We first need to introduce some basics on plane curves and differential geometry, which are necessary for understanding the energies to minimize. For a more thorough study on the subject, the reader might refer to book chapters [do Carmo, 1976, Chapter 1] and [Raussen, 2008, Chapter 2]. Let us consider a parameterized curve c ,

$$\begin{aligned} c : [0, 1] &\longrightarrow \mathcal{D} \subset \mathbb{R}^2 \\ c(u) &= [x(u) \ y(u)]^\top, \end{aligned}$$

where x and y are at least of class C^2 . In the entire chapter, curves will be assumed to be defined over the normalized range $[0, 1]$. The derivative c' is known as the *velocity vector*. Its norm $\|c'\|$ is the *velocity* or *speed*. The curve is said to be *regular* if its velocity vector never vanishes,

$$c'(u) \neq 0 \quad \forall u \in [0, 1].$$

If the curve is regular, the *unit tangent vector*,

$$t(u) = \frac{c'(u)}{\|c'(u)\|}, \tag{4.1}$$

is defined everywhere.

Orientation of curves requires to define the coordinate system. Our reference is the standard computer left-handed cartesian coordinate system (x -axis rightward, y -axis downward). In this setting, we choose the natural

orientation of curves as the clockwise one (negatively oriented), such that when one travels along the curve, looking forward, the interior of the curve is on the right. If $\mathbf{a} = [a_1 \ a_2]^\top$, the *perp* operator \perp is defined as a $90 - \text{degree}$ clockwise rotation:

$$\mathbf{a}^\perp = \begin{bmatrix} -a_2 \\ a_1 \end{bmatrix}$$

In such case, \mathbf{c}'^\perp points inward, and

$$\mathbf{n}(u) = \frac{\mathbf{c}'(u)^\perp}{\|\mathbf{c}'(u)\|} \quad (4.2)$$

is the *unit inward normal*. The *length* of the curve is

$$L[\mathbf{c}] = \int_{\mathbf{c}} ds = \int_0^1 \|\mathbf{c}'(u)\| du, \quad (4.3)$$

which is a real *functional*, i.e. a mapping of a function to a real value. For functionals, we use notation $[.]$.

The 2D *cross product* $\mathbf{a} \times \mathbf{b}$ is the determinant of the 2×2 matrix $[\mathbf{a}|\mathbf{b}]$. It can be equivalently defined as:

$$\mathbf{a} \times \mathbf{b} = \mathbf{a}^\perp \cdot \mathbf{b}$$

We can use it to write the *signed curvature*, at a given position u :

$$\kappa(u) = \frac{\mathbf{c}' \times \mathbf{c}''}{\|\mathbf{c}'\|^3} = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}. \quad (4.4)$$

The curvature represents the signed amount of change of the direction of the tangent along the curve. It is zero along a straight line, positive on convex parts and negative on concave parts. Since it is the inverse of the radius of the osculating circle, the curvature is, for example, constant ($= 1/r$) along a clockwise-oriented circle of radius r .

The *signed area* is

$$A[\mathbf{c}] = \frac{1}{2} \int_0^1 \mathbf{c} \times \mathbf{c}' du = \frac{1}{2} \int_0^1 xy' - yx' du, \quad (4.5)$$

which comes from Green-Riemann or divergence theorem.

A curve is said to be *simple* if it has no multiple point, i.e. if it does not self-intersect or self-overlap. In case a curve is simple and closed, it splits the image domain \mathcal{D} into two disjoint connected domains, the interior $\text{in}(\mathbf{c}) = \Omega$ and exterior $\text{out}(\mathbf{c})$ ¹. If, in addition, the curve is clockwise-oriented, the signed area is equal to the actual area of Ω .

4.3 Curves as minimizers of energies

4.3.1 Calculus of variations

Segmentation is formulated as the search for the curve minimizing an energy functional, made up of regularization terms for geometrical smoothness, and data terms relating the model to the data, chosen according to the purpose of segmentation. We restrict ourselves to energies that depend on the curve and its first and second order derivatives. The energy functional is an integral along the curve, with the general form

$$E[\mathbf{c}] = \int_0^1 h(\mathbf{c}(u), \mathbf{c}'(u), \mathbf{c}''(u)) du. \quad (4.6)$$

¹Incidentally, if $\partial\Omega$ is the boundary of domain Ω , \mathbf{c} is a parameterization of $\partial\Omega$

where $h : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ is a general function, continuously differentiable with respect to each of its three parameters. It is assumed to gather regularization and data-fitting energies. If \mathbf{c} is closed, integrals over $\text{in}(\mathbf{c})$ or $\text{out}(\mathbf{c})$ can be written under the form (4.6), using Green-Riemann or divergence theorem.

Hereafter, we rely on elements of calculus of variations [Forsyth, 1960, Chapter 1]. Energy $E[\mathbf{c}]$ attains a local minimum at \mathbf{c} if, for any small and smooth variation $\boldsymbol{\eta} : [0, 1] \rightarrow \mathbb{R}^2$,

$$E[\mathbf{c}] < E[\mathbf{c} + \boldsymbol{\eta}].$$

A necessary condition for $E[\mathbf{c}]$ being a local extremum is that its first variation vanishes,

$$\frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \Big|_{\epsilon=0} = 0. \quad (4.7)$$

Using the chain rule, we have

$$\frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \Big|_{\epsilon=0} = \int_0^1 \boldsymbol{\eta} \cdot \frac{\partial h}{\partial \mathbf{c}} + \boldsymbol{\eta}' \cdot \frac{\partial h}{\partial \mathbf{c}'} + \boldsymbol{\eta}'' \cdot \frac{\partial h}{\partial \mathbf{c}''} du.$$

If \mathbf{c} is closed (or if \mathbf{c} has fixed endpoints, assuming $\boldsymbol{\eta}(0) = \boldsymbol{\eta}(1) = 0$), integration by parts yields

$$\frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \Big|_{\epsilon=0} = \int_0^1 \boldsymbol{\eta} \cdot \left(\frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} + \frac{d^2}{du^2} \left\{ \frac{\partial h}{\partial \mathbf{c}''} \right\} \right) du.$$

The expression in factor of $\boldsymbol{\eta}$ is known as the functional derivative of E with respect to \mathbf{c} , at a given point u , that is denoted by

$$\frac{\delta E[\mathbf{c}]}{\delta \mathbf{c}(u)} = \frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} + \frac{d^2}{du^2} \left\{ \frac{\partial h}{\partial \mathbf{c}''} \right\}. \quad (4.8)$$

According to the fundamental lemma of calculus of variations, Eq. (4.7) is satisfied for any smooth function $\boldsymbol{\eta}$ if the functional derivative vanishes at every point u , which leads to a Partial Differential Equation (PDE) referred to as Euler-Lagrange equation:

$$\frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} + \frac{d^2}{du^2} \left\{ \frac{\partial h}{\partial \mathbf{c}''} \right\} = \mathbf{0}. \quad (4.9)$$

The Euler-Lagrange equation is cast into an evolution equation by adding a time variable. Minimizing energy (4.6) amounts to find the stationary solution of the following problem, made up of an evolution PDE and initial condition:

$$\begin{cases} \frac{\partial \mathbf{c}(u, t)}{\partial t} = -\frac{\delta E[\mathbf{c}]}{\delta \mathbf{c}(u, t)} \\ \mathbf{c}(\cdot, 0) = \mathbf{c}^{(0)}(\cdot) \end{cases} \quad (4.10)$$

4.3.2 Numerical solving

The classical procedure of variational methods is to minimize the energy by numerically solving Eq. (4.9) using a specific discretization scheme. In many cases, there is not a unique solution to Eq. (4.9). A sequence $(\mathbf{c}^{(t)})_t$ will converge to a local minimum of $E[\mathbf{c}]$, starting from an initial curve $\mathbf{c}^{(0)}$. If evolution equation (4.10) is discretized using the explicit (forward) Euler method, it boils down to minimize $E[\mathbf{c}]$ using gradient descent:

$$\mathbf{c}^{(t+1)}(u) = \mathbf{c}^{(t)}(u) - \delta t \frac{\delta E[\mathbf{c}]}{\delta \mathbf{c}(u)}, \quad (4.11)$$

where δt is the time step.

Curve sampling: snakes. If it is chosen to work with an explicit representation of the curve, as in the original active contour or *snake* [Kass et al., 1988, Ivins and Porrill, 1995, Jacob et al., 2004], the curve is sampled as a polygon. The functional derivative in the right-hand side of Eq. (4.11) involves at least first and second-order derivatives of c , which are discretized using centered finite difference schemes,

$$\begin{aligned} c'(u) &= \frac{c(u + \delta u) - c(u - \delta u)}{2\delta u} + O(\delta u^2) \\ c''(u) &= \frac{c(u + \delta u) - 2c(u) + c(u - \delta u)}{\delta u^2} + O(\delta u^2). \end{aligned} \quad (4.12)$$

Hence, gradient descent deforms the polygon by direct modifications of vertex coordinates. A drawback of this method which is usually raised is that the curve needs to be resampled as the deformation progresses. Moreover, a polygon is a discretization of a single closed contour. If one wishes to adapt to topological changes during the deformation, in case the target object is not a simply connected component of pixels, additional techniques for detection of self-crossings must be implemented beside the energy minimization algorithm [Perera et al., 2000, Nakhmani and Tannenbaum, 2012].

Implicit curves: level sets. Conversely, implicit implementations, based on the level set framework [Osher and Sethian, 1988, Sethian, 1999], handle the evolving curve as the zero level set of a smooth function, defined on the same domain as the image:

$$\psi : \mathcal{D} \longrightarrow \mathbb{R}.$$

The level set approach is often chosen for its natural handling of topological changes and intuitive extensibility to higher dimensions. If one chooses the convention that ψ is positive inside and negative outside, we have

$$\begin{aligned} \Omega &= \{\mathbf{x} \mid \psi(\mathbf{x}) \geq 0\} \\ \mathcal{D} \setminus \Omega &= \{\mathbf{x} \mid \psi(\mathbf{x}) < 0\}. \end{aligned}$$

A natural choice is to constrain ψ to be the signed-distance function to Ω :

$$\psi(\mathbf{x}) = \begin{cases} \min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\| & \text{if } \mathbf{x} \in \Omega \\ -\min_{\mathbf{y} \in \partial\Omega} \|\mathbf{x} - \mathbf{y}\| & \text{otherwise} \end{cases}$$

In this case, ψ verifies the identity $\|\nabla\psi\| = 1$ almost everywhere (except on critical points, where $\nabla\psi$ is undefined).

There are two different ways to employ level sets in the mathematical derivation. On the one hand, the Euler-Lagrange equation (4.9) is derived from the parametric curve representation c - i.e. the first variation is calculated on c - and subsequently cast into the level set formulation, as in [Malladi et al., 1995, Caselles et al., 1997]. On the other hand, the level set formulation can be introduced earlier, as soon as the energy is set. In this case, Eq. (4.6) is cast into the level set formulation, and calculus of variations is conducted over ψ , as in [Chan and Vese, 2001, Kim et al., 2005]. In both cases, the gradient descent equation involves at least first and second-order derivatives of ψ . Discretization on a cartesian grid using centered finite differences gives

$$\begin{aligned} \frac{\partial\psi}{\partial x}(x, y) &= \frac{\psi(x + \delta x, y) - \psi(x - \delta x, y)}{2\delta x} + O(\delta x^2) \\ \frac{\partial^2\psi}{\partial x^2}(x, y) &= \frac{\psi(x + \delta x, y) - 2\psi(x, y) + \psi(x - \delta x, y)}{\delta x^2} + O(\delta x^2), \end{aligned} \quad (4.13)$$

and similarly for derivatives with respect to y . The discretization of the mixed derivative is

$$\frac{\partial^2\psi}{\partial x\partial y}(x, y) = \frac{\psi(x + \delta x, y + \delta y) - \psi(x - \delta x, y + \delta y) - \psi(x + \delta x, y - \delta y) + \psi(x - \delta x, y - \delta y)}{4\delta x\delta y} + O(\delta x^2, \delta y^2),$$

but other discretization schemes, like the upwind scheme, can be used, as in [Chan and Vese, 2001]. Spatial steps are usually chosen as $\delta x = \delta y = 1$.

4.4 Energies

So far, we have addressed variational segmentation methods considering a general energy functional (4.6). The energy is usually a weighted sum of one or several regularization term(s) and data terms(s). We now review different common terms that were used in the literature. In what follows, curve position \mathbf{c} and its derivatives, normal \mathbf{n} and curvature κ are considered at u . For concision of notation, (u) is omitted in most of the equations.

4.4.1 Regularization term

Regularization is independent of input image f , and ensures that the obtained object contours are smooth. It is very common to consider the length of the curve, defined in Eq. (4.3), as the regularization term (see for example [Chan and Vese, 2001]). Its functional derivative is

$$\frac{\delta L[\mathbf{c}]}{\delta \mathbf{c}(u)} = -\kappa \|\mathbf{c}'\| \mathbf{n}, \quad (4.14)$$

which, when integrated into the evolution flow (4.10), induces a curvature-based shortening flow. The regularization term of the initial snake model [Kass et al., 1988] is made up of two terms penalizing stretching and bending of the curve, respectively,

$$E_{\text{reg}}^{\text{snake}}[\mathbf{c}] = \int_0^1 \frac{\alpha}{2} \|\mathbf{c}'\|^2 + \frac{\beta}{2} \|\mathbf{c}''\|^2 du,$$

and has functional derivative

$$\frac{\delta E_{\text{reg}}^{\text{snake}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = -\alpha \frac{\partial^2 \mathbf{c}}{\partial u^2} + \beta \frac{\partial^4 \mathbf{c}}{\partial u^4}.$$

It is common to add an area term, negatively weighted, to compensate the natural shrinking effect caused by length-based term [Cohen, 1991, Chan and Vese, 2001]. The area (4.5) has functional derivative

$$\frac{\delta A[\mathbf{c}]}{\delta \mathbf{c}(u)} = -\|\mathbf{c}'\| \mathbf{n},$$

which is proven, for example, in [Zhu and Yuille, 1996]. When integrated into the evolution flow (4.10), it adds an inflation component to the curve, which is referred to as the *balloon force* in [Cohen, 1991, Cohen and Cohen, 1993].

4.4.2 Data term

Edge-based data term

The data term depends on image intensity (or grayscale) f , and ensures that the curve fits to the contour of the target object. The first active contour models were mainly edge-based, where one wishes to minimize an edge penalty function along the curve. In its simplest form, the edge penalty is a decreasing function of the image gradient norm. The data term of the initial snake model [Kass et al., 1988] is

$$E_{\text{data}}^{\text{snake}}[\mathbf{c}] = \int_0^1 -\|\nabla f(\mathbf{c})\| du, \quad (4.15)$$

and has functional derivative

$$\frac{\delta E_{\text{data}}^{\text{snake}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = -\nabla \|\nabla f\|(\mathbf{c}),$$

which, in evolution flow (4.10), makes each curve point move towards the nearest edge. Instead of computing the gradient with finite differences, which generates noisy edges, contours can instead be extracted at a given scale σ using Gaussian derivatives. Then, a natural edge penalty is

$$g(\mathbf{x}) = \frac{1}{1 + \|(\mathbf{f} * \nabla G_\sigma)(\mathbf{x})\|}, \quad (4.16)$$

where G_σ is the 2D Gaussian with zero mean and diagonal covariance matrix $\sigma^2 \mathbf{I}$, and $*$ is the convolution product. In case the image is vector-valued, as defined in Eq. (3.2), its first-order derivatives are gathered in $\nabla \mathbf{f}$, the 3×2 Jacobian matrix

$$\nabla \mathbf{f} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} \end{bmatrix}.$$

It is then natural to use the Frobenius norm to formulate the edge penalty for vector-valued images²,

$$g(\mathbf{x}) = \frac{1}{1 + \|(\mathbf{f} * \nabla G_\sigma)(\mathbf{x})\|_{\text{F}}}.$$

The edge penalty function (4.16) is used in the geodesic active contour [Caselles et al., 1997],

$$E_{\text{data}}^{\text{geodesic}}[\mathbf{c}] = \int_0^1 g(\mathbf{c}) \|\mathbf{c}'\| \text{d}u. \quad (4.17)$$

Unlike in the data term of the snake model (4.15), the edge penalty is multiplied by the curve velocity, which makes the edge term intrinsic, *i.e.* independent from the parameterization. It is shown in [Caselles et al., 1997] that energy (4.17) has functional derivative

$$\frac{\delta E_{\text{data}}^{\text{geodesic}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = (\nabla g(\mathbf{c}) \cdot \mathbf{n} - g(\mathbf{c})\kappa) \|\mathbf{c}'\| \mathbf{n},$$

which induces a displacement normal to the curve. It includes a component driving curve points towards edges, as well as a curvature-dependent regularization component.

Region-based data term

Since they consider only local boundaries, edge-based models are relatively blind, in the sense they are unable to reach boundaries if their initial location is far from them. Moreover, they can get trapped in spurious edges generated by noise. Region-based data terms were proposed as a mean to overcome these limitations. Unlike edge-based terms which integrate an edge penalty function along \mathbf{c} , region-based data terms integrate image penalty functions on the entire domains $\text{in}(\mathbf{c})$ and $\text{out}(\mathbf{c})$.

The most used region-based data term is the two-phase piecewise-constant model [Chan and Vese, 2001]. It pushes the curve to split the image domain into two homogeneous regions. Thus, it penalizes intensity deviations inside and outside the curve, with respect to two reference intensities,

$$E_{\text{data}}^{\text{CV}}[\mathbf{c}, k_{\text{in}}, k_{\text{out}}] = \int_{\text{in}(\mathbf{c})} (f(\mathbf{x}) - k_{\text{in}})^2 \text{d}\mathbf{x} + \int_{\text{out}(\mathbf{c})} (f(\mathbf{x}) - k_{\text{out}})^2 \text{d}\mathbf{x} \quad (4.18)$$

²Here, the convolution product $\mathbf{f} * \nabla G_\sigma$ is matrix-valued (same size as the Jacobian $\nabla \mathbf{f}$). It should be understood in the sense of the outer product,

$$(\mathbf{f} * \nabla G_\sigma)(\mathbf{x}) = \int_{\mathbb{R}^2} \mathbf{f}(\mathbf{x} - \mathbf{y}) \otimes \nabla G_\sigma(\mathbf{y}) \text{ d}\mathbf{y}.$$

where k_{in} and k_{out} are reference intensities inside and outside the curve, respectively. In addition to \mathbf{c} , they are variables of the minimization problem. Minimization of $E_{\text{data}}^{\text{CV}}$ is alternatively done with respect to \mathbf{c} and $(k_{\text{in}}, k_{\text{out}})$. As regards \mathbf{c} , it can be proven [Zhu and Yuille, 1996], that the functional derivative of energy (4.18) is

$$\frac{\delta E_{\text{data}}^{\text{CV}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = (f(\mathbf{c}) - k_{\text{in}})^2 - (f(\mathbf{c}) - k_{\text{out}})^2 \mathbf{n}$$

so that the curve moves outward or inward along its normal, depending whether $f(\mathbf{c}(u))$ is closer to k_{in} or k_{out} . Fixing \mathbf{c} and zeroing the derivatives of $E_{\text{data}}^{\text{CV}}$ with respect to k_{in} and k_{out} gives a direct solution to the minimization step. k_{in} and k_{out} are assigned to the average intensities inside and outside the curve, respectively,

$$\begin{aligned} k_{\text{in}} &= \frac{1}{|\text{in}(\mathbf{c})|} \int_{\text{in}(\mathbf{c})} f(\mathbf{x}) d\mathbf{x} \\ k_{\text{out}} &= \frac{1}{|\text{out}(\mathbf{c})|} \int_{\text{out}(\mathbf{c})} f(\mathbf{x}) d\mathbf{x}. \end{aligned}$$

Note that the extension to vector-valued images is trivial. In Eq. (4.18), squared differences should be replaced with $\|\mathbf{f}(\mathbf{x}) - \mathbf{k}_{\text{in}}\|^2$ and $\|\mathbf{f}(\mathbf{x}) - \mathbf{k}_{\text{out}}\|^2$. The Chan-Vese model makes a very constraining assumption on the appearance of the object and background, as it is based on constancy of intensity within them. This condition is rarely met in real-world images. A notable class of region-based data terms alleviating this requirement uses probabilities of intensity/color inside and outside the curve. The assumption made in this context is that the object and background have distinct distributions of intensity or color, and that pixel intensities or colors are independent and identically distributed (i.i.d) in each region. We describe these non-parametric probability-based models for vector-valued images, the simplification for grayscale images being trivial.

For a given color \mathbf{q} in color space $\mathcal{Q} \subset \mathbb{R}^3$, as mentioned after Eq. (3.2), its probability to appear in $\text{in}(\mathbf{c})$ or $\text{out}(\mathbf{c})$, that we denote by

$$\Pr(\mathbf{f}(\mathbf{x}) = \mathbf{q} | \mathbf{x} \in \text{in}(\mathbf{c})) \text{ or } \Pr(\mathbf{f}(\mathbf{x}) = \mathbf{q} | \mathbf{x} \in \text{out}(\mathbf{c})),$$

can be approximated using Gaussian kernel density estimation³. We thus introduce two conditional density estimates \tilde{p}_{in} and \tilde{p}_{out} ⁴

$$\begin{aligned} \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}] &= \frac{1}{|\text{in}(\mathbf{c})|} \int_{\text{in}(\mathbf{c})} G_\sigma(\mathbf{q} - \mathbf{f}(\mathbf{x})) d\mathbf{x} \\ \tilde{p}_{\text{out}}[\mathbf{c}, \mathbf{q}] &= \frac{1}{|\text{out}(\mathbf{c})|} \int_{\text{out}(\mathbf{c})} G_\sigma(\mathbf{q} - \mathbf{f}(\mathbf{x})) d\mathbf{x}, \end{aligned} \tag{4.19}$$

where G_σ is the 3D Gaussian with zero mean and diagonal covariance matrix $\sigma^2 \mathbf{I}$. They are indeed probability densities over \mathcal{Q} , as they are non-negative and verify

$$\begin{aligned} \int_{\mathcal{Q}} \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}] d\mathbf{q} &= 1 \\ \int_{\mathcal{Q}} \tilde{p}_{\text{out}}[\mathbf{c}, \mathbf{q}] d\mathbf{q} &= 1. \end{aligned}$$

Estimating color densities in this way may be viewed as computing "smoothed" normalized color histograms within regions.

If Z is a random variable with density $p_Z(z)$, its entropy is

$$h(Z) = - \int p_Z(z) \log p_Z(z) dz.$$

³also known as Parzen window

⁴Note that it would be improper to write $\Pr(\mathbf{f}(\mathbf{x}) = \mathbf{q} | \mathbf{x} \in \text{in}(\mathbf{c})) = \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}]$. An intuitive distinction between densities and probabilities of continuous random variables can be found e.g. in [Bishop, 2006, Section 1.2.1]

In [Kim et al., 2005], it is proposed to minimize the entropy of the color distributions inside and outside \mathbf{c} . Introducing X as the random variable whose \mathbf{x} is a realization, we can express the conditional entropy of $\mathbf{f}(X)$ given that X has a realization in $\text{in}(\mathbf{c})$,

$$h(\mathbf{f}(X)|X \in \text{in}(\mathbf{c})) = - \int_{\mathcal{Q}} \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}] \log \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}] d\mathbf{q}.$$

and similarly if $\mathbf{x} \in \text{out}(\mathbf{c})$, using \tilde{p}_{out} . It is shown in [Kim et al., 2005] that $h(\mathbf{f}(X)|X \in \text{in}(\mathbf{c}))$ can be rewritten as an integral over $\text{in}(\mathbf{c})$,

$$h(\mathbf{f}(X)|X \in \text{in}(\mathbf{c})) = - \frac{1}{|\text{in}(\mathbf{c})|} \int_{\text{in}(\mathbf{c})} \log \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{f}(\mathbf{x})] d\mathbf{x},$$

and similarly for $h(\mathbf{f}(X)|X \in \text{out}(\mathbf{c}))$. This leads to the data term

$$E_{\text{data}}^{\text{entropy}}[\mathbf{c}] = - \int_{\text{in}(\mathbf{c})} \log \tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{f}(\mathbf{x})] d\mathbf{x} - \int_{\text{out}(\mathbf{c})} \log \tilde{p}_{\text{out}}[\mathbf{c}, \mathbf{f}(\mathbf{x})] d\mathbf{x}. \quad (4.20)$$

which already appeared in the famous region competition paper [Zhu and Yuille, 1996, Section 3].

In [Michailovich et al., 2007], it is proposed to use the Bhattacharyya coefficient between the conditional densities inside and outside \mathbf{c} ,

$$E_{\text{data}}^{\text{Bhat}}[\mathbf{c}] = \int_{\mathcal{Q}} \sqrt{\tilde{p}_{\text{in}}[\mathbf{c}, \mathbf{q}] \tilde{p}_{\text{out}}[\mathbf{c}, \mathbf{q}]} d\mathbf{q}. \quad (4.21)$$

As the Bhattacharyya coefficient is a similarity measure between distributions, minimizing it tends to move the curve so as to increase the distance between color distributions inside and outside \mathbf{c} .

4.5 Minimal paths

4.5.1 Basics

[Cohen and Kimmel, 1997] proposed the minimal path method, in order to find a global minimum of the geodesic active contour functional (4.17). The global minimum can be found with the additional constraint that the curve should be open and its two endpoints fixed. It is expected that these provided endpoints are located on the contour of the target object. To some extent, the minimal path approach addresses the issue of sensitivity to local minima of edge-based active contours [Caselles et al., 1997, Kass et al., 1988, Malladi et al., 1995], which are driven by gradient descent of the Euler-Lagrange equation derived from the energy.

As in active contours, one seeks for a regular curve of minimal energy, or cost, which can be viewed as the length according to an heterogeneous isotropic metric defined from a potential $P : \mathcal{D} \rightarrow \mathbb{R}^{*+}$,

$$C[\mathbf{c}] = \int_0^1 P(\mathbf{c}(u)) \|\mathbf{c}'(u)\| du. \quad (4.22)$$

In the context of contour extraction, curves should be located along edges. The potential should thus be chosen so as to take lower values on edges,

$$P(\mathbf{x}) = g(\mathbf{x}) + w, \quad (4.23)$$

where g is the edge penalty function defined in Eq. (4.16) and $w \in \mathbb{R}^{*+}$ is the weight of the Euclidean distance component, acting as a regularizing constant. A toy image and its corresponding potential are shown in Figs. 4.1(a) and 4.1(b), respectively. One may note that $C[\mathbf{c}]$ is the geodesic data term (4.17), added with the Euclidean length weighted by w ,

$$C[\mathbf{c}] = E_{\text{data}}^{\text{geodesic}}[\mathbf{c}] + wL[\mathbf{c}].$$

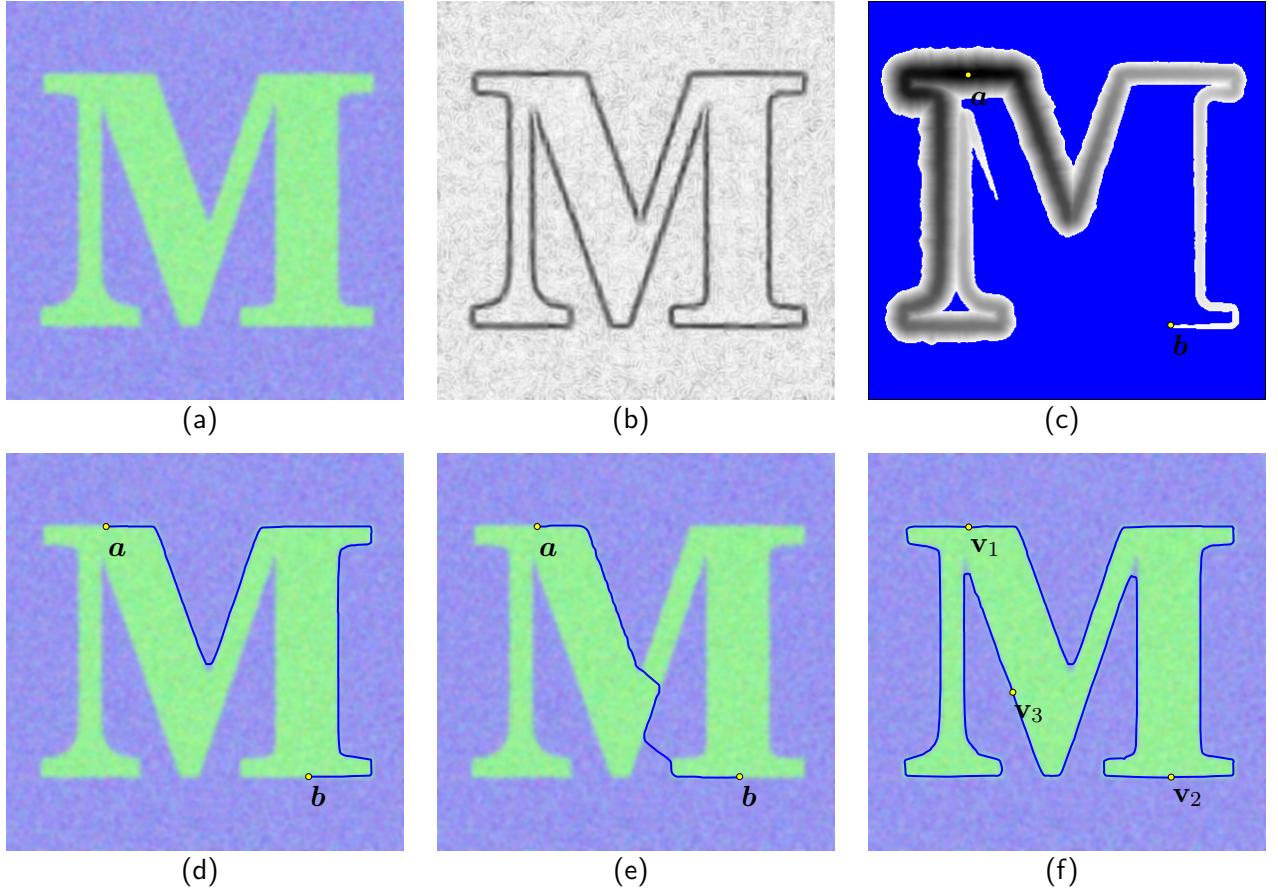


Figure 4.1: (a) Input image. (b) Potential P . (c) Action map U_a (d) With a sufficiently low regularization weight w , the geodesic between two given points follows the object contour. (e) Due to an excessive regularization weight w , the geodesic makes an undesirable shortcut. (f) Piecewise geodesic curve.

The interesting part of the object boundary is then extracted by finding a path of minimal cost among all paths connecting two given points a and b located on the boundary,

$$\gamma_{a,b} = \operatorname{argmin}_c \{C[c]\} \quad \text{s.t.} \quad \begin{cases} c(0) = a \\ c(1) = b. \end{cases} \quad (4.24)$$

Such a globally defined minimal path is called a geodesic. The solution of minimization problem (4.24) can be obtained by considering the geodesic distance map, also referred to as the *action map*, $U_a : \mathcal{D} \rightarrow \mathbb{R}^+$ which assigns, to each point $x \in \mathcal{D}$, the cost of the minimal path connecting x to a given point $a \in \mathcal{D}$:

$$U_a(x) = \inf_c \{C[c]\} \quad \text{s.t.} \quad \begin{cases} c(0) = a, \\ c(1) = x. \end{cases} \quad (4.25)$$

This map is the unique solution of the Eikonal equation

$$\begin{cases} \|\nabla U_a(x)\| = P(x), & \forall x \in \mathcal{D} \setminus \{a\}, \\ U_a(a) = 0, \end{cases} \quad (4.26)$$

see for instance [Crandall et al., 1992]. This allows to replace optimization problem (4.25) by a partial differential equation. Its discretized version, on a cartesian grid, can be efficiently solved by the Fast Marching algorithm [Tsitsiklis, 1995, Sethian, 1996, Sethian, 1999].

Once the action map has been numerically computed, the minimal path from any point x of \mathcal{D} to a can be extracted by a gradient descent on U_a ,

$$\begin{cases} \gamma_{x,a}'(u) = -\frac{\nabla U_a(\gamma_{x,a}(u))}{\|\nabla U_a(\gamma_{x,a}(u))\|}, \\ \gamma_{x,a}(0) = x, \end{cases} \quad (4.27)$$

This corresponds to a back-propagation from x to a . The potential being isotropic, and thus independent from the direction, $\gamma_{a,x}$ can be obtained by simply reversing $\gamma_{x,a}$.

In practice, since the Fast Marching is a monotonically-advancing front propagation method, finding the minimal path between two points does not require to propagate the action on the whole domain \mathcal{D} . Starting from one point, the Fast Marching can be stopped when the second point is reached, ensuring that the minimal path can be extracted with (4.27). Action map U_a , with propagation stopped when b is reached, is shown in Fig. 4.1(c). Blue pixels correspond to those that were not reached when propagation was stopped ($U_a(x) = \infty$).

An example of relevant geodesic given two endpoints a and b , is shown in Fig. 4.1(d), whereas a failure case is depicted in Fig. 4.1(e). In the latter, some portions of the minimal path do not follow the desired curve. This happens for instance when P is too noisy or not contrasted enough, when the length of the target curve is too important, or when the regularization constant w is too high.

Regardless of the choice of w , when endpoints are fixed, the minimal path method finds a global minimum of (4.24). In comparison, curve evolution by gradient descent of (4.17) only finds a local minimum. Note that, in very specific cases, Eq. (4.24) can have several global minimizers. Imagine, for example, a uniform disk in a uniform background, with a and b diametrically opposed on the disk contour. The intensities in the disk and background are supposed to be sufficiently different, so that potential P is low along the circle and high elsewhere. Starting from a , it is thus cheaper to reach b by travelling along the circle than to cut through it. Obviously, the cost is the same whether the path travels clockwise or counterclockwise, and Eq. (4.24) has thus two global minimizers. However, in most practical cases, the global minimizer will be unique.

4.5.2 Relation to Dijkstra's shortest path algorithm

Without any additional precision, the term *path* denotes both an ordered sequence of connected nodes in a graph or an open continuous regular curve between two endpoints. The Fast Marching method shares several aspects with Dijkstra's shortest path algorithm on a directed graph. The Dijkstra's counterpart of the action U is the distance, which is the sum of edge costs along the current path from the start node to the considered node. In both Fast Marching and Dijkstra, at each iteration, the node or grid point with the smallest distance/action is selected, and the distance/action of the adjacent unprocessed nodes or grid points is updated. Both algorithms use a priority queue (min-heap) - see for instance [Cormen et al., 2009, Chapter 24] - in order to efficiently retrieve the node or grid point with the smallest distance/action. The use of a priority queue allows a propagation in $O(N \log N)$ operations, where N is the number of nodes or grid points.

However, the similarities are limited to these algorithmic considerations. Paths obtained with the two algorithms are different. Considering Dijkstra in a first place, a directed graph with no attributes on nodes is agnostic to the notion of space (the same graph can be drawn in an infinity of ways). Spatial relationships between nodes are embodied by arcs and the costs associated to them. When building the graph, the connectivity between grid nodes - the criterion to set arcs - should be chosen. The two most intuitive ones are the 4-connectivity and 8-connectivity shown in Figs. 4.2(a) and 4.2(b), respectively. To keep the figure simple, each edge should be seen as a symmetric connection, *i.e.* a pair of arcs in opposite orientations. If one uniformly chooses the cost of arcs as the Euclidean distance between grid points corresponding to the nodes (1 for vertical and horizontal edges, $\sqrt{2}$ for diagonal edges), there are several shortest paths from a

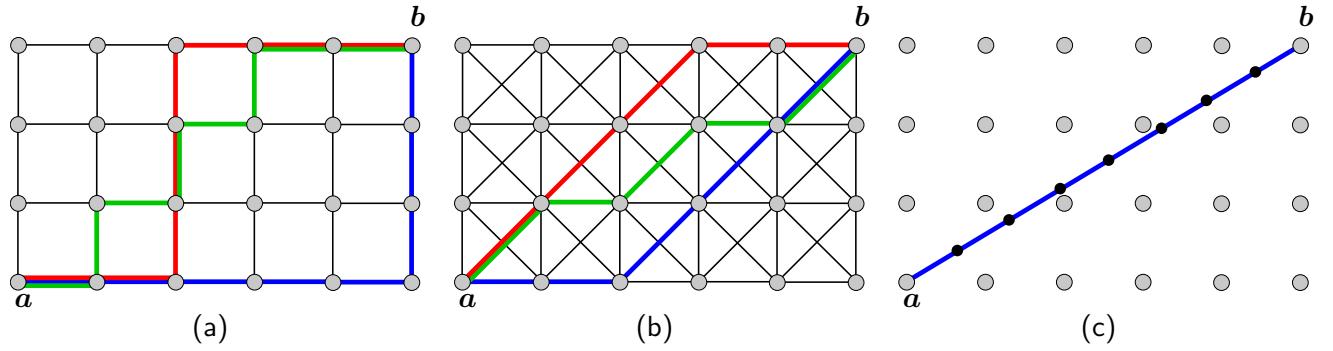


Figure 4.2: Dijksta versus Fast Marching on a grid, with uniform cost/potential. Using Dijksta on the graph, the shortest path is not unique and depends on the connectivity. In (a) and (b), the blue, green and red paths have the same cost (8 in (a) and $2 + 3\sqrt{2}$ in (b)). In (c), the Fast Marching generates a unique shortest path.

to b , and these paths depend on the chosen connectivity.

Unlike the shortest path on a graph, which is a discrete problem by essence, the minimal path method is a discretization of a continuous problem. In the Fast Marching, the action is interpolated between grid points. Its gradient, involved in back-propagation (4.27), is used to build path nodes that can be located between grid points. If one chooses a constant potential $P(\mathbf{x}) = 1 \forall \mathbf{x} \in \mathcal{D}$, which is the continuous counterpart of the choice of arc costs done previously, the back-propagation generates a unique straight line from a to b , as shown in Fig. 4.2(c). The space between black path nodes is the time step used in the discretization of Eq. (4.27). As explained in [Sethian, 1999, page 94], the Fast Marching does not suffer from metrification errors [Cohen and Kimmel, 1997] and is consistent with the initial continuous problem.

4.6 Building closed contours with minimal paths

In its basic formulation, the minimal path method extracts open curves between two fixed endpoints. It is however able to find closed curves, by modification of the initial framework.

4.6.1 Existing work

On one hand, a class of methods requires minimal interaction, i.e. a single user-provided seed on the object boundary. For instance, the construction of a closed curve from a single point was considered in [Cohen and Kimmel, 1997]. As previously, the action is propagated from the origin point using the Fast Marching. The first point where the front collides with itself, namely a *saddle point*⁵, is detected and taken as the origin of two opposite backpropagations, both sides apart from the colliding location, so that the two resulting paths can be assembled in order to form a closed contour.

In [Benmansour and Cohen, 2009], it is proposed to recursively detect intermediate keypoints during the propagation, with respect to a stopping criterion based on the Euclidean length of minimal paths. Each detected keypoint is taken as the origin of a new propagation, until the first point is reached again, thus making a closed contour.

As reported in [Miranda et al., 2012], many objects are just too complex to be segmented by providing a single seed on their contour. Considering for example the segmentation of anatomical structures in biomedical imaging, there can be numerous reasons for paths to travel along boundaries other than the desired ones:

- lack of contrast in some parts of the target object contour, creating "weak" edges with high potential

⁵At this stage, we do not provide a precise mathematical definition of saddle points yet. As they are used in the proposed CSPP method, a formal definition is given in Section 4.8.

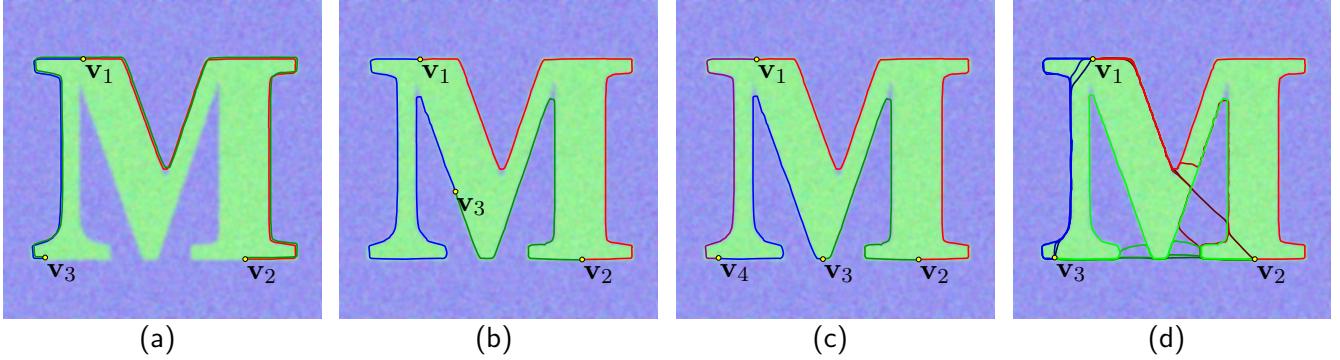


Figure 4.3: Towards a relevant initialization of the piecewise-geodesic curve: (a) Undesirable overlapping with unevenly spaced seeds, (b) improvement by even spacing of seeds, (c) improvement by addition of seed, (d) generation of several relevant paths per pair of successive seeds.

- complex shapes, with e.g bottleneck parts, favoring shortcuts inside the object
- cluttered backgrounds, containing other objects adjacent to the target, with stronger edges

In these situations, it is reasonable to think that, as soon as the algorithm can take advantage of multiple seeds on the object contour, better segmentation results can be expected. It is thus worth the effort to provide more manual interaction than a single point. In what follows, a first approach in this purpose is discussed.

4.6.2 Piecewise-geodesic contours

Closed contours may be constructed by linking together n ordered user-provided boundary seeds with minimum cost paths, as depicted in Fig. 4.1(f). In this purpose, we proposed a naive model, which can be referred to as piecewise-geodesic contour (PGC). It simply consists in linking each seed v_i to the next one, v_{i+1} , using the same potential map P . This model was developed during my postdoc and is discussed in [SSVM09]. It can be rigorously defined by introducing a curve concatenation operator. The curve resulting from the concatenation of two curves c_1 and c_2 , which is valid only if $c_1(1) = c_2(0)$, is defined as follows:

$$(c_1 \uplus c_2)(u) = \begin{cases} c_1(2u) & \text{if } 0 \leq u \leq \frac{1}{2} \\ c_2(2u - 1) & \text{if } \frac{1}{2} < u \leq 1. \end{cases} \quad (4.28)$$

Hence, the closed curve resulting from the assembly of successive geodesics is

$$\Gamma = \gamma_{v_1, v_2} \uplus \gamma_{v_2, v_3} \uplus \dots \uplus \gamma_{v_{n-1}, v_n} \uplus \gamma_{v_n, v_1}.$$

It is important to keep in mind that a concatenation of geodesics is not a geodesic itself, thus it is relevant to refer to Γ as *piecewise-geodesic*. Note that we should have $n \geq 3$.

As regards segmentation, the naive piecewise-geodesic contour turns out to have significant drawbacks. First, it does not take into account any region homogeneity criterion inside and outside the contour. It cannot integrate region-based data terms. Second, it is not necessarily a simple closed curve. As depicted in Fig. 4.3(a), this can occur when the initial seeds are unevenly distributed around the target boundary. In this case, geodesics gather on particular sides of the target boundary, as γ_{v_2, v_3} takes a way opposed to the expected one, and overlaps γ_{v_1, v_2} and γ_{v_3, v_1} . Comparison with other proposed methods in Section (4.10) will also report this behavior. The reason is that each geodesic is generated independently of the others, such that the obtained piecewise-geodesic curve does not depend on the visiting order of pairs of adjacent seeds.

This problem can be seen as a sampling one. Intuitively, one could think of imposing evenly spaced seeds, as depicted in Fig. 4.3(b), or adding seeds near the parts of the target boundary which are not covered by the piecewise geodesic curve, like in Fig. 4.3(c). In the considered context, such sampling criteria are difficult to express, since the target boundary is unknown and the sequence of seeds is a fixed input. Otherwise, one could think of imposing hard constraints on the overlapping between paths or penalizing paths enclosing a region with excessively small area, but it is natural to expect that the construction of the closed contour should not depend on the order seeds are processed. The independent construction of paths, which allows parallel implementation, prevents such constraints to be implemented.

4.6.3 Making minimal paths and region-based data terms work together?

Whether based on a single or several user-provided point(s), minimal path-based approaches were not designed to consider the regions inside and outside the contour in the optimization process. Indeed, they fall into the “edge-based only” category of segmentation methods, the edge-fitting criterion being indirectly contained in the definition of minimum cost paths. However, adding region-based data terms into the energy functional has proven to go beyond limitations of “edge-based only” energies, as presented in Section 4.4.2.

However, if one desires to integrate region terms into segmentation methods relying on minimal paths, one quickly gets faced with a “chicken and egg” dilemma: inner and outer regions are well defined only when the contour is closed. Thus, region homogeneity can be expressed and optimized, even locally, only when all minimal paths are available. Conversely, if one wishes to compute minimal paths exactly and in agreement to some region homogeneity criterion, the regions should already exist. Hence, minimal paths and region-based energies cannot be held together in a classical variational framework.

Our following contributions gradually build a solution than can integrate region-based terms. The contributions share two aspects:

- The required input is a sequence of n seeds, $(\mathbf{v}_i)_{1 \leq i \leq n}$, with $n \geq 2$. They are assumed to be located on the target object boundary⁶. The ordering of these seeds should be consistent with the orientation of curves discussed in Section 4.2. Hence, they are expected to be ordered clockwise, in the standard computer left-handed coordinate system.
- The final closed curve, which is expected to fit to the object boundary, is built by assembling paths linking successive seeds. The built contour is independent of the processing order of pairs $(\mathbf{v}_i, \mathbf{v}_{i+1})$. There is no reason to favor one pair, as the starting one, over the others. As regards computational cost, such construction allows parallel implementation.

4.7 The CDP method: combination of disjoint paths

When linking two successive seeds $(\mathbf{v}_i, \mathbf{v}_{i+1})$ with a path, the key idea is that we should not limit ourselves to the geodesic, but generate several relevant paths for each pair of successive seeds, as depicted in Fig. 4.3(d). The final segmentation should be built by selecting the combination of paths generating the best closed contour, with respect to some energy. The first attempt in that direction was the CDP method, for Combination of Disjoint Paths, presented in [VISAPP12]⁷.

4.7.1 Disjoint paths

Instead of the unique geodesic γ_i for each pair of successive seeds $(\mathbf{v}_i, \mathbf{v}_{i+1})$, a set \mathcal{A}_i of K short paths available for this pair is considered:

$$\mathcal{A}_i = \{\gamma_{i,j}\}_{1 \leq j \leq K}.$$

⁶The object is assumed to be simply connected, i.e. made up of a single connected component with no hole

⁷It was not referred to as CDP in the conference paper. We use this acronym for convenience in this manuscript

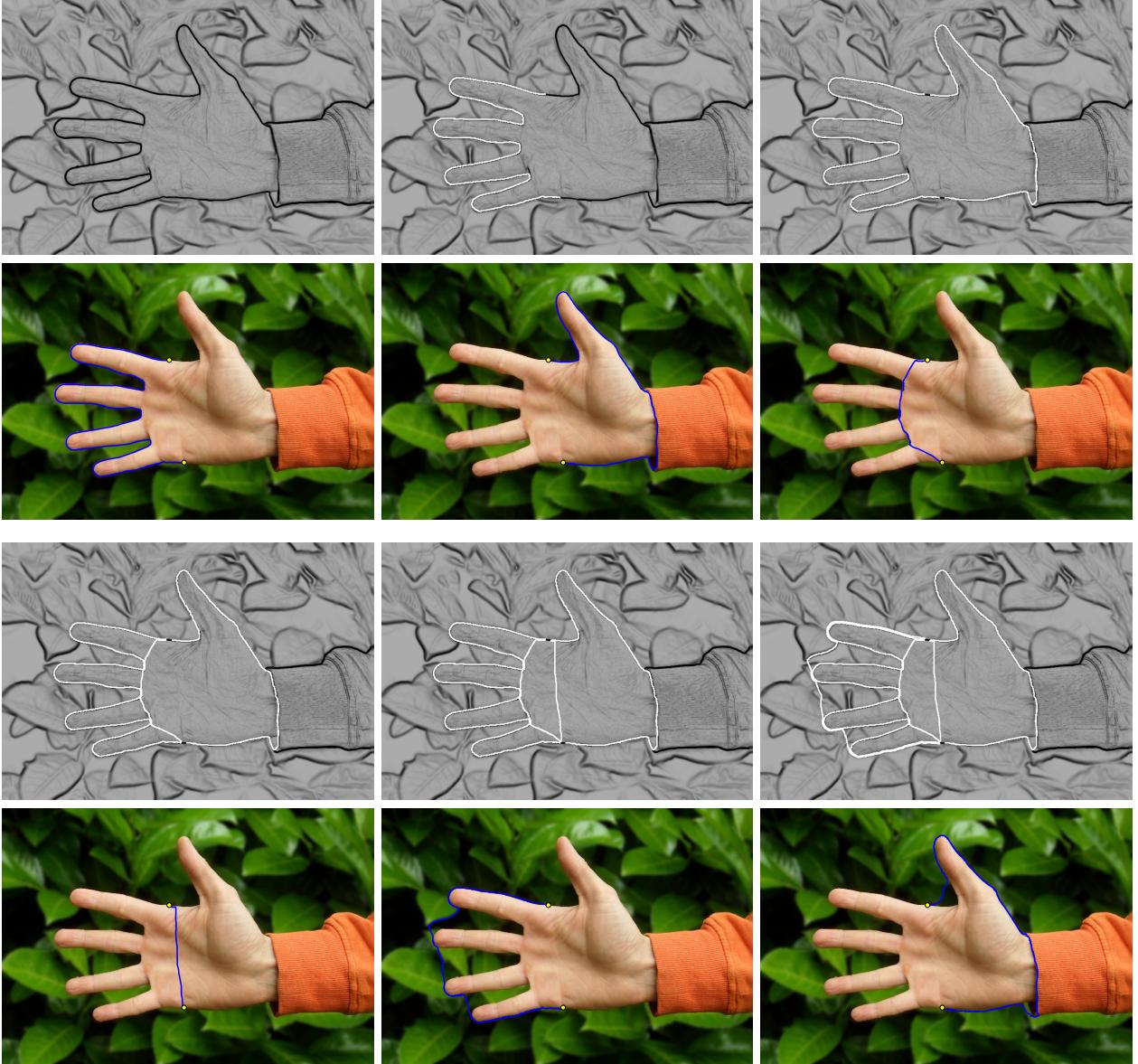


Figure 4.4: Generation of $K = 6$ disjoint paths for a given pair of endpoints. (Top row) Successive potential maps P_j , where the black-to-gray scale corresponds to range $[0, 1]$ and white corresponds to $+\infty$. (Bottom row) Corresponding paths $\gamma_{i,j}$, with $j = 1 \dots 6$.

Paths in \mathcal{A}_i are sorted by cost in ascending order, so that $\gamma_{i,1}$ actually corresponds to the minimal path between \mathbf{v}_i and \mathbf{v}_{i+1} whereas the remaining curves $\gamma_{i,j}$, $2 \leq j \leq K$, are only short paths of increasing cost. Conversely to the sets of paths in Fig. 4.3(d), which illustrates the general principle, paths are disjoint within each set \mathcal{A}_i . They are referred to as *admissible* paths.

The K admissible paths are built by successive deletion of already existing paths from the potential map, hence imposing paths to be disjoint. Curve $\gamma_{i,1}$ is the minimal path between \mathbf{v}_i and \mathbf{v}_{i+1} in the space endowed with the initial potential $P_1 = P$. Once the minimal path $\gamma_{i,1}$ has been computed, the second admissible path $\gamma_{i,2}$ is sought under the constraint that it should not pass through points belonging to $\gamma_{i,1}$. Hence, $\gamma_{i,2}$ is not a geodesic in the space induced by potential P , but in the space induced by a modified potential P_2 . The deletion of $\gamma_{i,1}$ in the modified potential map is achieved by setting the potential to $+\infty$ at all points of the geodesic. Extending this principle to the construction of the $j + 1^{\text{th}}$ admissible path $\gamma_{i,j+1}$, a recursive

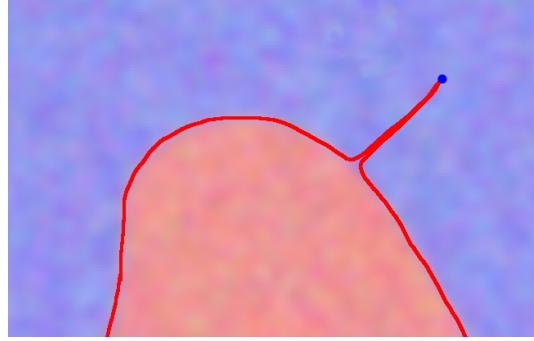


Figure 4.5: When an input seed is located at some distance of the boundary, paths linking this seed tend to overlap until they reach the object contour, and take different directions

definition of potential functions can be written as:

$$P_{j+1}(\mathbf{x}) = \begin{cases} +\infty & \text{if } \mathbf{x} \in \gamma_{i,j} \\ P_j(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (4.29)$$

with initial condition $P_1 = P$. This approach should not be mistaken for the so-called *K shortest paths* problem [Yen, 1971, Eppstein, 1998], which, in its basic formulation, does not impose paths to be disjoint. In the present case, the non-overlap constraint simplifies the generation of several paths. Intuitively, in a graph, the disjoint paths between a pair of seeds can be found by running several instances of the shortest path algorithm, after removal of seeds and incident edges belonging to already found paths. In the current continuous framework, the generation of paths obeys the same principle, as the Fast Marching algorithm needs to be run as many times as additional paths are required. An example of a set of successive disjoint paths $\gamma_{i,j}$ and corresponding potential maps P_j is depicted in Fig. 4.4.

4.7.2 Detecting self-tangency

Let us now assume that several admissible paths are concatenated one behind the other, yielding an assembled closed contour Γ . One of the desirable properties of Γ is that it should be simple, i.e. with no multiple point. Instead of imposing simplicity as a hard constraint, which might exclude relevant contours, it is encouraged by an additional energy. Indeed, it is reasonable to allow a certain degree of non-simplicity, e.g. when seeds are not located precisely on target boundaries, which might cause several admissible paths to have common sections before splitting up, as depicted in Fig. 4.5. Dealing with the geometrical and topological properties of the obtained curve, we wish to measure in what extent the curve is not simple.

If a curve \mathbf{c} is non-simple, it has a number of *multiple points* that should be studied. We only consider *double points*⁸, which may be of two kinds: self-tangencies and self-intersections. Only self-tangencies are taken into account in the CDP method.

Let $(u, v) \in [0, 1]^2$ s.t. $u \neq v$ be the pair of curve positions identifying a double point: $\mathbf{c}(u) = \mathbf{c}(v)$. If (u, v) corresponds to a point of self-tangency, velocity vectors $\mathbf{c}'(u)$ and $\mathbf{c}'(v)$ are colinear:

$$\frac{|\mathbf{c}'(u) \cdot \mathbf{c}'(v)|}{\|\mathbf{c}'(u)\| \|\mathbf{c}'(v)\|} = 1.$$

On the other hand, if (u, v) corresponds to a self-intersection - also known as an *ordinary double point* or *crunode* - velocity vectors point towards different directions, making the curve cross itself. This distinction allows to address separately two different defects on curves, which are not necessarily related. A curve with

⁸In our framework, curves with points of multiplicity > 2 are detected and excluded from the search

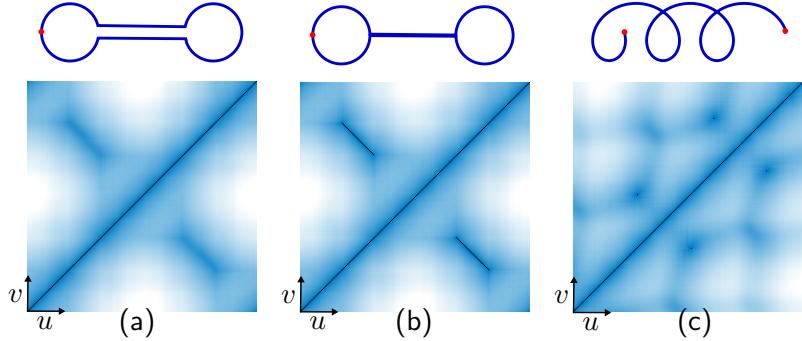


Figure 4.6: Distances ϕ_c plotted in (u, v) -space for several types of curve: (a) On a simple closed curve, ϕ_c vanishes only on the graph diagonal. (b) On a curve with a section of self-tangency, ϕ_c additionally exhibits two symmetrical zero lines. (c) On a curve with self-intersections, ϕ_c additionally exhibits isolated zeros.

points of self-tangencies will exhibit self-overlapping segments, as depicted in Fig. 4.6(b), whereas a curve with self-intersections shown in Fig. 4.6(c) will split the image domain into more than two connected regions.

As regards the first kind of double points, the amount of self-tangency is quantified by measuring the length of overlapping curve segments. Considering function ϕ_c measuring the Euclidean distance between two points on curve c ,

$$\phi_c(u, v) = \|\mathbf{c}(u) - \mathbf{c}(v)\|,$$

the zero level set of ϕ_c ,

$$\mathcal{Z}_c = \{(u, v) \mid \mathbf{c}(u) = \mathbf{c}(v)\},$$

is the set of pairs of positions giving equal points. Trivially, this set is never empty, since it contains at least all pairs (u, u) . The length of \mathcal{Z}_c in the (u, v) -space corresponds to the total length of the overlapping segments. Following [Chan and Vese, 2001, Section II.B], the length of the zero-level set is written

$$\begin{aligned} |\mathcal{Z}_c| &= \int_0^1 \int_0^1 \|\nabla H(\phi_c(u, v))\| \, du \, dv \\ &= \int_0^1 \int_0^1 \delta(\phi_c(u, v)) \|\nabla \phi_c(u, v)\| \, du \, dv, \end{aligned} \tag{4.30}$$

where ∇ is the gradient operator in the (u, v) -space, H is the Heaviside step function and δ is the Dirac delta distribution.

It can be proven that for any simple regular curve c , the amount of self-tangency $|\mathcal{Z}_c|$ is $\sqrt{2}$ (a detailed proof with a more general distance function and parameterization is given in appendix A.1.1). It can be intuitively understood by observing the plots of ϕ in the curve parameter space in Fig. 4.6, where dark lines correspond to self-overlapping segments. If the curve is non-simple, $|\mathcal{Z}_c| \geq \sqrt{2}$. Moreover, this term is advantageously intrinsic, i.e. independent of parameterization, and invariant to scaling. Note that intersection points may be viewed as overlapping segments of length zero, and thus have no contribution in $|\mathcal{Z}_c|$. Details about the discretization and computation of Eq. (4.30) are given in Appendix A.2.2.

4.7.3 Combining paths

The computation of a closed contour Γ consists in selecting one path out of each set \mathcal{A}_i , such that the contour resulting from the concatenation of selected paths minimizes energy E ,

$$(p_1^*, p_2^*, \dots, p_n^*) = \operatorname{argmin}_{(p_1, p_2, \dots, p_n) \in [1..K]^n} E[\gamma_{1,p_1} \uplus \gamma_{2,p_2} \uplus \dots \uplus \gamma_{n,p_n}].$$

Energy E is made up of an edge-fitting and region homogeneity terms, as well as a term penalizing self-overlapping of the curve,

$$E[\Gamma] = E_{\text{overlap}}[\Gamma] + \omega_{\text{edge}} E_{\text{edge}}[\Gamma] + \omega_{\text{region}} E_{\text{region}}[\Gamma]. \quad (4.31)$$

The three terms are normalized between 0 and 1. The edge-based energy integrates the edge penalty function g along the geodesics:

$$E_{\text{edge}}[\Gamma] = \frac{1}{L[\Gamma]} \int_0^1 g(\Gamma(u)) \|\Gamma'(u)\| du. \quad (4.32)$$

In order not to penalize lengthy contours, it is normalized by the Euclidean length

$$L[\Gamma] = \sum_{i=1}^n L[\gamma_i].$$

One may note that the edge indicator g is used instead of the potential P so that the Euclidean component of the curve length is not taken into account. This ensures that short curves, which could be undesirable shortcuts, are not preferred over longer ones.

In addition, the region-based energy allows to overcome limitations of the edge-based only models. Using the piecewise-constant data term in Eq. (4.18), we assume that curve Γ should partition the image into two homogeneous regions,

$$E_{\text{region}}[\Gamma] = \frac{1}{|\mathcal{D}|} E_{\text{data}}^{\text{CV}}[\Gamma]$$

The overlap term is a normalized version of $|\mathcal{Z}_\Gamma|$, defined in Eq. (4.30),

$$E_{\text{overlap}}[\Gamma] = \frac{|\mathcal{Z}_\Gamma| - \sqrt{2}}{\sqrt{2}}, \quad (4.33)$$

such that $E_{\text{overlap}}[\Gamma] = 0$ for a simple curve, and $E_{\text{overlap}}[\Gamma] = 1$ for a curve that fully overlaps itself once. For example, for the following parameterization,

$$\mathbf{c}(u) = [r \cos(4\pi u), r \sin(4\pi u)]^\top,$$

which makes two circular loops, we would have $E_{\text{overlap}}[\mathbf{c}] = 1$.

The best combination of paths is determined by an exhaustive search over all possible combinations. With n seeds and K paths per pair of successive seeds, such brute force search leads to an exponential complexity in $O(K^n)$. The full CDP method is summarized in Algorithm 1.

4.7.4 Limitations of the CDP method

The CDP method was a first attempt at combining admissible paths. As a preliminary work, it suffers from several shortcomings. Some of them are observable on segmentation results, in the experiments reported in Section (4.10). As regards the theoretical framework, improvements brought by the next work are justified in a first time by highlighting these shortcomings, particularly the generation of disjoint paths.

When computing K disjoint paths, preventing each path to pass through already visited points arbitrarily removes sections of contour that could be taken by other relevant paths. Moreover, in case of unsharp contours, low potential areas may get undesirably thick, so that two paths may agglomerate, i.e. take the same contour sections, the second path being most often an offset curve of the first one. For instance, this is the case for the 2nd and 6th paths shown in Fig. 4.4, respectively. This behavior, referred to as the *path agglomeration* issue, is further discussed in Section 4.8.3 once the proposed construction of path is presented.

Algorithm 1: Summary algorithm of the CDP

Input: :
image f , sequence of n seeds $\{\mathbf{v}_i\}_{1 \leq i \leq n}$

begin

- # Generate sets of admissible paths
- Compute potential P
- foreach** seed pair $(\mathbf{v}_i, \mathbf{v}_{i+1})$ **do**
- Set potential $P_1 = P$
- for** $j := 1 \dots K$ **do**
- Propagate action map $U_{\mathbf{v}_i}$ based on potential P_j , using Fast Marching
- Compute path $\gamma_{i,j}$ using backpropagation, and add it to \mathcal{A}_i
- Compute potential P_{j+1} using Eq. (4.29)
- end**
- end**
- # Exhaustive search optimization
- foreach** combination $(p_1, \dots, p_n) \in [1..K]^n$ **do**
- Compute $E[\gamma_{1,p_1} \cup \dots \cup \gamma_{n,p_n}]$ using Eq. (4.31)
- Store combination of minimal energy
- end**
- end**

Moreover, imposing a constant number K of admissible paths per pair of seed may be unsuitable in many situations, and raises the issue of choosing K appropriately. When the object and background are complex between seeds \mathbf{v}_i and \mathbf{v}_{i+1} , many relevant paths may be found and excessively small values of K would restrict their generation. Conversely, when the object and background are simple and well separated from each other, there may be a single obvious path and one would like to avoid computing supplemental paths which would turn out useless for the generation of the final contour.

As regards energy (4.31), the use of a piecewise-constant region-based data term is questionable, as it assumes that color is uniform within the object and background. It can cope with images where the average color in the object is sufficiently far from the one in the background. However, in real-world images, object and background may have close average colors while having very different color distributions. More general data terms, described in Section 4.4.2, which are not based on a parametric modeling of color distributions, can be considered. Moreover, the overlap term alone, in Eq. (4.33), is not sufficient to measure in what extent a curve is not simple. Consider an "8"-shaped curve. It does not overlap itself, except at the central crossing. However, it is far from being simple curve, since half of the curve needs to be "untwisted" to obtain a simple loop. Combination of geodesics can lead to twisted configurations, without being self-overlapping. Hence, in addition to self-overlap, there is need to account for this amount of twisting.

Finally, we point out the shortcomings related to the computational cost. Since the potential map is modified after the construction of one path, the action map should be propagated again so that the next path can be extracted. This implies to run K instances of the Fast Marching algorithm. Even if the $O(KN \log N)$ complexity is reasonable, extracting all paths from the same action map generated by a single propagation in $O(N \log N)$ would be more convenient. As regards the computation of the combination minimizing (4.31), exhaustive search is easy to implement but remains a major shortcoming. Despite the separability of the energy, exhaustive search, which has an exponential complexity $O(K^n)$, limits the practical usability of the approach. For obvious reasons, search methods with lower algorithmic complexity, even heuristical approaches, are preferable.

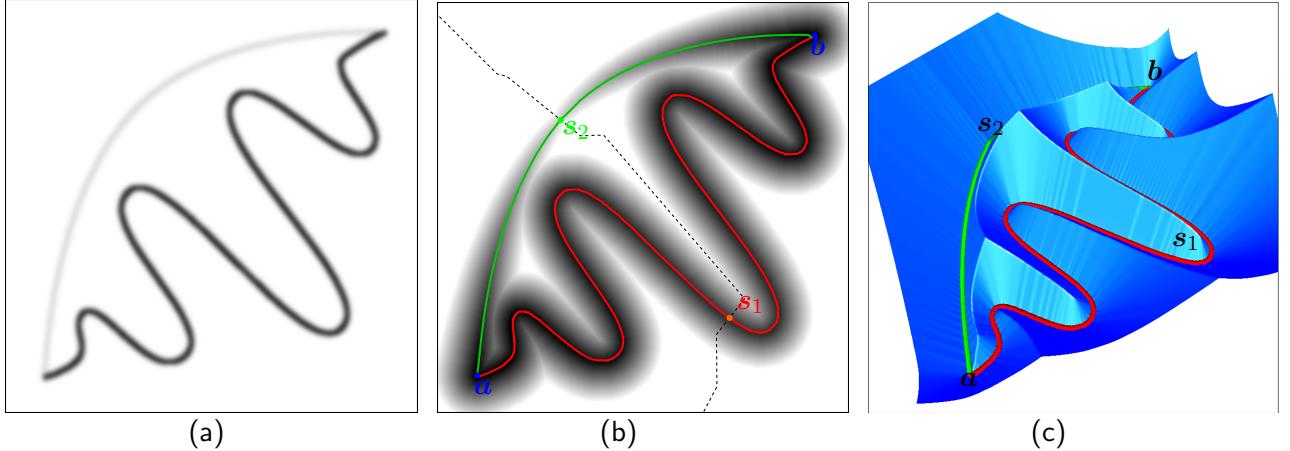


Figure 4.7: Saddle points as a basis for determining relevant admissible paths between two given endpoints a and b : (a) Potential highlights two possible distinctive paths, (b) Action map with two admissible paths with their respective saddle points located halfway and (c) corresponding 3D plot.

4.8 The CSPP method: combination of saddle point-based paths

This contribution was published in [BMVC13, IJCV15]. We replace the construction of disjoint paths by an approach based on the extraction of *saddle points*. We refer this method to as CSPP, for Combination of Saddle Point-based Paths. It is summarized in Algorithm 2.

Firstly, we generalize the definition of the admissible set \mathcal{A}_i , as a non-empty set of K_i admissible paths:

$$\mathcal{A}_i = \{\gamma_{i,j}\}_{1 \leq j \leq K_i}.$$

Unlike in the CDP approach, the number of admissible paths is allowed to vary with respect to the considered pair $(\mathbf{v}_i, \mathbf{v}_{i+1})$. As in the CDP, we assume that admissible sets are organized such that $\gamma_{i,1}$ is the geodesic path from \mathbf{v}_i to \mathbf{v}_{i+1} and remaining curves $\gamma_{i,j}$, $2 \leq j \leq K_i$, are other interesting paths.

4.8.1 Valleys and saddle points

Let us consider two distinct points a and b and the edge-based potential P . The geodesic linking a and b may be actually extracted in three different ways:

- Propagate the minimal action from a , stop when b is reached, and perform gradient descent from b
- Propagate the minimal action from b , stop when a is reached, and perform gradient descent from a
- Propagate **simultaneously** from a and b , stop at the first location where the two fronts collide, perform two gradient descents both sides apart from the meeting location, and assemble the two obtained paths adequately

The third principle is the basis for the generation of multiple paths. When the action map is propagated from two source points a and b - which generates what we refer to as the *combined action map* $U_{a,b}$ -, the two propagation fronts meet at the **saddle points** of $U_{a,b}$ (see Fig 4.7). If one intuitively thinks of the action as the height in a mountainous area, the saddle points are the highest points on the different roads travelling from a to b . These roads may either be disjoint or share common sections, but generally each road lies in the bottom of a particular valley. A formal definition of saddle points is given by first introducing the medial set $\mathcal{M}_{a,b}$, made up of points geodesically equidistant from a and b :

$$\mathcal{M}_{a,b} = \{x \mid U_a(x) = U_b(x)\}.$$

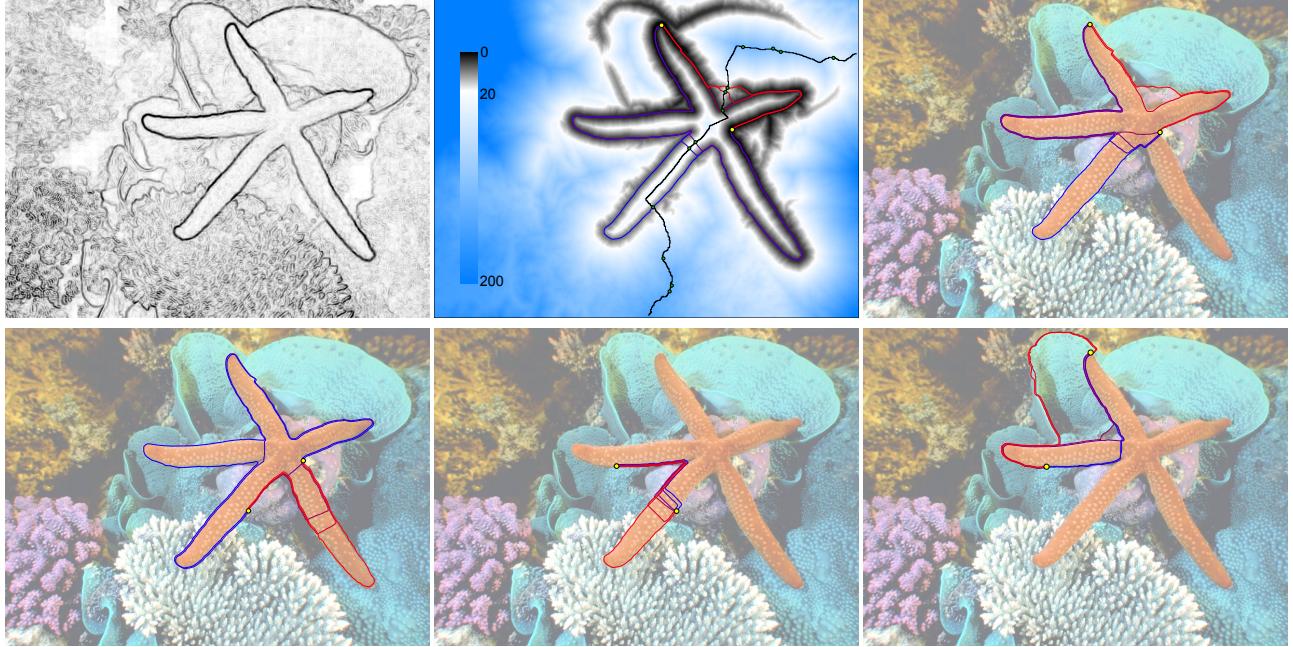


Figure 4.8: Admissible paths for every pair of successive input seeds. (Top row, left) Potential P and (Top row, middle) Medial curve (black) and saddle points (green) with corresponding paths drawn over combined action map (Top row, right, and bottom row) Admissible set for each pair of successive seeds with $n = 4$. Paths are sorted (blue to red) according to their exteriority.

The medial set $\mathcal{M}_{a,b}$ forms a crest on the combined action map and is thus a set of critical points of $U_{a,b}$, put another way $\nabla U_{a,b}$ is undefined over $\mathcal{M}_{a,b}$. However, set $\mathcal{M}_{a,b}$ may be parameterized by a simple regular curve - which may be closed or open, depending on the locations of the endpoints and the potential function - and the combined action map may be differentiated along it. Let $\mathbf{m}_{a,b} : [0, 1] \rightarrow \mathcal{D}$ be the *medial curve*, assumed to be of class C^2 , sweeping along the equidistant points. The saddle points on the combined action map between a and b are the local minima of $U_{a,b}$ along $\mathbf{m}_{a,b}$.

4.8.2 Admissible paths as piecewise-geodesic curves

The generation of admissible paths is the first stage of the CSPP method. For each pair of successive seeds $(\mathbf{v}_i, \mathbf{v}_{i+1})$, a combined action map $U_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ is propagated, generating two monotonically advancing fronts from \mathbf{v}_i and \mathbf{v}_{i+1} . The locations where the two fronts collide are detected and stored until the combined action map has been propagated over the entire image domain. Only locations that are local minima of $U_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ in the direction orthogonal to the propagation direction, i.e. along the medial curve, are kept as the K_i selected saddle points. Two gradient descents are performed both sides apart from each saddle point s_j , one path going down to \mathbf{v}_i and the other one to \mathbf{v}_{i+1} . In other words, we take as starting points for the gradient descents

$$\mathbf{m}(u_j) + \epsilon \frac{\mathbf{m}'^\perp(u_j)}{\|\mathbf{m}'(u_j)\|} \text{ and } \mathbf{m}(u_j) - \epsilon \frac{\mathbf{m}'^\perp(u_j)}{\|\mathbf{m}'(u_j)\|},$$

where \mathbf{m} stands for $\mathbf{m}_{\mathbf{v}_i, \mathbf{v}_{i+1}}$, u_j is the curve parameter such that $\mathbf{m}(u_j) = s_j$ and ϵ is an infinitesimal offset. These two points are necessarily on different slopes of the mountain, their respective gradient vectors $\nabla U_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ being almost opposite. The two gradient descents generate paths $\gamma_{s_j, \mathbf{v}_i}$ and $\gamma_{s_j, \mathbf{v}_{i+1}}$ respectively. The j^{th} admissible path in \mathcal{A}_i is obtained by reversing the first path and concatenating it with the other path,

$$\gamma_{i,j} = \gamma_{\mathbf{v}_i, s_j} \uplus \gamma_{s_j, \mathbf{v}_{i+1}}.$$

Algorithm 2: Summary algorithm of the CSPP method

Input: :

image f , sequence of n seeds $\{\mathbf{v}_i\}_{1 \leq i \leq n}$

begin

- # Generate sets of admissible paths
- Compute potential P
- foreach** seed pair $(\mathbf{v}_i, \mathbf{v}_{i+1})$ **do**

 - Propagate combined action map $U_{\mathbf{v}_i, \mathbf{v}_{i+1}}$ using Fast Marching
 - Extract saddle points of $U_{\mathbf{v}_i, \mathbf{v}_{i+1}}$
 - Build admissible set \mathcal{A}_i by performing gradient descents from saddle points
 - Sort paths in \mathcal{A}_i with respect to their exteriorty measure in Eq. (4.37)

- end**

- # Local search for the best combination
- Set current combination of admissible paths to $(1, \dots, 1)$
- while** at least one index in the combination can be increased **do**

 - Compute energy of current combination using Eq. (4.35)
 - Compute energies of at most n candidate combinations using Eq. (4.35)
 - Select the candidate combination with lowest energy as the current combination for the next iteration

- end**
- end**

Among all saddle points, the one with the lowest action, which is also the first colliding location of fronts,

$$\mathbf{s}_{\min} = \underset{\mathbf{s} \in \mathcal{S}_{\mathbf{v}_i, \mathbf{v}_{i+1}}}{\operatorname{argmin}} \{U_{\mathbf{v}_i, \mathbf{v}_{i+1}}(\mathbf{s})\},$$

is located halfway, in the geodesic sense, on the minimal path between \mathbf{v}_i and \mathbf{v}_{i+1} :

$$\gamma_{\mathbf{v}_i, \mathbf{s}_{\min}} \uplus \gamma_{\mathbf{s}_{\min}, \mathbf{v}_{i+1}} = \gamma_{\mathbf{v}_i, \mathbf{v}_{i+1}}.$$

The same relation does not hold for all higher saddle points s_j . Paths from \mathbf{v}_i to \mathbf{v}_{i+1} passing through these saddle points may be only referred to as *piecewise-geodesic*. Details about the discrete implementation of the Fast Marching and the extraction of saddle points are given in Section A.2.1.

4.8.3 Advantages over disjoint paths

We believe that considering saddle point-based paths, i.e. valley roads, instead of disjoint paths generates a more relevant and less redundant set of candidate paths. Indeed, different valleys may correspond to different sections of the boundary of an object, whereas different *short paths* may run through the same valley and thus be redundant. If the image contours were infinitesimally thick, only a single path could run through each valley, thus the notion of valley roads and disjoint shortest paths would be nearly equivalent. However, in real images where contours may be arbitrarily smooth, areas with low potential lead to large valleys on the action map. In continuous space, there is an infinite number of “almost minimal paths” passing along the bottom of the lowest valley.

Consider the example in Fig. 4.7(a) again, where the combined action map gives rise to two saddle points. Let s_1 be the saddle point on the low and curvy valley, and s_2 the saddle point on the higher and straighter valley. Let $\gamma_{a,b}^1$ and $\gamma_{a,b}^2$ be the paths passing through these respective saddle points, drawn in Fig. 4.7(b). Regardless of the Euclidean length, $\gamma_{a,b}^1$ is the shortest path in the geodesic sense: $\gamma_{a,b}^1 = \gamma_{a,b}$. Should the lower valley be large enough, i.e. the potential would be low over a thick band, the second shortest disjoint

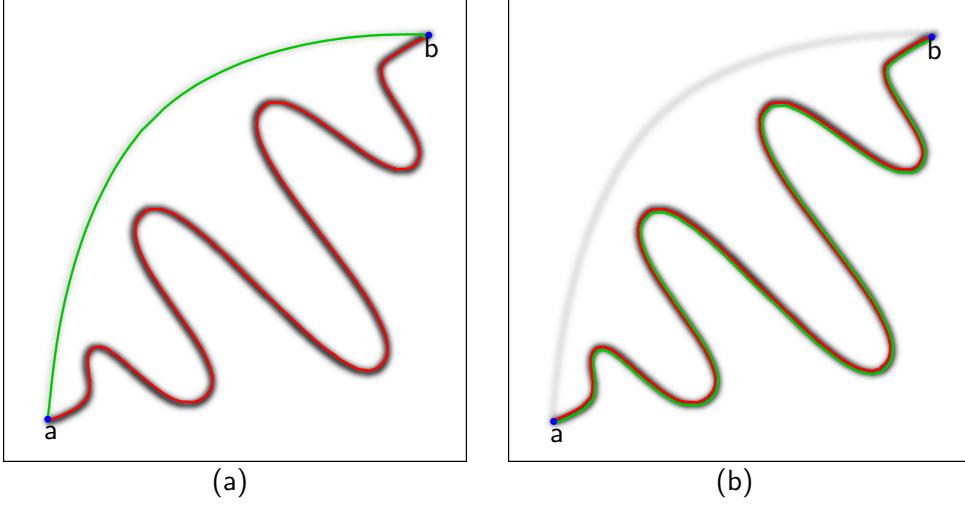


Figure 4.9: Saddle point-based paths versus disjoint paths on smooth contours. (a) The second path (green) extracted with the higher saddle point follows a contour distinct from the one followed by $\gamma_{a,b}$ (b) The second shortest path (green) constrained to be disjoint from $\gamma_{a,b}$ turns out to be an offset curve of $\gamma_{a,b}$.

path would be very likely an offset curve of $\gamma_{a,b}$, as depicted in Fig. 4.9. This illustrates the *path agglomeration* issue mentioned in Section 4.7.4. In the disjoint paths approach, this issue is dependent on the choice of the thickness of paths. A curve being by essence a mathematical object with infinitesimal thickness, setting P to $+\infty$ according to Eq. (4.29) has rigorously no effect since it does not prevent subsequent paths to take the same way, up to an infinitesimal offset. Actually, one has to add an artificial thickness to the curve, i.e. choose a bandwidth B over which subsequent paths will not be allowed to pass. Accounting for this consideration, a practical redefinition of Eq. (4.29) would be

$$P_{j+1}(\mathbf{x}) = \begin{cases} +\infty & \text{if } \exists u \in [0, 1] \text{ s.t. } \|\mathbf{x} - \gamma_{i,j}(u)\| \leq \frac{B}{2} \\ P_j(\mathbf{x}) & \text{otherwise.} \end{cases}$$

From an implementation point of view, since the potential is discretized over the image grid, it actually needs to be set to $+\infty$ at all integer-coordinates pixels within a ball of radius $B/2$ centered at each path point. B may be referred to as the *erasure thickness* and adds a supplemental parameter, which is needed both for defining the correct mathematical notion of *path disjointness* and for effective implementation. Unless the erasure thickness B is chosen as large as the thickness of low-potential valleys, subsequent paths tend to be offset curves of the geodesic curve. The thickness of valleys is an intractable property, depending on the sharpness of contours and varying both between valleys and along each valley, which prevents from choosing an appropriate value of B without prior knowledge. Conversely, in the proposed approach, we endeavour to develop a more mathematically sound framework, which is not influenced by an artificial erasure thickness added to curves. The proposed saddle point-based approach is not affected by the geometry of valleys, as it depends only on the number of local minima of the action map.

Allowing a variable number K_i of admissible paths per pair $(\mathbf{v}_i, \mathbf{v}_{i+1})$ is more relevant than the fixed K disjoint paths of the CDP. The action map having no flat plateau, the number of local minima along the medial curve is finite. Hence, the number of admissible paths is also finite, and varies with respect to the underlying image structures. In images containing smooth regions with sharp contours, the action maps exhibit few local minima along the medial curves and the resulting sets of admissible paths are advantageously small. In presence of cluttered or noisy data, there can be a large amount of local minima along the medial curve. In the CDP, there may be an infinite set of short disjoint paths, which has to be truncated by fixing a threshold K , regardless of the considered pair of seeds. On one hand, in case of images with smooth regions,

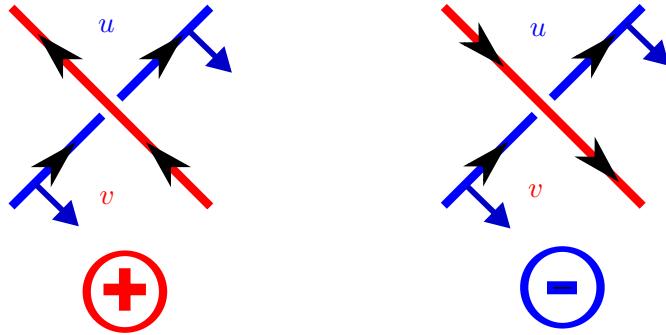


Figure 4.10: Positive and negative crossings defined by ordered pair of positions (u, v) . $c(u)$ and $c(v)$ are the intersected and intersecting sections in blue and red respectively.

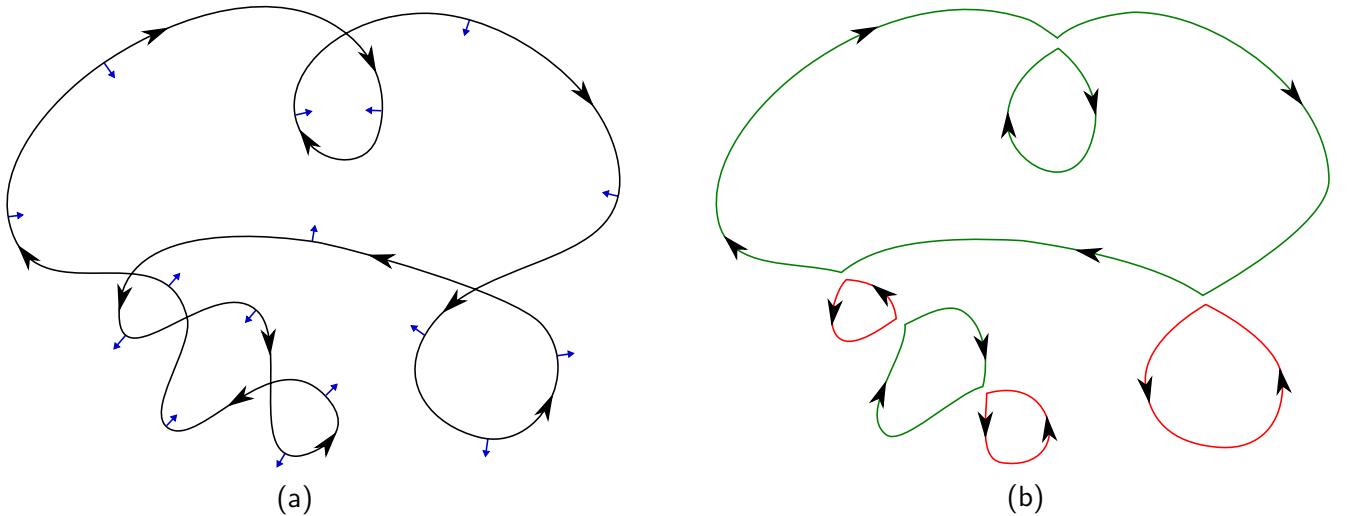


Figure 4.11: Self-intersecting curve and inverted segments: (a) A nonsimple loop with ordinary double points which could be transformed into a set of (b) disjoint simple loops using a sequence of *uncrossing moves*. Resulting badly-oriented (counterclockwise) simple loops are drawn in red.

redundant paths may be generated if K is too large, increasing the computational time of the optimization step. On the other hand, in case of cluttered images, relevant contours can be missed if K is too small.

Eventually, in the current CSPP approach, one propagation in $O(N \log N)$ is sufficient per pair of successive seeds. Each admissible path is computed on the combined action map regardless of other paths, which allows to compute admissible paths in parallel. Notice that, despite this possibility, computation times reported in Section 4.10 are obtained without any parallel implementation.

4.8.4 Twisting in non-simple curves

Let us recall that a non-simple curve can have multiple points of two kinds: tangent double points, which allowed to penalize self-tangencies in the CDP method, and ordinary double points, that indicate the presence of self-intersections. The problem of the overlap energy of the CDP is that it is not penalized by self-intersections. The proposed simplicity term in the CSPP method take them into account.

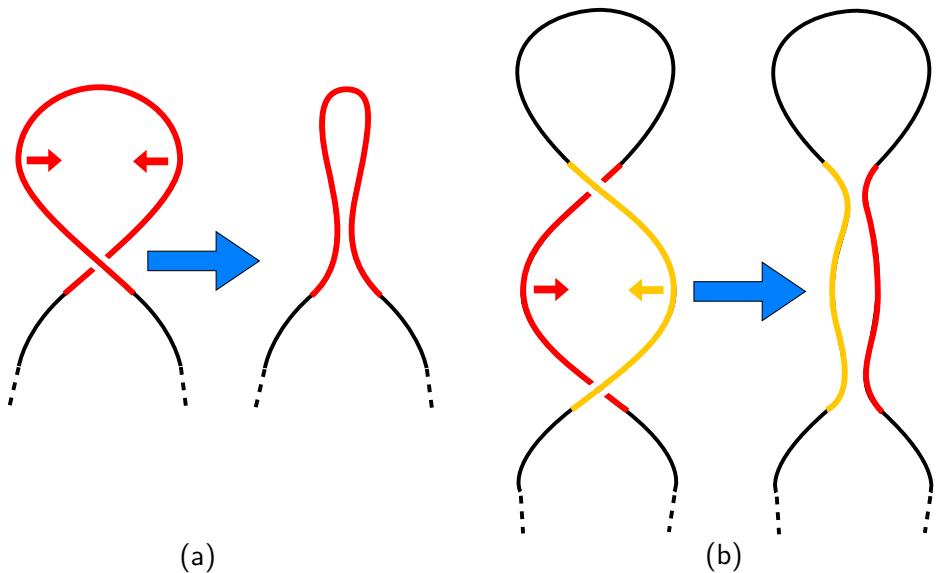


Figure 4.12: Untwisting simple and double loops by (a) Reidemeister move of type I and (b) Reidemeister move of type II. Only colored parts of the curve are moved.

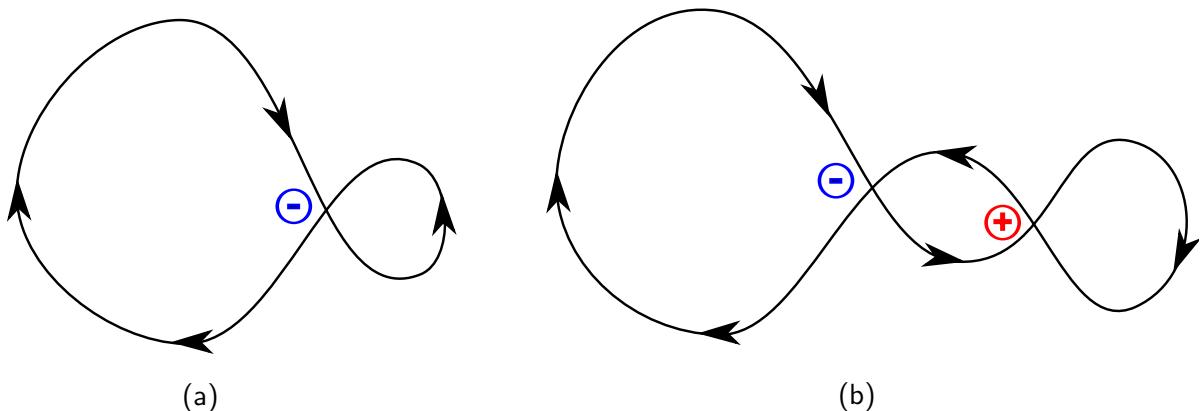


Figure 4.13: Simple and double loops making inverted curve segments. (a) Simple loop made by a single negative crossing and (b) double loop made by a couple of positive and negative crossings.

Self-crossings are pairs of curve positions (u, v) where tangent vectors are not colinear:

$$\frac{|\mathbf{c}'(u) \cdot \mathbf{c}'(v)|}{\|\mathbf{c}'(u)\| \|\mathbf{c}'(v)\|} \neq 1.$$

Pairs of positions corresponding to crossing points are ordered, so that u is the position where the curve is *intersected* and v is the position where the curve is *intersecting*. Whether a position on the curve is intersected or intersecting only depends on the order in which these positions are met while one travels along the curve, so it follows that $u < v$. Whitney [Whitney, 1937] studied crossing points of closed regular curves and divided them into two categories with respect to their orientation. Let us recall the orientation of curves, set up in the standard computer left-handed cartesian coordinate system. In this setting, we choose the natural orientation of curves as the clockwise one, such that when one travels along the curve, looking forward, the interior of the curve is on the right. In such case, \mathbf{c}'^\perp is the inward normal. The crossing will be *positive* if the intersecting part of the curve $\mathbf{c}(v)$ goes from right to left, or *negative* if it goes from left to right (see Fig. 4.10). In a positive (resp. negative) crossing, the intersecting section arrives from the interior (resp. exterior) of the intersected section. It follows that:

$$\begin{aligned} \mathbf{c}'(u)^\perp \cdot \mathbf{c}'(v) &< 0 \quad \text{for a positive crossing} \\ \mathbf{c}'(u)^\perp \cdot \mathbf{c}'(v) &> 0 \quad \text{for a negative crossing.} \end{aligned}$$

Consider the self-intersecting curve shown in Fig. 4.11(a). It splits the image domain into disjoint subdomains, some of which are demarcated by *inverted segments*, i.e. portions of curves along which the normal vector points outward. If one decomposes the curve using *uncrossing moves* - replacing each couple of crossing sections by two new non-crossing sections - one obtains a collection of disjoint simple loops, as in Fig. 4.11(b), also known as *Seifert circles* [Adams, 2004, p. 94]. While some simple loops are well oriented (clockwise), others are inverted (counter-clockwise). We propose to quantify the twisting of \mathbf{c} as the proportion of area demarcated by inverted segments to the total area of \mathbf{c} , or equivalently, the area inside the counter-clockwise oriented loops one would obtain when decomposing the curve. To some extent, this proportion is related to the energy that one should exert to untwist the curve.

A self-intersecting curve can be thought of as a planar projection of a knot, i.e. an embedding of the unit circle into Euclidean \mathbb{R}^3 space. Following the theory of knots [Adams, 2004, p. 13] [Arnold, 1994], we consider twisted configurations of curves in which corresponding knots could be unknotted by Reidemeister moves of type I and II (see Fig. 4.12). The simple loop, described by a single inverted segment and involving a single intersection, is related to the Reidemeister move of type I, whereas the double loop, consisting of two segments and involving two intersections, is related to the Reidemeister move of type II. The two types of crossing points, that were previously presented, are used to detect such loops. A simple inverted loop will be detected thanks to a negative crossing point (u, v) such that there is no other crossing from u to v , as in Fig. 4.13(a). A double inverted loop will be detected thanks to a negative crossing (u, v) such that the path from u to v contains a positive crossing, as in Fig. 4.13(b). Actually, we do not explicitly split the curve into simple loops, as in Fig. 4.11(b). We rather detect self-crossings, determine their orientations, and extract inverted loops on the initial curve.

Let $SL(\mathbf{c}) \subset [0, 1]^2$ be the set of ordered pairs of curve positions (u, v) s.t. $u < v$, describing intersections involved in single inverted loops, and $DL(\mathbf{c}) \subset [0, 1]^2 \times [0, 1]^2$ be the set of double ordered pairs $((u_1, v_1), (u_2, v_2))$ s.t. $u_1 < u_2$ and $v_1 > v_2$ describing the couples of intersections involved in double inverted loops. Sets SL and DL are extracted by the loop detection algorithm detailed in Appendix A.2.2. When a portion of curve \mathbf{c} from s to t , denoted $\mathbf{c}|_{s \rightarrow t}$, is closed and simple, its signed area can be calculated using (4.5). Eventually, the total area of inverted loops of \mathbf{c} , denoted by $\mathcal{I}[\mathbf{c}]$ is expressed by considering all simple and double inverted loops in SL and DL :

$$\mathcal{I}[\mathbf{c}] = - \sum_{(s,t) \in SL(\mathbf{c})} \int_s^t \frac{\mathbf{c}'^\perp \cdot \mathbf{c}'}{2} du - \sum_{\substack{((s_1,t_1),(s_2,t_2)) \\ \in DL(\mathbf{c})}} \int_{s_1}^{s_2} \frac{\mathbf{c}'^\perp \cdot \mathbf{c}'}{2} du + \int_{t_2}^{t_1} \frac{\mathbf{c}'^\perp \cdot \mathbf{c}'}{2} du. \quad (4.34)$$

Note that $\mathcal{I}[c]$ is positive, as the signed area of every inverted loop is negative.

4.9 Combination of admissible paths using the simplicity energy

4.9.1 Energy terms

The proposed energy functional extends the energy of the CDP in Eq. (4.31). It is designed to penalize contours exhibiting strongly overlapping sections or self-intersections, poorly fitting to image edges or enclosing regions with indistinct color distributions:

$$E[\Gamma] = E_{\text{simplicity}}[\Gamma] + \omega_{\text{edge}} E_{\text{edge}}[\Gamma] + \omega_{\text{region}} E_{\text{region}}[\Gamma]. \quad (4.35)$$

The simplicity term involves the self-tangency and twisting measures defined in Eqs. (4.30) and (4.34),

$$E_{\text{simplicity}}[\Gamma] = \frac{|\mathcal{Z}_\Gamma| - \sqrt{2}}{\sqrt{2}} + \frac{1}{A[\Gamma]} \mathcal{I}[\Gamma]. \quad (4.36)$$

As for the overlap term of the CDP method, $E_{\text{simplicity}}[\Gamma]$ vanishes when Γ is a simple curve.

As regards image-based terms, the same edge energy is kept as in the CDP, so E_{edge} is defined in Eq. (4.32). However, for the region term, instead of using the piecewise-constant model as in the CDP, which limits the segmentation to relatively homogeneous objects and backgrounds, we use the Bhattacharyya coefficient between the color probability distributions inside and outside Γ , following Eq (4.21),

$$E_{\text{region}}[\Gamma] = E_{\text{data}}^{\text{Bhat}}[\Gamma]$$

4.9.2 Local search method

As in the CDP method, the computation of an admissible closed contour can be formulated as determining the sequence of labels (p_1, p_2, \dots, p_n) minimizing energy (4.35), where label p_i corresponds to the chosen path in set \mathcal{A}_i :

$$(p_1^*, p_2^*, \dots, p_n^*) = \underset{\substack{(p_1, p_2, \dots, p_n) \in \\ [1..K_1] \times [1..K_2] \times \dots \times [1..K_n]}}{\operatorname{argmin}} E [\gamma_{1,p_1} \sqcup \gamma_{2,p_2} \sqcup \dots \sqcup \gamma_{n,p_n}],$$

Let K_{\max} be the maximum allowed number of admissible paths over all pairs of successive seeds,

$$K_{\max} = \max_{1 \leq i \leq n} K_i.$$

A brute-force search as in the CDP would yield an exponential time-complexity, upper-bounded by $O(K_{\max}^n)$. We propose a greedy search in $O(n^2 K_{\max})$ relying on a specific ordering of paths. In each admissible set \mathcal{A}_i , paths are sorted according to increasing *exteriority* \mathcal{X} , i.e. the signed area formed by a given path c and the line segment from $c(1)$ returning to $c(0)$,

$$\mathcal{X}[c] = \frac{1}{2} \int_0^1 c^\perp \cdot c' du + \frac{1}{2} c(1)^\perp \cdot c(0), \quad (4.37)$$

which is proven in Appendix A.1.2. If the straight line from $c(0)$ to $c(1)$ is taken as a reference horizontal axis, the exteriority is negative (resp. positive) if c is predominantly below (resp. above) the axis. The seeds being located clockwise, admissible paths are sorted from the innermost to the outermost (see Fig. 4.8).

Starting from an initial labeling corresponding to the most interior configuration, i.e. $(1, \dots, 1)$, labels are changed according to a local search, by iteratively testing candidate labelings. At each iteration, given a

current sequence of labels \mathcal{S} , candidate sequences are tested that differ from a single label from \mathcal{S} , by increasing labels only. For instance, if $(2, 3, 1)$ is the current sequence, candidate sequences will be $(3, 3, 1)$, $(2, 4, 1)$ and $(2, 3, 2)$. The candidate sequence leading to the smallest energy, regardless of the current energy, is chosen as the base sequence for the next iteration. The best sequence found since the beginning of the procedure is updated as well. Since increasing labels correspond to paths of increasing exteriarity, generating candidate sequences by solely increasing labels makes the contour expand monotonically. This local search is summarized in the **while** loop of Algorithm 2. A detailed procedure is given in Appendix A.2.3.

4.10 Experiments and discussion

We demonstrate the ability of the CSPP to recover closed boundaries of objects in natural color images, given a variable number of user-provided points along the target boundary. Most of the tests are carried out on the Grabcut dataset [Rother et al., 2004]. These experiments include an independent study of the CSPP, as well as comparisons against the piecewise-geodesic curve (PGC, described in Section 4.6.2), the Riverbed algorithm [Miranda et al., 2012] and the previous CDP method, in order to show the benefits brought by the use of saddle point-based paths and additional energy terms.

The experiments are limited to simply connected objects, as every tested method is designed to recover a single closed outer boundary of one object. One may note that it would be possible to extend the model such that the set of initial seeds would be separated into subsets: one subset containing the seeds on the outer boundary and several subsets containing seeds on inner boundaries. However, we chose to restrict ourselves to objects without inner boundaries.

As regards the selection of parameters, each method is assessed in the most favorable configuration. Parameters such as the regularization weight w (used in the PGC, CDP and CSPP) or the energy weights ω_{edge} and ω_{region} (for the CDP and CSPP) are tuned separately each time, in order to achieve the most relevant segmentation. The appropriate color space is also chosen for each image, i.e. RGB or the more perceptually uniform Lab, which affects potential P and the color PDFs involved in the region energy. In order for the reader to get an accurate idea of the importance of the regularization weight w in balance with the gradient magnitude in potential (4.23), notice that color components of f are normalized between 0 and 1. The maximum number K_{\max} of admissible paths per set \mathcal{A}_i is typically chosen between 3 and 10.

4.10.1 Influence of edge, region and simplicity terms

One of the main benefits of the CSPP method over usual minimal path-based segmentation algorithms is the ability to formulate a region-based criterion, as in classical active contours. It allows to formulate the simplicity term as well. Fig. 4.14 illustrates the interest of using the region and simplicity terms, in addition to the edge term. The LEAF image is processed with regularization weight w set to 0.01, and $K_{\max} = 5$. Chosen color space is RGB. When corresponding energy terms are used, weights are set to $\omega_{\text{edge}} = 1$ and $\omega_{\text{region}} = 2$, 0 otherwise. The undesirable overlapping phenomenon yielded by the piecewise-geodesic curve is shown in Fig. 4.14(b). With the edge term only (Fig. 4.14(c)), the CSPP selects the strongest contours, leading to the lowest potential, regardless of their lengths. Hence, it tends to capture any boundary in the image which is more salient than the actual object boundaries. On this particular image, where the object and background have strongly overlapping color distributions, adding the region term (Fig. 4.14(d)) yields a contour including an almost empty interior region. The inner color PDF is thus very compact, which limits the overlapping with the outer color PDF and thus minimizes the region term. A satisfactory segmentation is achieved as soon as the simplicity term is added, as depicted in Figs. 4.14(e) and 4.14(f). The combination of region and simplicity terms allows to remove the small remaining background areas that were mistakenly included into the object.

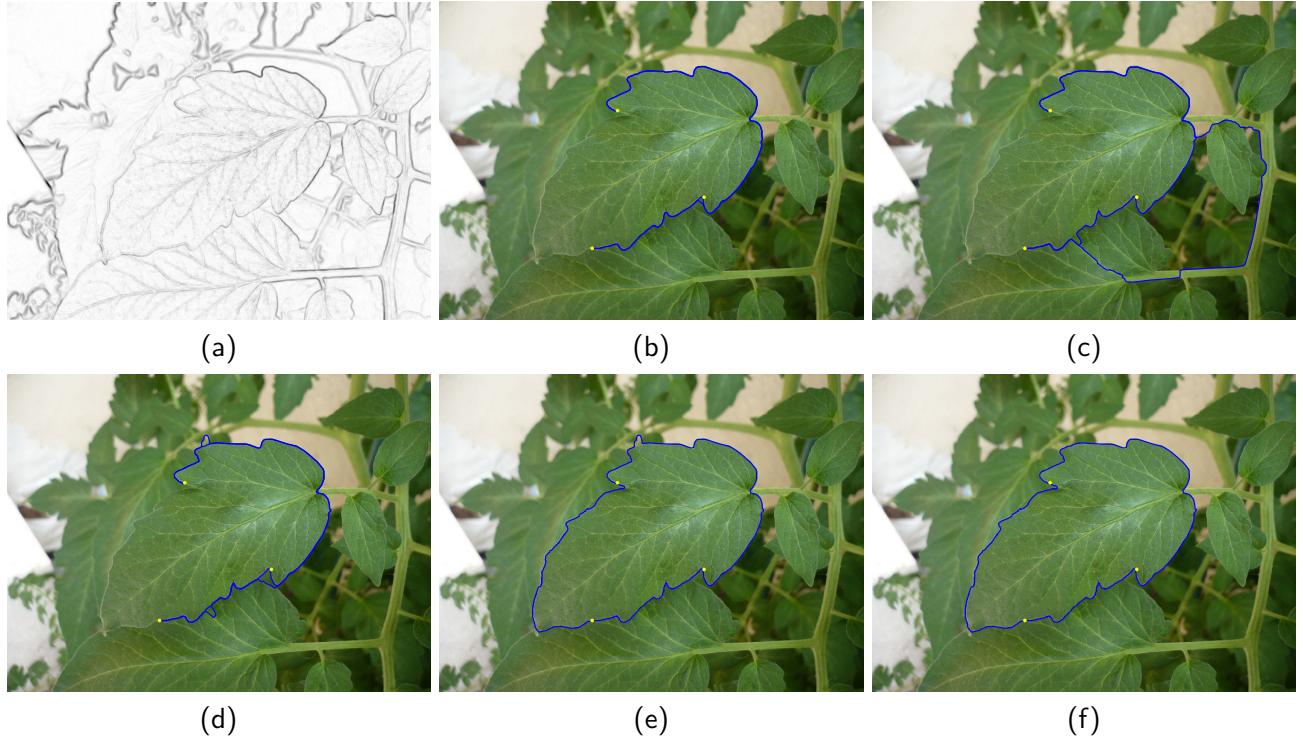


Figure 4.14: Influence of energy terms on the final segmentation: (a) Potential, comparison against (b) PGC, CSPP with (c) edge term only (d) edge and region terms (e) edge and simplicity terms and (f) edge, region and simplicity terms.

4.10.2 Overview of the Riverbed algorithm

We give a summary of the Riverbed algorithm by Miranda *et al.*, so that the reader can get the essence of this segmentation method without going into the specificities of the implementation. We especially focus on the properties which make this algorithm suitable for comparison. For further details, we refer the reader to [Miranda *et al.*, 2012].

First, it rests upon a discrete framework, as the image domain is modeled as a directed 4-connected grid graph. Miranda *et al.* developed their method with a generic edge cost function e , leaving the choice depending on the selected features of interest in the image. In order to use a cost function similar to the potential P used in our approach, we chose the edge cost $e(\mathbf{x}, \mathbf{y})$ from node (pixel) \mathbf{x} to node \mathbf{y} as

$$e(\mathbf{x}, \mathbf{y}) = P(\mathbf{y}).$$

Hence, as in our approach, favoring low-cost edges is equivalent to following high-gradient areas. A sequence of n ordered seeds is placed along the object contour. Minimum cost paths between these seeds are extracted thanks to the Image Foresting Transform (IFT) [Falcão *et al.*, 2004], which may be considered as a heuristic generalization of Dijkstra’s shortest path algorithm. In addition to computing the distance from a set of seed nodes to every other node in the grid, it generates a predecessor map, which links every node to its preceding node in the corresponding path. Let $\pi_{\mathbf{x}}$ be a path ending at node \mathbf{x} . Assuming that \mathbf{y} is a successor node of \mathbf{x} , we denote by $\langle \pi_{\mathbf{x}}, \mathbf{y} \rangle$ the extension of $\pi_{\mathbf{x}}$ by arc (\mathbf{x}, \mathbf{y}) . Instead of considering an additive cost function only,

$$C(\langle \pi_{\mathbf{x}}, \mathbf{y} \rangle) = C(\pi_{\mathbf{x}}) + e(\mathbf{x}, \mathbf{y}),$$

the IFT computes the distances and predecessors for arbitrary path cost functions. If the cost function satisfies smoothness conditions, it is proven that the distance and predecessor map are optimal. Otherwise,

they correspond to a local solution. This is the case with the Riverbed approach, which uses the following cost function:

$$C_{\text{riverbed}}(\langle \pi_{\mathbf{x}}, \mathbf{y} \rangle) = e(\mathbf{x}, \mathbf{y}).$$

Therefore, at any time, the IFT propagation algorithm decides to take the most interesting arc, regardless of the past history. As a summary of the justifications stated in [Miranda et al., 2012], it allows not to favor shortest distance paths and to avoid undesirable shortcuts. Given n initial seeds, n instances of the IFT are run. The i^{th} instance takes seed \mathbf{v}_i as the single seed node and generates a path to \mathbf{v}_{i+1} . Between two instances of the IFT, nodes belonging to previously-built paths are “frozen”, so that subsequent paths cannot pass through them. Hence, by construction, the Riverbed forbids overlapping between paths as a hard constraint. Unlike our approach, the resulting contour depends on the order in which pairs of successive seeds are considered.

4.10.3 Qualitative comparison

To keep a critical look at our contribution, Fig. 4.15 depicts typical cases of comparison between the PGC, Riverbed, CDP and CSPP algorithms, on a subset of the Grabcut dataset [Rother et al., 2004]. At this stage, we do not measure any distance between the obtained contour and ground-truth contour, the assessment being visual only. Quantitative assessment is done further in Section 4.10.4. The combination of paths may lead to strong global improvements, or slight localized improvements, or no improvement when the edge strength and locations of seeds are favorable to all segmentation algorithms. For each image, the same set of initial seeds is used.⁹.

The BANANA image (row 1) depicts a situation where the CSPP approach does not clearly improve segmentation over other algorithms, for this particular configuration of initial points. Despite the complex background and object containing many inner edges, boundaries are well defined and seeds are evenly distributed along the boundary so that the PGC and Riverbed manage to extract the object. The FLOWER (row 2) and DOLL (row 3) images are cases where the PGC exhibits strong overlapping between geodesics when few seeds are provided, although the seeds are reasonably well distributed along the object boundary. Since it does not have any non-overlapping constraint, the PGC systematically favors portions of contours with the lowest potential. Hence, smooth boundary segments are ignored, not because of their length, but because they may contain sparsely weak edges making the potential increase in small parts of the contour. Conversely, these boundary segments, despite from not being part of the minimal path, make valleys in the distance map and are very likely to be considered as parts of admissible paths by our approach. This proves the CSPP to be inherently less sensitive to weak edges. As regards the Riverbed approach, it does not generate self-overlapping contours, but tends to favor the sharpest edges in the neighboring structures of the object.

The CERAMIC (row 4) and TEDDY (row 5) images depicts situations where the PGC makes *shortcuts* through the object, due to the presence of inner edges stronger than the actual boundaries. Both simplicity and region energies contribute to solve this issue in the CSPP. The former prevents overlapping while the latter favors high discrepancy between inner and outer color distributions, hence avoiding to select the undesirable shortcuts, which would yield to less distinct color histograms than the actual boundaries would. Finally, the MUSHROOM (row 6) combines the issues of inner shortcut and strong self-overlapping.

4.10.4 Quantitative comparison with random locations of seeds

The qualitative study with user-provided seeds cannot give alone an insight of the robustness of the compared segmentation algorithms. The experienced user has the essence of the algorithms in mind, and thus tends to place seeds in favorable configurations, such as even spacing of seeds along the boundary, or dense sampling

⁹In [IJCV15], we did not include the CDP method in our qualitative comparison. This is an additional material brought by this manuscript

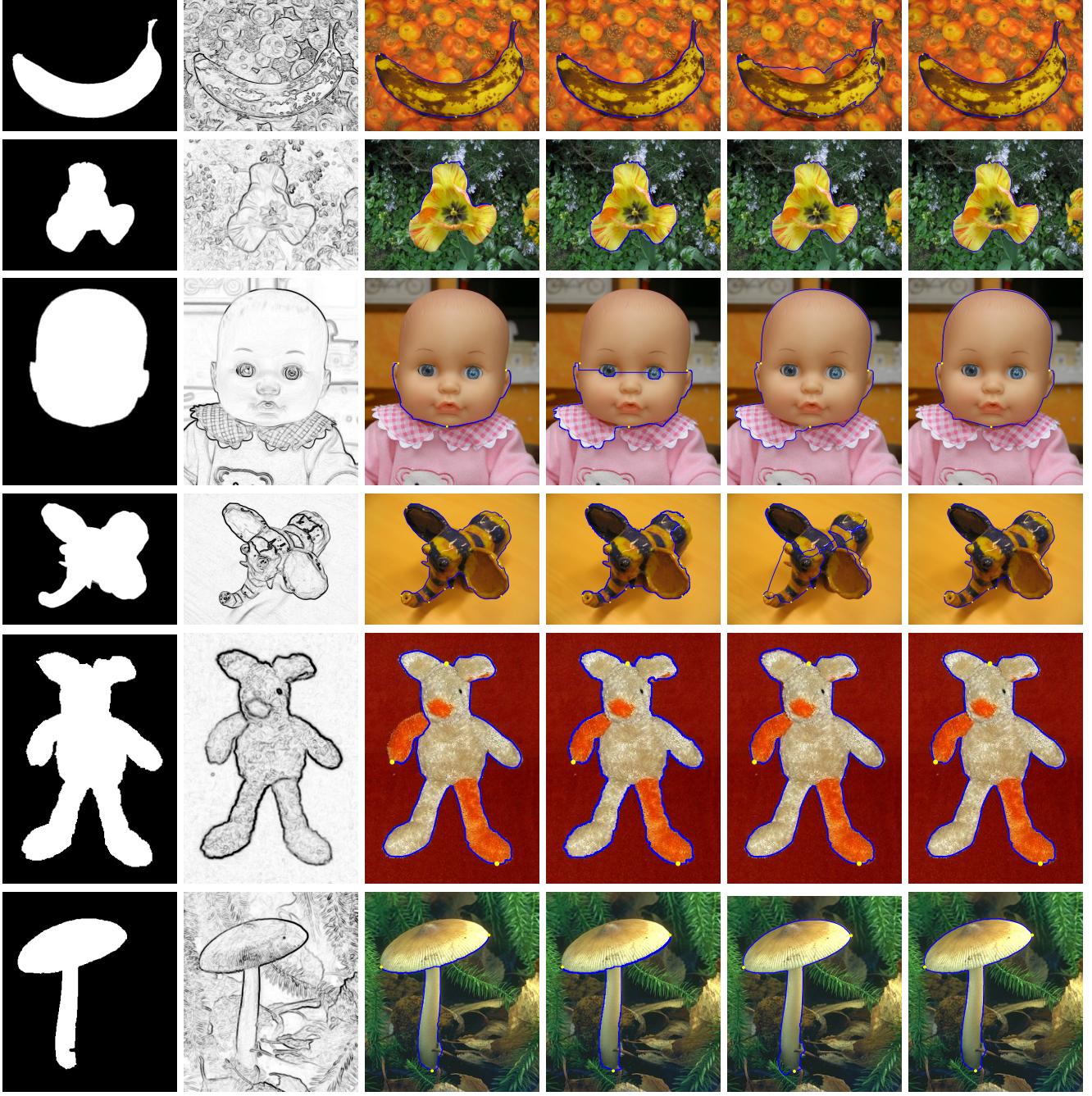


Figure 4.15: Qualitative comparison between PGC, Riverbed, CDP and CSPP on a sample of the Grabcut dataset. Column 1: ground truth segmentation, Column 2: potential, Column 3: piecewise-geodesic curve (PGC), Column 4: Riverbed, Column 5: CDP, Column 6: CSPP.

along blurry edges and sharp turns of the boundary.

In order to provide an evaluation less dependent of the user-induced bias, we assess the robustness of our method by running several tests where seeds are initialized randomly along the contour. The actual object boundary is extracted from the available binary ground truth image and a random start position u_1 along the curve is generated. The number of seeds n being fixed for each image, the ground truth contour is split into n segments of equal length, starting from u_1 . Each segment is subsequently split into two subsegments: a free subsegment, covering the largest part of the segment, and a safety margin of a few pixels (typically, in the order of 10 pixels). Each seed v_i is randomly generated within the i^{th} free subsegment, the safety margins preventing two successive seeds from getting excessively close. In this way, the distribution of seeds

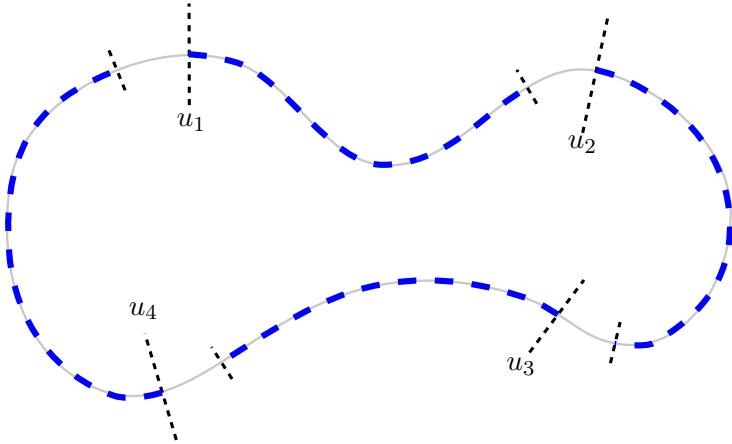


Figure 4.16: Splitting the ground truth contour into segments of equal length for random locations of initial seeds: example with $n = 4$.

roughly covers the entire contour, without being necessarily evenly spaced. Fig. 4.16 illustrates the splitting of a contour with $n = 4$. Segments are separated by long dashed lines normal to the curve and free subsegments are highlighted with dashed blue lines. For each image, we generate 20 random configurations of seeds. The same random configuration is used for initializing the four algorithms. Let $S \subset \mathbb{Z}^2$ and $G \subset \mathbb{Z}^2$ be the segmented object and ground truth object, respectively. The estimated object S is obtained by discretizing the n selected admissible paths on the image grid - see appendix A.2.2 for details on the discretization - and flood-filling from the outside, *i.e* starting from any point on the image border. S is then the set of pixels unreach by the filling. Segmentation accuracy is assessed by measuring the overlap between regions, using the intersection-over-union ratio,

$$J(S, G) = \frac{|S \cap G|}{|S \cup G|}, \quad (4.38)$$

also known as the Jaccard index. The minimum, maximum and average values of the Jaccard percentage, $100J$, computed over the 20 runs per image, are listed in Table 4.1. The best segmentation for each image, yielding the maximal Jaccard score, is shown in Fig. 4.17. Notice the distribution of seeds, which may be unevenly spaced along the object boundary, thus lending itself to assess the robustness of the algorithm with respect to unfavorable initial conditions.

If one considers the 'Min' columns in Table 4.1, it appears that segmentation accuracy can be almost null in the least favorable cases, for every algorithm. For a start, there are no image for which the poorest segmentation obtained with CDP and CSPP is significantly worse than the poorest segmentations obtained with the PGC and Riverbed algorithm. On the contrary, the CSPP even guarantees a higher standard, since there is an important proportion of images where the worst segmentation is improved by the CSPP. As can be seen by considering simultaneously the 'Max' columns in Table 4.1 and the contours in Fig. 4.17, Jaccard percentage values around 80% correspond to segmentations where important parts of the object are missing or background areas are mistakenly included into the object. As a general remark regarding the maximum Jaccard percentages, they are most often obtained with Riverbed, CDP or CSPP. The improvement in the maximum quality can be seen on a small subset of images, such as the BANANA3 and CERAMIC data, but cannot be generalized to the entire dataset. One may get an idea of the most significant improvement brought by our approach by considering the average Jaccard percentages, both on each image independently and on the global score. The CSPP leads to highest average score, which conveys the overall robustness of the CSPP approach with respect to initial conditions.

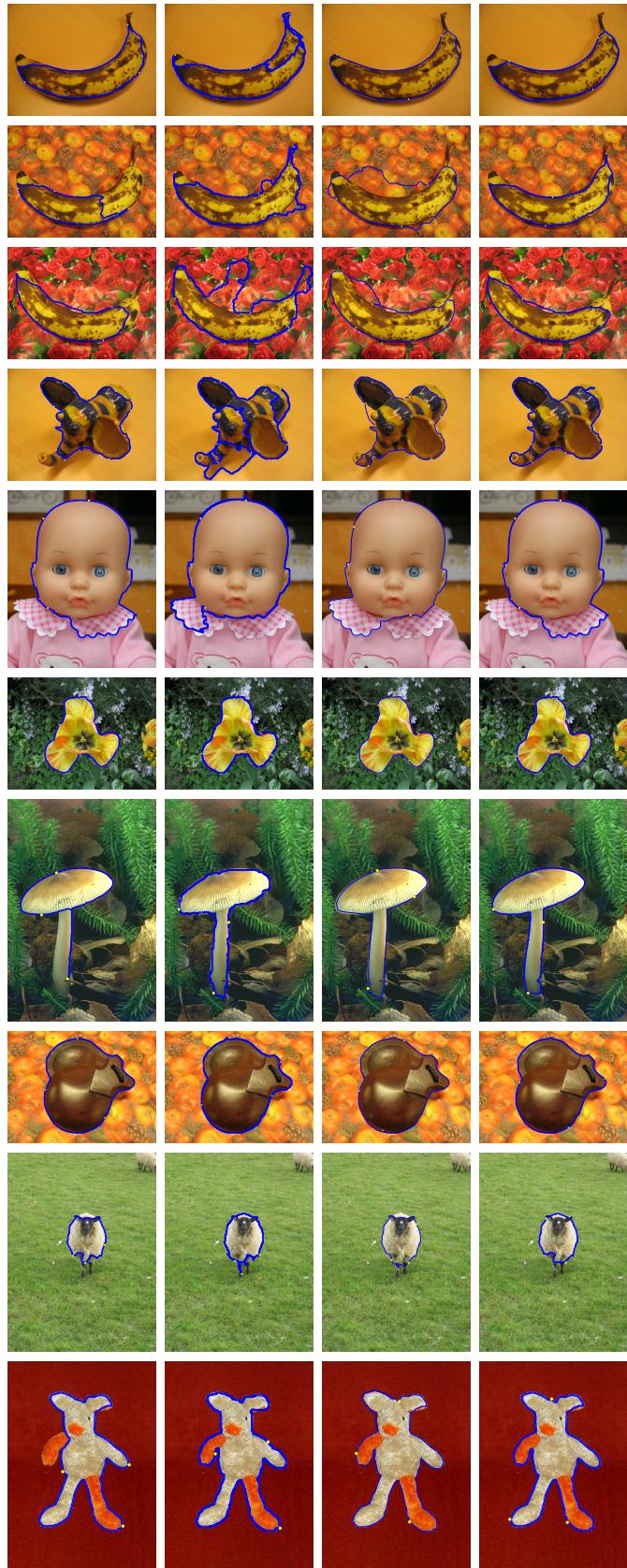


Figure 4.17: Best segmentations (leading to the maximum Jaccard segmentation accuracy index) over 20 runs per image, with randomly-located initial seeds. Column 1: PGC, Column 2: Riverbed, Column 3: CDP, Column 4: CSPP

Image	PGC			Riverbed			CDP			CSPP		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
BANANA1	1.0	84.0	23.3	5.9	88.8	53.5	17.0	91.4	65.1	17.0	89.3	71.7
BANANA2	0.8	55.5	11.4	1.8	84.6	39.5	0.1	71.6	35.6	0.1	82.9	44.6
BANANA3	0.2	76.6	32.6	2.7	72.2	43.3	1.2	83.1	52.1	1.2	81.5	55.7
CERAMIC	77.1	89.0	82.2	59.2	78.8	64.2	51.6	90.7	83.3	85.3	93.3	88.4
DOLL	0.3	87.6	45.0	0.7	90.7	62.0	55.2	87.0	66.6	78.9	87.6	81.0
FLOWER	1.0	98.2	20.9	2.0	98.1	80.3	1.0	98.0	68.0	1.0	98.3	82.8
MUSHROOM	1.4	65.3	15.0	1.5	91.5	37.7	29.6	92.3	67.9	22.0	91.1	66.4
MUSIC	0.7	98.6	83.9	0.9	98.5	47.7	93.7	98.6	96.8	97.3	98.6	97.9
SHEEP	1.6	83.9	11.5	36.5	90.7	54.0	52.3	89.4	71.4	55.9	89.3	78.2
TEDDY	2.0	89.2	55.8	88.5	96.3	90.5	51.9	95.8	75.7	51.9	96.9	79.3
Overall	0.2	98.6	38.1	0.7	98.5	57.3	0.1	98.6	68.3	0.1	98.6	74.6

Table 4.1: Quantitative comparison on a subset of the Grabcut dataset: minimum, maximum and average values of the Jaccard segmentation accuracy percentage computed over 20 runs per image, with randomly-located initial seeds, for the PGC, Riverbed, CDP and CSPP.

4.10.5 Local search versus brute-force search

The search method described in Section 4.9.2 minimizes the energy over combinations of admissible paths in a heuristic way. In order to assess the efficiency of the local search against the exhaustive search of the CDP, we should determine whether the local minimum obtained is significantly worse than the actual optimal solution found by a brute-force algorithm. As in the previous experiments, we perform 20 runs per image with randomly selected initial seeds, reporting the Jaccard index, energy and computation time obtained with both search methods. Statistics of Jaccard coefficients are listed in Table 4.2. Notice that this is not a comparison between CDP and CSPP, as we wish to challenge the search method only. The construction of paths and energies are the ones of the CSPP. Notice also that the Jaccard coefficients for the CSPP with local search are not equal to the ones appearing in Table 4.1, as they correspond to another sequence of runs, with different randomly located seeds. Statistics on Jaccard coefficients show that the local search does not significantly damage segmentation quality over the exhaustive search.

As regards combinations of paths and their energies, on the same dataset, checking final combinations shows that the local search finds the same combination of admissible paths as the brute-force method in 65% of the cases. Obviously, for the remaining 35% of cases, the combinations determined by the local search have greater energies than the ones found by the brute-force algorithm. However, it happens that the local search leads to a better segmentation quality. This is a phenomenon that arises in many optimization-based segmentation methods, such as active contours, as the global minimum of the energy does not necessarily correspond to the best expected segmentation.

As regards computation time, the generation of admissible paths (i.e. propagating using Fast Marching, detecting saddle points, building paths and sorting them with respect to their exteriorty) is relatively fast. The major part of the computational cost lies in the search for the best combination of admissible paths, which highly depends on the number of initial seeds n and the upper bound K_{\max} on the number of admissible paths. This significantly favors the heuristic local search over the brute-force algorithm. Reported execution times are obtained with a C++ implementation running on a standard Intel Core2 Duo 2.8GHz architecture with 4Gb RAM. The MUSHROOM image, with $n = 3$ and $K_{\max} = 5$, was fully processed in 3s300ms with the local search method and 12s with the brute-force search (both including 400ms to generate admissible paths). The CERAMIC image, with $n = 7$ and $K_{\max} = 5$, was fully processed in 17s with the local search method and 1h10mn23s with the brute-force search (both including 2s to generate admissible paths).

Image	CSPP Local search				CSPP Brute-force search			
	Min	Max	Avg	Std	Min	Max	Avg	Std
BANANA1	20.0	89.1	60.4	0.20	20.0	89.1	59.2	0.20
BANANA2	0.7	88.3	47.3	0.25	0.7	91.1	52.2	0.31
BANANA3	26.1	86.6	62.5	0.15	26.1	86.6	59.7	0.18
CERAMIC	74.8	89.8	85.6	0.03	74.4	88.1	85.0	0.05
DOLL	72.5	87.7	80.8	0.04	78.9	87.7	82.6	0.07
FLOWER	1.4	98.2	88.1	0.29	1.3	98.2	88.4	0.29
MUSHROOM	33.0	91.1	61.3	0.17	25.3	86.3	56.6	0.19
MUSIC	97.3	98.6	97.8	0.01	97.3	98.6	97.9	0.05
SHEEP	4.5	90.2	77.0	0.18	2.9	90.2	76.2	0.19
TEDDY	17.6	96.7	74.9	0.17	17.6	96.7	74.9	0.17
Overall	0.7	98.6	73.5	0.23	0.7	98.6	73.3	0.24

Table 4.2: Quantitative comparison between the CSPP and a modified CSPP with brute-force search: minimum, maximum, average and standard deviation values of the Jaccard segmentation accuracy percentage computed over 20 runs per image.

4.11 Conclusion

4.11.1 Summary of contributions

In this chapter, we addressed user-guided segmentation of a single simply-connected object. At initialization, the user provides an ordered sequence of n seed points on the object boundary, and the segmentation problem amounts to link successive seeds with paths, so that the concatenation of paths fits to the actual object contour. We raised the following questions:

- how to build relevant candidate paths between successive seeds?
- how to formulate the energy on the concatenation of paths?
- how to minimize this energy?

The first contribution was the piecewise-geodesic curve (PGC). In this naive method, there is a unique candidate path for each pair of successive seeds, being the minimal path. We did not perform any further energy minimization, as the concatenation of minimal paths was directly taken as the final closed contour. We highlighted the obvious shortcomings of such approach. The resulting closed contour may not be simple. Moreover, if it is simple, it does not necessarily make a relevant object/background partition, as no region-based energy can be optimized.

Hence, we switched to the notion of admissible path. A path between two seeds may be optimal according to the edge-based cost function, but might turn out irrelevant when assembled with paths emanating from other seeds to build a closed contour. Thus, the key idea was to generate several candidate paths for each pair of successive seeds.

The second approach - the first one dealing with admissible paths - was the Combination of Disjoint Paths (CDP). A fixed number K of disjoint paths is extracted per pair of successive seeds. The K^n possible combinations are tested by a brute-force search. The selected combination is the one minimizing the energy, made up of an edge-based term, a region-based term as well as a novel term penalizing self-overlap, in order to favor simple curves. Shortcomings of disjoint paths, the overlap energy, and the exponential brute-force search were discussed.

The final method, the Combination of Saddle Point-based Paths (CSPP), solved these limitations. Instead of calculating disjoint paths, we extract admissible paths by inspecting the saddle points of the geodesic distance map between two endpoints. These saddle-points are the midpoints of the different valleys connecting the two endpoints. The overlap energy was extended to the more general simplicity energy, taking self-intersections into account. In order to avoid testing all combinations, a local search method was investigated.

Comparison between the PGC, CDP and CSPP, as well as the competing Riverbed algorithm, showed the advantages of the CSPP, in terms of segmentation quality. As regards computational cost, we also showed the interest of the local search against the brute-force one.

4.11.2 Possible research tracks

The CDP and CSPP methods were developed over the 2012-2014 period. Along with main collaborator Sébastien Bougleux, before the long-version paper [IJCV15] and code¹⁰ were published, we envisioned several mid-term extensions that could be investigated. At the time of writing this manuscript, minimal paths and Fast Marching-related methods is a very specific subfield of image processing, still investigated by a handful of researchers [Chen et al., 2017, Chen and Cohen, 2018, Mirebeau and Portegies, 2019, Liu et al., 2020].

Orientation-dependent potential The input sequence of seeds $(\mathbf{v}_i)_{1 \leq i \leq n}$ is constrained to be ordered clockwise¹¹. Thus, paths selected in the final combination travel, along the object boundary, clockwise as well. However, no orientation constraint is formulated on minimal paths of the PGC, nor on admissible paths of the CDP and CSPP methods. The interior and exterior of the target object are unknown at the stage of construction of paths. Some issues, raised in the experiments reported in Section 4.10, arise from the fact that some shortest or admissible paths travel counterclockwise. This is especially visible for the PGC (see column 3 in Fig. 4.15).

In order to constrain the PGC to be a simple curve, and avoid generating irrelevant admissible paths in the CDP and CSPP, the potential could be revisited. Indeed, the potential in Eq. (4.23) is a function of position only, and is thus direction-independent. One could consider generalizing to a direction-dependent potential $P : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^{*+}$, extending the cost functional (4.22) to

$$C[\mathbf{c}] = \int_0^1 P(\mathbf{c}(u), \mathbf{c}'(u)) du.$$

Special cases of this general potential were studied in [Bougleux et al., 2008, Chen et al., 2017], under the form

$$P(\mathbf{c}(u), \mathbf{c}'(u)) = \sqrt{\mathbf{c}'(u)^\top \mathbf{M}(\mathbf{c}(u)) \mathbf{c}'(u)},$$

where $\mathbf{M} : \mathcal{D} \rightarrow \mathbb{R}^{2 \times 2}$ is a symmetric positive definite 2×2 matrix, embodying the *metric*. Such form is orientation-independent, as we have $P(\mathbf{c}, -\mathbf{c}') = P(\mathbf{c}, \mathbf{c}')$. Given a reference intensity/color f_{ref} inside the object, we should consider an orientation-dependent potential penalizing curves having image data far from f_{ref} on their right side and close to f_{ref} on their left side, in compliance with the definition of the inward normal before Eq. (4.2). An intuitive choice would be

$$P(\mathbf{c}, \mathbf{c}') = g(\mathbf{c}) + \beta ((f(\mathbf{c} + \epsilon \mathbf{c}') - f_{\text{ref}})^2 - (f(\mathbf{c} - \epsilon \mathbf{c}') - f_{\text{ref}})^2),$$

with some weight β and offset ϵ . This would however require an extra interaction, since the user should mark an area inside the object to provide f_{ref} .

¹⁰<https://github.com/julien-mille/comb-pgeo-paths>

¹¹in the standard computer left-handed coordinate system

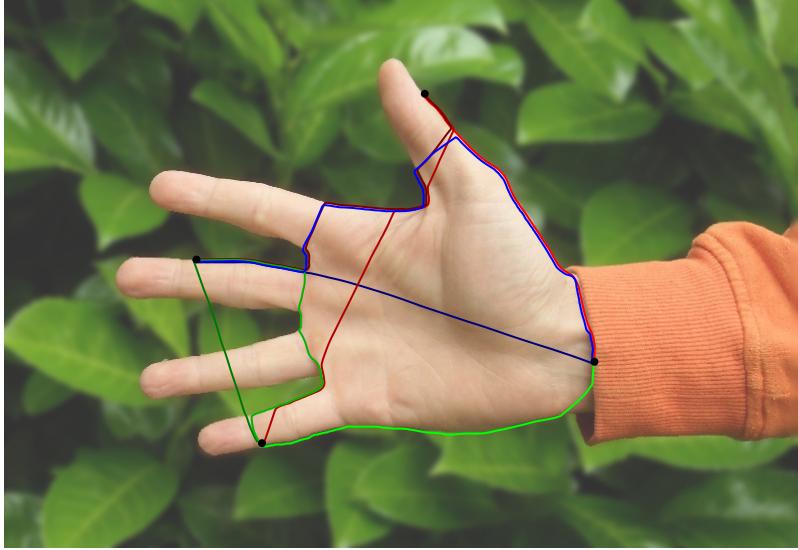


Figure 4.18: If the user-provided input was an unordered set of seeds, admissible paths could be generated for every seed pair

Unordered set of seeds as input Another interesting challenge would consist in throwing off the clockwise-ordering constraint, *i.e.* considering the input as an **unordered** set of seeds $\{v_i\}_{1 \leq i \leq n}$, and let the algorithm decide of the most relevant ordering. This would allow true interaction, instead of prior interaction only. Seeds could be moved or added, without needing to specify their position in the sequence, and let the user iteratively improve the segmentation.

Without any seed ordering, it would be relevant to build admissible paths for every pair (v_i, v_j) , as illustrated in Fig. 4.18. If we connect each seed v_i to every other seed v_j , with $j > i$, and limit the number of admissible paths for each pair to K_{\max} , we have

$$K_i = O(K_{\max}(n - i)).$$

The total number of admissible paths would be bounded above, as

$$\sum_{i=1}^n K_i = O\left(\frac{K_{\max}n(n-1)}{2}\right).$$

Considering an undirected graph \mathcal{G} , with seeds as vertices, and admissible paths as edges, \mathcal{G} would be a *complete multigraph*¹². Building a simple closed curve would amounts to determine an Hamiltonian cycle in \mathcal{G} , still minimizing a weighted sum of region, edge and simplicity energies. This extension could be seamlessly combined with the previous one. The number of Hamiltonian cycles in a simple complete graph with n vertices is

$$\frac{(n-1)!}{2},$$

according to e.g. [Cormen et al., 2009, Chapter 34]. If we have at most K_{\max} edges per vertex pair, it is very likely that the number of Hamiltonian cycles in a complete multigraph with n vertices is in

$$O\left(\frac{K_{\max}^n(n-1)!}{2}\right),$$

which would dramatically increase the complexity in comparison to the current problem. Approximation algorithms [Cormen et al., 2009, Chapter 35] should be considered. These are rather immediate extensions. Longer-term perspectives, especially the possible link between interactive segmentation and on-going work, are developed in Chapter 7.

¹²as opposed to a *simple* graph, which has at most one edge per vertex pair

Chapter 5

Objects as moving entities: video object segmentation with background modeling

Segmenting objects in image sequences has many applications, for example, in video surveillance or traffic estimation. In this context, an *object* is any moving entity of interest in the video. The shape of nonrigid, moving objects may vary a lot along image sequences due to, for instance, deformations or occlusions, which puts additional constraints on the segmentation process. In the recent survey [Yao et al., 2020], a pixelwise labeling of one or several object(s) in each frame of a video is referred to as *video object segmentation* (VOS). We will use the same acronym throughout the chapter.

We describe two main contributions, which are both based on novel modelings of the background. The first one arises from a collaboration with Jean-Loic Rose during the 2009-2011 period, who was assistant professor at the LIRIS. It extends elements developed in my PhD, on region-based active contours [CVIU09]. The VOS task is performed on the entire video with an initially-labeled first frame. This contribution does not address the *detection* part, as new objects appearing during the video are not taken into account. Temporal consistency of segmentation is addressed using color distributions inside the objects and background. Our method shares common ideas with probabilistic color tracking [Pérez et al., 2002] or with the mean shift tracker [Comaniciu et al., 2003], in the extent we seek for regions in which color statistics match references resulting from previous frames. Related publications in Section 1.5 are [BMVC11, ICIP11].

The second contribution is related to the supervision of PhD student Imtiaz Ali, who defended in early 2012. This work deals with a background subtraction method, which integrates both detection and segmentation problems. As in most of background subtraction techniques [Stauffer and Grimson, 2000, Elgammal et al., 2002], detection is performed by measuring the mismatch between a pixelwise appearance model and a reference background model. The pixel is classified as object if the mismatch is above a given threshold. Semantic objects are built as connected components of pixels marked as 'object'. The PhD of Imtiaz Ali focused on tracking objects within backgrounds exhibiting local periodic motions, e.g water waves. The proposal is a novel background modeling method based on frequency, well suited to spatiotemporal texture patterns. The local Fourier Transform was applied to construct a pixel-wise representation of local frequency components. The related publication in Section 1.5 is [PRL12].

Section 5.1 is a survey on VOS from two different perspectives. The first one deals with methods based on a labeled initial frame, the detection of new objects being ignored. The second perspective integrates the detection task. Section 5.3 presents the Bayesian framework which is at the core of our first contribution. Sections 5.4 and 5.5 describe our contributions.

5.1 Video object segmentation

In the highly-cited object tracking survey of [Yilmaz et al., 2006, Section 2], the VOS task is called *contour* or *silhouette tracking*. Notable examples also refer to this task as *joint segmentation and tracking* [Aeschliman et al., 2010, Wen et al., 2015] or *region tracking* [Mansouri, 2002]. These designations all refer to a precise pixelwise object/background labeling, as opposed, for example, to object tracking with bounding boxes or ellipses.

To formulate the input, we add a discrete time variable to the vector-valued image defined in Eq. (3.2). The video is a color¹ image sequence $(\mathbf{f}_t)_{t=1\dots T}$, with T frames, such that, for each $t = 1 \dots T$,

$$\begin{aligned}\mathbf{f}_t : \mathcal{D} \subset \mathbb{R}^2 &\longrightarrow \mathcal{Q} \subset \mathbb{R}^3 \\ \mathbf{f}_t(\mathbf{x}) &= [f_{t,1}(\mathbf{x}) \ f_{t,2}(\mathbf{x}) \ f_{t,3}(\mathbf{x})]^\top.\end{aligned}$$

The general VOS task can be stated as determining the labeling of each frame t in at most $n + 1$ domains, where label 0 corresponds to the background, and remaining labels $i = 1 \dots n$ correspond to the n objects. The labeling can be written as a sequence of integer-valued functions $(\phi_t)_{t=1\dots T}$,

$$\phi_t : \mathcal{D} \longrightarrow \{0, \dots, n\} \quad (5.1)$$

The background or i^{th} object at time t is

$$\Omega_t^i = \{\mathbf{x} | \phi_t(\mathbf{x}) = i\} \subset \mathcal{D}.$$

In the most general form of VOS, n should be understood as the total number of distinct objects in the entire video. Region Ω_t^i can be empty (for $i > 0$), as the i^{th} object may appear in a subsequence of frames only, and not be present in frame t . Moreover, it is not necessarily simply-connected, as objects may have arbitrary topology.

5.1.1 VOS with initial annotation

The first class of methods we review takes as input a labeling of the first frame. We thus consider that ϕ_1 is given, and the number of objects n is fixed at initialization. The problem of detecting new objects, which might appear in the scene, is ignored. In the survey [Yao et al., 2020], this is known as *semi-supervised video object segmentation*. If each of the n objects is known to appear in each frame, an additional constraint is set beside Eq. (5.1),

$$\{\mathbf{x} | \phi_t(\mathbf{x}) = i\} \neq \emptyset, \forall t \in \{1, \dots, T\}, \forall i \in \{0, \dots, n\}.$$

In the first frame, the user-provided indication about objects can be a fine annotation, as in [Mansouri, 2002]. In this case, to construct ϕ_1 , the annotation can be taken as it is. The indication can also be a coarse annotation, as in [Paragios and Deriche, 2005, Fundana et al., 2008], e.g. scribbles inside the object, or a bounding rectangle/ellipse around the object. In order to construct ϕ_1 , this coarse annotation is evolved to a fine annotation, by minimization of an energy (see Sections 4.3 and 4.4). Note that this user-guided segmentation in the first frame makes assumptions on image distribution within the objects and the background, which were discussed in Section 4.4.2, and does not use any temporal information.

We focus on *causal* joint segmentation and tracking, which means that, at time t , the next labeling (at time $t + 1$) is estimated from the past labelings and observations, considering that future observations are not available yet. The estimation of labelings is thus sequential, which allows on-line VOS². At time t , this can

¹General models and mathematical derivations are done with vector-valued images, the simplification for grayscale images being trivial

²as opposed to off-line VOS, which considers that the entire sequence is available from the beginning

be set up as the maximization of posterior probability of ϕ_{t+1} given previous labelings and images³,

$$\phi_{t+1}^* = \operatorname{argmax}_{\phi_{t+1}} \Pr(\phi_{t+1} | \mathbf{f}_{1:t+1}, \phi_{1:t}), \quad (5.2)$$

where indexing by $1 : t$ corresponds to the full image or labeling sequence from time 1 to time t . The problem can also be formulated as an energy minimization,

$$\phi_{t+1}^* = \operatorname{argmin}_{\phi_{t+1}} E[\phi_{t+1} ; \phi_{1:t}, \mathbf{f}_{1:t+1}],$$

without coming from a probabilistic basis. At this point, the reader should think of the probability, or energy, as a very general and abstract way of formulating the relevance of ϕ_{t+1} with respect to past labelings and images. We do not yet give its explicit form, and how it can factorize into likelihood and prior probability. Mathematical derivations leading to a more explicit form are given in Section 5.3. For now, it is enough to know that the probability encompasses one or several data-fitting term(s), depending on $\phi_{1:t}$ and $\mathbf{f}_{1:t+1}$, and one or several prior term(s), which depend(s) only on $\phi_{1:t}$.

A large variety of data-fitting terms has been proposed and applied to VOS. They can be classified in the same categories that are used to classify tracking features in survey [Yilmaz et al., 2006, Section 3]:

- *color* of individual pixels, as in [Chiverton et al., 2008, Freedman and Zhang, 2004, Fundana et al., 2008]. The data-fitting term enforces matching between color distribution of objects and/or background through time. It can also encourage dissimilarity between the color distributions of objects and the one of the background. Color densities can be estimated in a non-parametric [Chiverton et al., 2008, Freedman and Zhang, 2004] or parametric way [Fundana et al., 2008]. Our first contribution, presented in Section 5.4, belongs to the category of non-parametric color density-based VOS. In order to position our method with respect to existing work, we give a detailed study of color density-based VOS methods in Section 5.2.
- *superpixels*, as in [Wen et al., 2015]. Instead of considering individual pixels in the matching term, groups of nearby pixels can be exploited. Current and next frames are oversegmented into superpixels [Achanta et al., 2012], which are groups of neighboring pixels sharing common image properties. A superpixel can be involved in the matching term using a representative color, e.g. its mean, or texture attributes. This drastically reduces the number of variables in the labeling, as the assignment to regions Ω_{t+1}^i needs to be made on superpixels instead of pixels. One critical aspect is that superpixels should gather pixels from the same region.
- *keypoints*, as in [Aeschliman et al., 2010, Wang et al., 2019]. Interest points are detected in current and next frames, using a standard keypoint detector, like SIFT [Lowe, 2004]. These points can serve as an additional constraint for segmentation, in the sense that keypoints detected at time $t + 1$, matched to keypoints belonging to region Ω_t^i , should be assigned to Ω_{t+1}^i . Keypoints lend themselves to the construction of long-term trajectories [Wang et al., 2019].
- *optical flow*, as in [Bertalmio et al., 2000, Brox et al., 2006]. At each frame, a dense flow field representing the individual motion of pixels [Horn and Schunck, 1981, Barron et al., 1994, Baker et al., 2011] is estimated. In this case, in addition to joint segmentation and tracking, motion estimation is done as well. The flow field is expected to be piecewise-smooth, i.e. smooth in each region Ω_{t+1}^i . The matching of optical flow between Ω_{t+1}^i and Ω_t^i is done in conjunction with color matching.

Note that image properties of individual pixels or texture components of pixel neighborhoods are often gathered under the designation *appearance*, as opposed to shape and motion. The different features listed, and the data-fitting terms they generate, are not mutually exclusive. For example, the joint segmentation and tracking method of [Aeschliman et al., 2010] uses simultaneously color of individual pixels and keypoints.

³We use a condensed notation for probabilities. It should be understood that $\Pr(a|b)$ is an abbreviation of $\Pr(A = a|B = b)$, where A and B are random variables, and a and b respective realizations of these variables

5.1.2 VOS with detection: background models

VOS algorithms integrating detection do not require any user input. In the survey [Yao et al., 2020], these algorithms are referred to as *unsupervised video object segmentation*. They assume that objects to be detected have different appearance and/or motion than the background, which requires a reference background model to be available. The class of methods which consists in marking pixels as object, when their dissimilarity with some reference background model exceeds some tolerance threshold, is called *background subtraction*. The short survey in this section focuses on this kind of methods. For detailed surveys on background modeling, the reader can refer to [Bouwmans et al., 2008, Bouwmans, 2014, Brutzer et al., 2011, García García and Bouwmans, 2020].

Background subtraction methods became popular with the early work of [Wren et al., 1997]. In their simplest form, they assign pixels either to an object class - or foreground - or background class. Identification of separate objects is not taken into account. In this context, the labeling in Eq. (5.1) is considered with $n = 1$. If an identification of objects in separate frames is required, working with an arbitrary n , the detection method needs to be augmented with a pixel grouping method. The simplest method is that pixels in the same connected component⁴ will form a specific object. If, in addition, one wishes to know the correspondence between objects detected at time t and those detected at time $t + 1$, a matching procedure should be applied [Yilmaz et al., 2006, Section 5].

Foreground detection can be done by learning a pixelwise statistical representation of the background. At each point \mathbf{x} , let us assume that a reference color distribution can be learnt. During the video, if $f_t(\mathbf{x})$ deviates from this reference distribution, $\phi_t(\mathbf{x})$ is set to 1.

Background models can be parametric. In a fixed-camera setting, with a purely static background, color distributions would be unimodal. A single Gaussian per pixel would be sufficient to account for the reference color and Gaussian noise, as in [Wren et al., 1997]. However, these ideal conditions are rarely met in real-world sequences. If one imagines a countryside landscape under a sunny sky, with a bit of wind, a pixel \mathbf{x} located on a boundary between the sky and a tree leafage can be alternatively blue and green, and the plot of color distribution at \mathbf{x} during the video would exhibit two modes. Background models based on mixtures of Gaussians [Stauffer and Grimson, 2000, Lee, 2005] were notable examples taking these multimodal distributions into account.

A stationary pixelwise Gaussian-mixture background model can be defined in the following manner. For each $\mathbf{x} \in \mathcal{D}$, the background model is made up of K weighted multivariate normal distributions,

$$\text{BM}(\mathbf{x}) = (\omega_j(\mathbf{x}), \boldsymbol{\mu}_j(\mathbf{x}), \boldsymbol{\Sigma}_j(\mathbf{x}))_{j=1 \dots K},$$

where $\omega_j(\mathbf{x}) \in [0, 1]$, $\boldsymbol{\mu}_j(\mathbf{x}) \in \mathcal{Q} \subset \mathbb{R}^3$, $\boldsymbol{\Sigma}_j(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$. Weights ω_j sum to 1. The probability of an image value \mathbf{q} being explained by the background model at position \mathbf{x} is

$$\begin{aligned} \Pr(f_t(\phi_t(\mathbf{x}) = 0) = \mathbf{q} | \mathbf{x}) &= \sum_{j=1}^K \omega_j(\mathbf{x}) G(\mathbf{q}; \boldsymbol{\mu}_j(\mathbf{x}), \boldsymbol{\Sigma}_j(\mathbf{x})) \\ &= \sum_{j=1}^K \omega_j(\mathbf{x}) \frac{1}{(2\pi)^{\frac{3}{2}} \det(\boldsymbol{\Sigma}_j(\mathbf{x}))^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (\mathbf{q} - \boldsymbol{\mu}_j(\mathbf{x}))^\top \boldsymbol{\Sigma}_j(\mathbf{x})^{-1} (\mathbf{q} - \boldsymbol{\mu}_j(\mathbf{x})) \right). \end{aligned}$$

This stationary model would remain unchanged during the image sequence. Intuitively, one could think of setting $\phi_t(\mathbf{x}) = 1$ if $\Pr(f_t(\mathbf{x}) = \mathbf{q} | \phi_t(\mathbf{x}) = 0)$ fell below a threshold.

Actually, [Stauffer and Grimson, 2000] defined a time-varying (non-stationary) pixelwise background model, to account e.g. for global illumination changes and moving shadows. The components of the Gaussians are

⁴4-pixel or 8-pixel connectivity is generally considered

indexed by t ,

$$\text{BM}_t(\mathbf{x}) = (\omega_{t,j}(\mathbf{x}), \boldsymbol{\mu}_{t,j}(\mathbf{x}), \boldsymbol{\Sigma}_{t,j}(\mathbf{x}))_{j=1\dots K}.$$

Initial parameters are estimated using the EM algorithm on the N first frames of the video, where it is assumed that no object is present. The sample means and covariances are taken as the reference means $\boldsymbol{\mu}_j$ and covariances $\boldsymbol{\Sigma}_j$. To label (\mathbf{x}, t) , color $\mathbf{f}_t(\mathbf{x})$ is matched to each of the Gaussians. [Stauffer and Grimson, 2000] considers that a match happens when the Mahalanobis distance is below a threshold. Parameters of matched Gaussians are updated using a moving average scheme. Doing so, a moving object that stops at some place progressively becomes part of the background. This makes sense as only moving objects are of interest.

Conversely, in non-parametric background models like [Elgammal et al., 2002], no assumption is made about the underlying color distribution. Classical kernel density estimation is used. The background model at point \mathbf{x} is

$$\text{BM}(\mathbf{x}) = (\mathbf{q}_j(\mathbf{x}))_{j=1\dots K},$$

where K is the "memory" of the model, i.e. the number of stored significative image values. The probability of an image value \mathbf{q} being explained by the background model at position \mathbf{x} is estimated as⁵

$$\Pr(\mathbf{f}_t(\mathbf{x}) = \mathbf{q} | \phi_t(\mathbf{x}) = 0) = \frac{1}{K} \sum_{j=1}^K G_\sigma(\mathbf{q} - \mathbf{q}_j(\mathbf{x})),$$

Again, $\phi_t(\mathbf{x})$ is set to 1 when $\Pr(\mathbf{f}_t(\mathbf{x}) = \mathbf{q} | \phi_t(\mathbf{x}) = 0)$ is below a threshold.

So far, we reviewed background models based on densities of colors at individual pixels. They do not account for the spatial layout of neighboring pixels. To account for the spatial patterns that can occur in videos, texture-based background models were addressed. At the core of these methods is the dynamic texture model of [Doretto et al., 2003]. A dynamic texture is a Linear Dynamical System (LDS) representing the evolution of an entire image or a small patch,

$$\begin{cases} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{y}_t + \mathbf{v}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{w}_t, \end{cases} \quad (5.3)$$

where

- $\mathbf{y}_t \in \mathbb{R}^m$ is the observation, i.e. the full image or an image patch, flattened as a vector. For example, if we consider 5×5 grayscale patches, $m = 25$.
- $\mathbf{x}_t \in \mathbb{R}^n$ represents the state of the texture process, with $n \ll m$.
- $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the *state-transition matrix*
- $\mathbf{C} \in \mathbb{R}^{m \times n}$ is the *observation matrix*
- $\mathbf{v}_t \in \mathbb{R}^n$ is the Gaussian *driving noise process*, with zero mean and covariance \mathbf{Q}
- $\mathbf{w}_t \in \mathbb{R}^m$ is the *observation noise*, with zero mean and covariance \mathbf{R}

Covariance matrix \mathbf{R} is assumed to be diagonal. Its diagonal elements are denoted by r_i , $i = 1 \dots m$. The generative model corresponding to LDS (5.3) is

$$\begin{aligned} \Pr(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \frac{1}{(2\pi)^{\frac{n}{2}} \det(\mathbf{Q})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})^\top \mathbf{Q}^{-1}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})\right) \\ \Pr(\mathbf{y}_t | \mathbf{x}_t) &= \frac{1}{(2\pi)^{\frac{m}{2}} \det(\mathbf{R})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t)^\top \mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t)\right). \end{aligned}$$

⁵For the multivariate Gaussian, we use notation G_σ when it has zero mean and diagonal covariance $\sigma^2 \mathbf{I}$

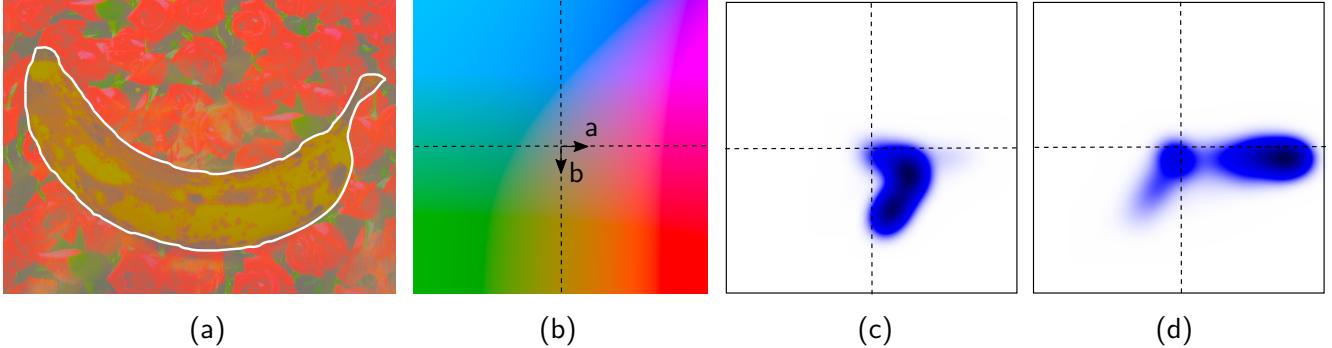


Figure 5.1: (a) The L-equalized banana. (b) Correspondence between RGB and an (a,b)-plane of Lab color space (at constant L=0.5). (c) Color density \tilde{p}^1 of the banana, in ab color space (L is ignored) (f) Color density \tilde{p}^0 of the background in ab color space (L is ignored).

[Zhong and Sclaroff, 2003, Chan and Vasconcelos, 2008] build upon this dynamic texture model to perform background subtraction. During a learning phase, where no object is supposed to be present, parameters \mathbf{A} and \mathbf{C} are learnt using a maximum likelihood criterion. At test time, one searches for the best estimate \mathbf{x}_t with a Maximum A Posteriori criterion,

$$\begin{aligned}\hat{\mathbf{x}}_t &= \underset{\mathbf{x}_t}{\operatorname{argmax}} \Pr(\mathbf{y}_t | \mathbf{x}_t) \Pr(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ &= \underset{\mathbf{x}_t}{\operatorname{argmin}} -\log \Pr(\mathbf{y}_t | \mathbf{x}_t) - \log \Pr(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ &= \underset{\mathbf{x}_t}{\operatorname{argmin}} (\mathbf{y}_t - \mathbf{Cx}_t)^T \mathbf{R}^{-1} (\mathbf{y}_t - \mathbf{Cx}_t) + (\mathbf{x}_t - \mathbf{Ax}_{t-1})^T \mathbf{Q}^{-1} (\mathbf{x}_t - \mathbf{Ax}_{t-1})\end{aligned}\quad (5.4)$$

The first term in the right-hand side of (5.4) is the error between the actual observation and the estimated one. Since \mathbf{R} is diagonal, this term can be rewritten

$$\sum_{i=1}^m \left(\frac{\mathbf{y}_i - [\mathbf{Cx}_t]_i}{\sqrt{r_i}} \right)^2,$$

where $[\mathbf{Cx}_t]_i$ is the i^{th} element of vector \mathbf{Cx}_t . Once \mathbf{x}_t has been estimated by solving Eq. (5.4), the foreground mask is obtained by thresholding this error⁶, for each position i . Since there is a straightforward one-to-one correspondence between position x and index i , $\phi_t(x)$ is set to one if the error at i exceeds the threshold.

It is interesting to note that, conversely to color density-based background models, this texture-based model is not pixelwise, in the sense that a unique linear system (5.3) - and the subsequent derived generative model - represents the evolution of the entire image.

5.2 Non-parametric color density-based VOS

Before digging into our first contribution, we give a detailed study of color density-based VOS methods that were mentioned in Section 5.1.1. The context is semi-supervised VOS, in the sense that the initial labeling ϕ_1 is known. All objects are present in the first frame,

$$\Omega_1^i = \{\mathbf{x} | \phi_1(\mathbf{x}) = i\} \neq \emptyset, \forall i \in \{0, \dots, n\},$$

and no detection of new object is performed.

⁶The L^2 error is actually replaced by a robust error norm, but we chose to keep this detail apart from our short survey

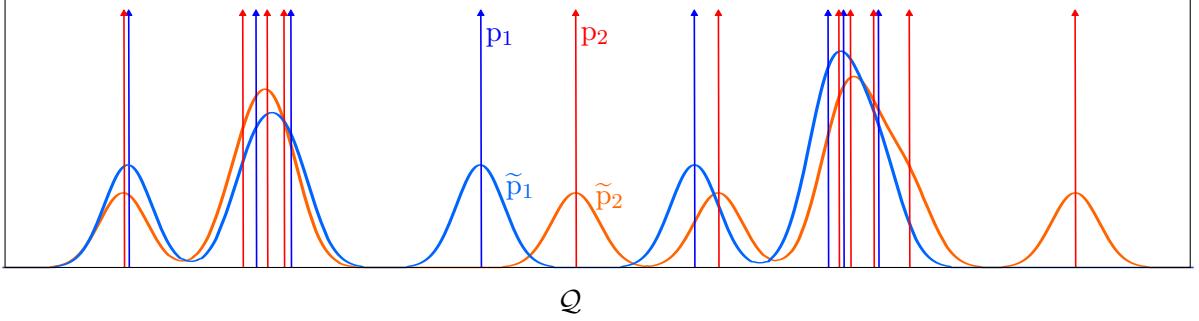


Figure 5.2: Two empirical densities (Dirac mixtures), p_1 and p_2 and their corresponding smooth kernel density estimates \tilde{p}_1 and \tilde{p}_2

5.2.1 Information-theoretic measures

Further derivations require to define basic concepts of information theory. Let $p : \mathcal{Q} \rightarrow \mathbb{R}^+$ be a probability density function (PDF) over \mathcal{Q} , thus verifying

$$\int_{\mathcal{Q}} p(\mathbf{q}) d\mathbf{q} = 1.$$

The entropy H of PDF p , which measures the level of "disorder" or "uncertainty" of p , is

$$H(p) = \mathbb{E}_p[-\log p] = - \int_{\mathcal{Q}} p(\mathbf{q}) \log p(\mathbf{q}) d\mathbf{q}, \quad (5.5)$$

where \mathbb{E}_p is the expectation with respect to density p . The cross entropy CE of a PDF p_2 relative to a PDF p_1 is

$$CE(p_1, p_2) = \mathbb{E}_{p_1}[-\log p_2] = - \int_{\mathcal{Q}} p_1(\mathbf{q}) \log p_2(\mathbf{q}) d\mathbf{q}.$$

It measures in which extent p_2 deviates from p_1 . We denote by D_{KL} the Kullback-Liebler (KL) divergence, also known as relative entropy, of PDF p_2 relative to PDF p_1 ,

$$D_{KL}(p_1 \| p_2) = \mathbb{E}_{p_1} \left[\log \left(\frac{p_1}{p_2} \right) \right] = \int_{\mathcal{Q}} p_1(\mathbf{q}) \log \left(\frac{p_1(\mathbf{q})}{p_2(\mathbf{q})} \right) d\mathbf{q}.$$

The entropy, cross entropy and KL divergence are linked as follows,

$$D_{KL}(p_1 \| p_2) = CE(p_1, p_2) - H(p_1). \quad (5.6)$$

5.2.2 Kernel density estimates and Dirac mixtures

In the density-based VOS methods that we review in this section, the underlying assumption is that the probability of observing $f_t(\mathbf{x})$ only depends on $\phi_t(\mathbf{x})$. It does not depend on \mathbf{x} itself, in the sense that a given color will have the same probability of appearance in a given object, regardless of the location considered within this object. The conditional probability $\Pr(f_t(\mathbf{x}) = \mathbf{q} | \phi_t(\mathbf{x}) = i)$ is linked to a PDF, that can be estimated.

The sample distribution of colors within a region Ω_t^i is a discrete distribution, as each pixel within Ω_t^i contributes to one isolated color $\mathbf{q} \in \mathcal{Q}$ in the distribution. The link between discrete and continuous densities can be made using the Dirac distribution. The empirical density of image f_t on Ω_t^i , which represents the sample distribution, is formulated using a *Dirac mixture*,

$$p_t^i(\mathbf{q}) = \frac{1}{|\Omega_t^i|} \int_{\Omega_t^i} \delta(\mathbf{q} - f_t(\mathbf{x})) d\mathbf{x}, \quad (5.7)$$

where δ is the Dirac distribution defined over \mathbb{R}^3 . Note that p_t^i is defined iff object Ω_t^i is non-empty. Rigorously, p_t^i is not a PDF, as δ is a distribution rather than a function. Instead, p_t^i can be referred to as a *generalized PDF*. It still verifies

$$\int_{\mathcal{Q}} p_t^i(\mathbf{q}) d\mathbf{q} = 1.$$

Density-based VOS will involve to calculate dissimilarities between densities. Measuring the dissimilarity between two Dirac mixtures using pointwise divergences, such as the Bhattacharyya distance or the KL divergence, does not make sense. Let p_1 and p_2 be two Dirac mixtures. Suppose that p_1 is a slight offset of p_2 , that is

$$p_2(\mathbf{q}) = p_1(\mathbf{q} + \boldsymbol{\epsilon}), \quad \forall \mathbf{q} \in \mathcal{Q},$$

for an infinitesimal offset $\boldsymbol{\epsilon}$. Despite the fact that the two densities are very similar, their supports do not overlap, which would make their KL divergence go to infinity, or their Bhattacharyya coefficient be 0. A more general case is illustrated in Fig. 5.2, where the match between p_1 and p_2 cannot be measured if we simply look at the "overlap" between their Dirac impulses.

It is more relevant to consider smooth approximations of PDFs, which can be obtained by kernel-density estimation. The kernel density estimate of f_t over region Ω_t^i is defined as

$$\tilde{p}_{t,\sigma}^i(\mathbf{q}) = \frac{1}{|\Omega_t^i|} \int_{\Omega_t^i} G_\sigma(\mathbf{q} - f_t(\mathbf{x})) d\mathbf{x}, \quad (5.8)$$

where standard deviation σ controls the spatial extent of the approximation. The estimation \tilde{p}_t^i is a smoothed version of p_t^i ,

$$\tilde{p}_{t,\sigma}^i(\mathbf{q}) = (p_t^i * G_\sigma)(\mathbf{q}),$$

and p_t^i can be intuitively viewed as a pointwise limit of $\tilde{p}_{t,\sigma}^i$ as σ approaches 0,

$$\lim_{\sigma \rightarrow 0} \tilde{p}_{t,\sigma}^i = p_t^i.$$

For conciseness of notation, from now on, we omit index σ on kernel-based densities. Calculating pointwise dissimilarities between kernel densities will be relevant, unlike calculating them on Dirac mixtures. Indeed, there is a strong overlap between \tilde{p}_1 and \tilde{p}_2 if their corresponding Dirac mixtures p_1 and p_2 have neighboring Dirac impulses, as depicted in Fig. 5.2. An example of color kernel density estimates of real object and background, on a projection of Lab color space, is illustrated in Fig. 5.1.

When taken at time $t+1$, color densities are unknown, as they are functions of ϕ_{t+1} . We make the dependence on the unknown ϕ_{t+1} explicit, by using the functional notation $\tilde{p}_{t+1}^i[\phi_{t+1}]$. For a given color \mathbf{q} , we have

$$\tilde{p}_{t+1}^i[\phi_{t+1}; \mathbf{q}] = \frac{1}{|\Omega_{t+1}^i|} \int_{\Omega_{t+1}^i} G_\sigma(\mathbf{q} - f_{t+1}(\mathbf{x})) d\mathbf{x} = \frac{\int_{\{\phi_{t+1}=i\}} G_\sigma(\mathbf{q} - f_{t+1}(\mathbf{x})) d\mathbf{x}}{|\{\phi_{t+1}=i\}|}, \quad (5.9)$$

where $\{\phi_{t+1} = i\}$ is a short-hand notation for $\{\mathbf{x} | \phi_{t+1}(\mathbf{x}) = i\}$. The Dirac mixture $p_{t+1}^i[\phi_{t+1}]$ can be developed similarly.

Further mathematical derivations will involve measures mixing Dirac mixtures p_t^i and kernel-density estimates \tilde{p}_t^i . We introduce the *smooth Kullback-Leibler divergence* $\widetilde{D}_{\text{KL}}$, as a functional of a Dirac mixture p_1 and a smooth density \tilde{p}_2 :

$$\widetilde{D}_{\text{KL}}(p_1 \| \tilde{p}_2) = \mathbb{E}_{p_1} \left[\log \left(\frac{\tilde{p}_1}{\tilde{p}_2} \right) \right] = - \int_{\mathcal{Q}} p_1(\mathbf{q}) \log \left(\frac{\tilde{p}_1(\mathbf{q})}{\tilde{p}_2(\mathbf{q})} \right) d\mathbf{q}. \quad (5.10)$$

Note that it is different from the KL divergence of \tilde{p}_2 relative to \tilde{p}_1 . Moreover, $\widetilde{D}_{\text{KL}}$ does not need to have \tilde{p}_1 as a parameter, as \tilde{p}_1 can be deduced from p_1 . The smooth KL divergence and cross entropy are linked by

$$\widetilde{D}_{\text{KL}}(p_1 \| \tilde{p}_2) = \text{CE}(p_1, \tilde{p}_2) - \text{CE}(p_1, \tilde{p}_1). \quad (5.11)$$

5.2.3 Matching densities

The optimal labeling minimizes an energy containing a data term, which is a functional of color densities of all regions at time t and $t+1$,

$$\phi_{t+1}^* = \operatorname{argmin}_{\phi_{t+1}} E_{\text{data}}[(\tilde{p}_t^i)_{i=0 \dots n}, (\tilde{p}_{t+1}^i[\phi_{t+1}])_{i=0 \dots n}] + E_{\text{reg}}[\phi_{t+1}].$$

Let us set aside the regularization term $E_{\text{reg}}[\phi_{t+1}]$ and focus on the data-fitting term E_{data} . We review the data terms of [Freedman and Zhang, 2004], [Zhang and Freedman, 2005] and [Chiverton et al., 2008]. These three methods make single-object VOS, so the labeling is restricted to

$$\phi_t : \mathcal{D} \longrightarrow \{0, 1\}.$$

In [Freedman and Zhang, 2004], it was proposed to enforce matching between the densities of object Ω_{t+1}^1 and Ω_t^1 , the background being ignored. Matching is done either by maximizing the Bhattacharyya coefficient,

$$E_{\text{data}}^{\text{Freedman1}}[\tilde{p}_t^1, p_{t+1}^1[\phi_{t+1}]] = - \int_{\mathcal{Q}} \sqrt{\tilde{p}_t^1(\mathbf{q}) p_{t+1}^1[\phi_{t+1}; \mathbf{q}]} d\mathbf{q},$$

or minimizing the KL divergence from $p_{t+1}^1[\phi_{t+1}]$ to \tilde{p}_t^1 ,

$$E_{\text{data}}^{\text{Freedman2}}[\tilde{p}_t^1, p_{t+1}^1[\phi_{t+1}]] = D_{\text{KL}}(\tilde{p}_t^1 \| p_{t+1}^1[\phi_{t+1}]) = \int_{\mathcal{Q}} \tilde{p}_t^1(\mathbf{q}) \log \left(\frac{\tilde{p}_t^1(\mathbf{q})}{p_{t+1}^1[\phi_{t+1}; \mathbf{q}]} \right) d\mathbf{q}.$$

This approach was extended by the same authors in [Zhang and Freedman, 2005], to avoid the drifting effect of considering only the color density of the object and ignoring the background. Experiments in [Zhang and Freedman, 2005] highlighted this effect. The mismatch between density of background Ω_{t+1}^0 and object Ω_t^1 was added. Again, both Bhattacharyya coefficient and KL divergence are tested. With the Bhattacharyya coefficient, it comes

$$E_{\text{data}}^{\text{Zhang}}[(\tilde{p}_t^1), (p_{t+1}^i[\phi_{t+1}])_{i=0 \dots 1}] = - \int_{\mathcal{Q}} \sqrt{\tilde{p}_t^1(\mathbf{q}) p_{t+1}^1[\phi_{t+1}; \mathbf{q}]} d\mathbf{q} + \int_{\mathcal{Q}} \sqrt{\tilde{p}_t^1(\mathbf{q}) p_{t+1}^0[\phi_{t+1}; \mathbf{q}]} d\mathbf{q}.$$

In [Chiverton et al., 2008], the energy is not explicitly formulated as a match or mismatch between densities, in the sense that its initial formulation does not involve dissimilarity measures between densities as integrals over \mathcal{Q} . It integrates log-probability ratios over object and background at time $t+1$,

$$\begin{aligned} E_{\text{data}}^{\text{Chiverton}}[\tilde{p}_t^1, (\tilde{p}_{t+1}^i[\phi_{t+1}])_{i=0 \dots 1}] &= - \int_{\{\phi_{t+1}=0\}} \log \left(\frac{\tilde{p}_{t+1}^0[\phi_{t+1}; \mathbf{f}_{t+1}(\mathbf{x})]}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x} \\ &\quad + \int_{\{\phi_{t+1}=1\}} \log \left(\frac{\tilde{p}_{t+1}^1[\phi_{t+1}; \mathbf{f}_{t+1}(\mathbf{x})]}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x}, \end{aligned}$$

which we prove to be equivalent to maximizing the smooth KL divergence from \tilde{p}_t^1 to $p_{t+1}^0[\phi_{t+1}]$, while minimizing the one from \tilde{p}_t^1 to $p_{t+1}^1[\phi_{t+1}]$. The smooth KL divergences are multiplied by areas,

$$E_{\text{data}}^{\text{Chiverton}}[\tilde{p}_t^1, (p_{t+1}^i[\phi_{t+1}])_{i=0 \dots 1}] = - |\Omega_{t+1}^0| \widetilde{D_{\text{KL}}}(p_{t+1}^0[\phi_{t+1}] \| \tilde{p}_t^1) + |\Omega_{t+1}^1| \widetilde{D_{\text{KL}}}(p_{t+1}^1[\phi_{t+1}] \| \tilde{p}_t^1).$$

We develop our proof of this result in appendix B.1.2. Fig. 5.3 graphically summarizes how color densities in object and background at time t and $t+1$ are related in the energy terms of these three citations.

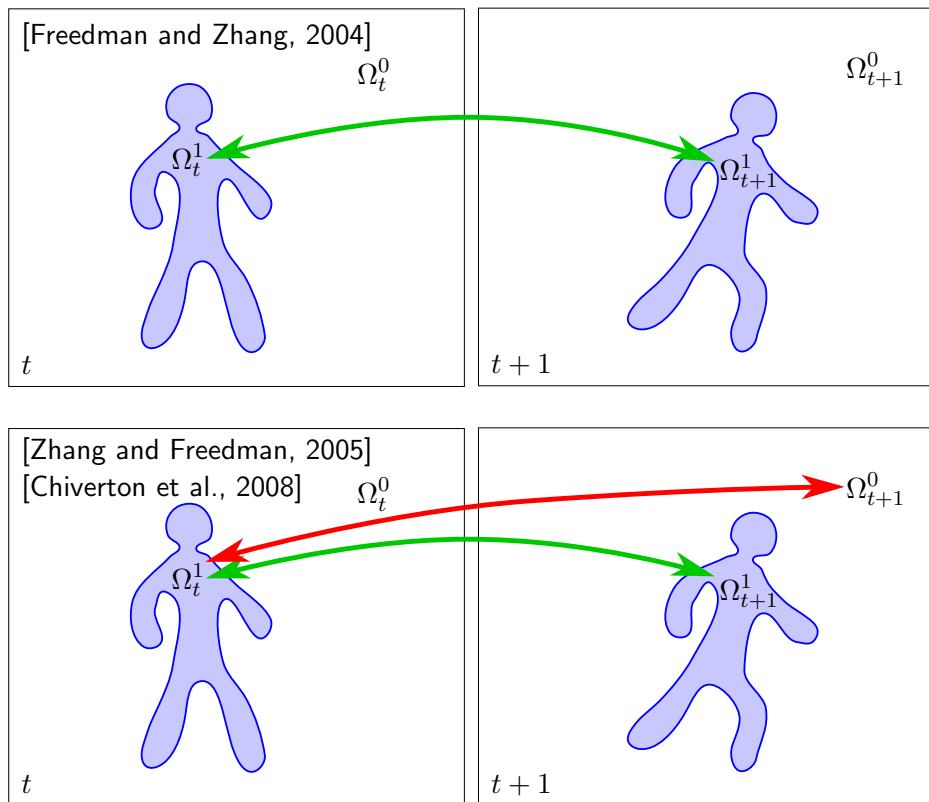


Figure 5.3: How are color densities in object and background linked? Desired **similarity** and **dissimilarity** between color densities are shown with **green** and **red** arrows, respectively

5.2.4 Limitations

The three reviewed methods are designed for single-object video segmentation. At the time we investigated VOS (circa 2010-2011), we were unaware of existing multi-object color density-based video segmentation methods. The extension of [Zhang and Freedman, 2005] and [Chiverton et al., 2008] to multiple video object segmentation is not natural. Should the mismatch term between the background at time $t + 1$ and object at time t be extended to all objects at time t ? If so, what is the underlying probabilistic model?

In order to set up a multi-object video segmentation with clear underlying probabilistic foundations, the Bayesian interpretation of segmentation can be considered.

5.3 Segmentation as a Bayesian estimation problem

5.3.1 General framework for a single image

Let us leave time for a while and go back to the simpler problem of the segmentation of a static image \mathbf{f} . We seek for an ordered partition P of the image domain \mathcal{D} into $n + 1$ disjoint regions,

$$P = (\Omega_i)_{i=0 \dots n}.$$

Again, as in Section 5.1, we choose the convention such that Ω_0 is the background and Ω_i is the i^{th} object. Segmentation is stated as the search of the partition maximizing the posterior probability, commonly referred to as *Maximum a posteriori* (MAP) estimation,

$$P^* = \operatorname{argmax}_{P \in \mathcal{P}} \Pr(P|\mathbf{f}),$$

where \mathcal{P} is the set of possible partitions. If we look at segmentation under the Bayesian viewpoint, the posterior probability of P given \mathbf{f} can be expanded to

$$\Pr(P|\mathbf{f}) = \frac{\Pr(\mathbf{f}|P)\Pr(P)}{\Pr(\mathbf{f})}, \quad (5.12)$$

where

- $\Pr(\mathbf{f}|P)$ is the likelihood of \mathbf{f} given P
- $\Pr(P)$ is the prior probability of P , i.e. before observing \mathbf{f} ,
- $\Pr(\mathbf{f})$ is the prior probability of \mathbf{f} .

Note that looking for the MAP estimate is not a fully-Bayesian treatment of segmentation, as discussed in [Bishop, 2006, Section 1.2.6]. In a fully-Bayesian framework, we would seek to calculate the predictive distribution $\Pr(P|\mathbf{f})$ itself, thanks to exact inference (which is possible only under very restricted conditions, e.g. Gaussian prior) or approximate inference, e.g. variational inference [Bishop, 2006, Chapter 10].

It is common to carry on the mathematical derivation with the partition, as in [Mansouri et al., 2006, Mitiche and Ben Ayed, 2011, Paragios and Deriche, 2002, Paragios and Deriche, 2005]. However, since we deal with a labeling function in our contributions, it is more convenient to introduce it early in the progression. Consider the labeling function $\phi : \mathcal{D} \rightarrow \{0, \dots, n\}$. There is a one-to-one correspondence between a labeling and its resulting partition. Without loss of generality, we switch to

$$\phi^* = \operatorname{argmax}_{\phi} \Pr(\phi|\mathbf{f}) = \operatorname{argmax}_{\phi} \frac{\Pr(\mathbf{f}|\phi)\Pr(\phi)}{\Pr(\mathbf{f})}.$$

Since $\Pr(\mathbf{f})$ does not depend on ϕ ,

$$\phi^* = \operatorname{argmax}_{\phi} \Pr(\mathbf{f}|\phi) \Pr(\phi).$$

The problem is cast into a minimization problem by taking the negative logarithms,

$$\phi^* = \operatorname{argmin}_{\phi} -\log \Pr(\mathbf{f}|\phi) - \log \Pr(\phi), \quad (5.13)$$

where the first right-hand term is the data-fitting term and the second right-hand term is the regularization term. We make several assumptions to simplify the corresponding likelihood and prior.

As regards likelihood $\Pr(\mathbf{f}|\phi)$, the first simplifying assumption to get a tractable expression is conditional independence between $\mathbf{f}(\mathbf{x})$ and $\mathbf{f}(\mathbf{y})$ given ϕ , for any pair of distinct points $\mathbf{x} \neq \mathbf{y}$. This leads to

$$\Pr(\mathbf{f}|\phi) = \prod_{\mathcal{D}} \Pr(\mathbf{f}(\mathbf{x})|\phi) d\mathbf{x}.$$

Additional assumption is that the likelihood of having an image value at given point \mathbf{x} only depends on the labeling at this point, regardless of location \mathbf{x} ,

$$\Pr(\mathbf{f}(\mathbf{x})|\phi) = \Pr(\mathbf{f}(\mathbf{x})|\phi(\mathbf{x})).$$

Following [Zhu and Yuille, 1996] and Eq. (5.8), conditional probability $\Pr(\mathbf{f}(\mathbf{x}) = \mathbf{q}|\phi(\mathbf{x}) = i)$ is related to the density value $\tilde{p}_i(\mathbf{q})$. This color density depends on ϕ . We make this dependence explicit by using functional notation $\tilde{p}_i[\phi; \mathbf{q}]$,

$$\begin{aligned} \Pr(\mathbf{f}|\phi) &= \prod_{\mathcal{D}} \Pr(\mathbf{f}(\mathbf{x})|\phi(\mathbf{x})) d\mathbf{x} \\ &= \prod_{\mathcal{D}} \tilde{p}_{\phi(\mathbf{x})}[\phi; \mathbf{f}(\mathbf{x})] d\mathbf{x} \\ &= \prod_{i=0}^n \prod_{\{\phi=i\}} \tilde{p}_i[\phi; \mathbf{f}(\mathbf{x})] d\mathbf{x}. \end{aligned} \quad (5.14)$$

Taking the negative logarithm, we obtain

$$-\log \Pr(\mathbf{f}|\phi) = -\sum_{i=0}^n \int_{\{\phi=i\}} \log \tilde{p}_i[\phi; \mathbf{f}(\mathbf{x})] d\mathbf{x}, \quad (5.15)$$

which is the "multilabel equivalent" of the entropy data term $E_{\text{data}}^{\text{entropy}}$ defined in Eq. (4.20), formulated as a functional of a labeling, instead of a functional of a curve.

As regards the prior term, for regularization, it is common to take the total length of boundaries of regions [Zhu and Yuille, 1996],

$$-\log \Pr(\phi) = \int_{\Gamma(\phi)} ds, \quad (5.16)$$

where s is an arc-length parameter and $\Gamma(\phi)$ is the set of boundaries of ϕ ,

$$\Gamma(\phi) = \bigcup_{i=0}^n \partial \Omega_i = \bigcup_{i=0}^n \partial \{\phi = i\}.$$

Transformation (5.16) is a mathematical convenience for segmentation, rather than a pure probabilistic interpretation. Putting back (5.15) and (5.16) into (5.13), the final form of the single-image segmentation problem is

$$\phi^* = \operatorname{argmin}_{\phi} -\sum_{i=0}^n \int_{\{\phi=i\}} \log \tilde{p}_i[\phi; \mathbf{f}(\mathbf{x})] d\mathbf{x} + \int_{\Gamma(\phi)} ds.$$

5.3.2 Temporal extension

Let us now reintegrate the segmentation framework in time. The derivation developed in this section is a extended version of the original one appearing in [BMVC11, Section 2.1]. The general model in Eq. (5.2) is restricted to a first-order scheme in time, on a *frame-to-frame* basis. At time t , we assume that the next labeling can be determined using the current labeling, the current and next image only,

$$\Pr(\phi_{t+1}|\mathbf{f}_{1:t+1}, \phi_{1:t}) = \Pr(\phi_{t+1}|\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t).$$

Using Bayes theorem, we obtain a temporal extension of Eq. (5.12),

$$\begin{aligned}\Pr(\phi_{t+1}|\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t) &= \frac{\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\mathbf{f}_t, \phi_t, \phi_{t+1})}{\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)} \\ &= \frac{\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)\Pr(\mathbf{f}_t, \phi_t)}{\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)}.\end{aligned}$$

Such factorization can be found e.g. in [Mansouri, 2002]. In the maximization with respect to ϕ_{t+1} , terms independent of ϕ_{t+1} can be dropped,

$$\begin{aligned}\phi_{t+1}^* &= \operatorname{argmax}_{\phi_{t+1}} \Pr(\phi_{t+1}|\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t) \\ &= \operatorname{argmax}_{\phi_{t+1}} \frac{\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)\Pr(\mathbf{f}_t, \phi_t)}{\Pr(\mathbf{f}_t, \mathbf{f}_{t+1}, \phi_t)} \\ &= \operatorname{argmax}_{\phi_{t+1}} \Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t).\end{aligned}$$

Maximization of the posterior probability is turned into minimization problem

$$\begin{aligned}\phi_{t+1}^* &= \operatorname{argmin}_{\phi_{t+1}} E_{\text{data}}[\phi_{t+1}] + E_{\text{reg}}[\phi_{t+1}] \\ &= \operatorname{argmin}_{\phi_{t+1}} -\log \Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) - \log \Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t).\end{aligned}\tag{5.17}$$

As in the static image model in Section 5.3.1, we now make several simplifying transformations on the two terms, namely the likelihood $\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1})$ and prior $\Pr(\phi_{t+1}|\mathbf{f}_t, \phi_t)$.

Regarding the likelihood term, to obtain a tractable expression, the assumption of conditional independence between $\mathbf{f}_{t+1}(\mathbf{x})$ and $\mathbf{f}_{t+1}(\mathbf{y})$, for any pair of distinct points $\mathbf{x} \neq \mathbf{y}$, given the labelings at times t and $t+1$, gives

$$\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) = \prod_{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}) d\mathbf{x}.\tag{5.18}$$

Probability $\Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1})$ is the likelihood of observing a particular color at space-time location $(\mathbf{x}, t+1)$ given current image and both current and next labelings. A further simplification is obtained by making the reasonable assumption that the likelihood of observing $\mathbf{f}_{t+1}(\mathbf{x})$ depends on \mathbf{f}_t , ϕ_t and next labeling at \mathbf{x} only,

$$\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) = \prod_{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x})) d\mathbf{x}.\tag{5.19}$$

Until now, the mathematical derivation has just led to a multilabel extension of [Mansouri, 2002, Section 2]. It does not involve novel models nor assumptions. These ones appear on stage in the contribution detailed in the next section.

5.4 Short-sight modeling of background

The development of a novel modeling of the likelihood (5.19), and the algorithm to minimize the energy derived from this likelihood, are the core of our contribution on supervised VOS, which we describe in the current section. We start by linking the general likelihood (5.19) to color density-based tracking, and relate it to existing approaches. Doing so, we obtain a specific calculable form for this likelihood, which can be minimized. Then, by highlighting the shortcomings of this form, we develop our short-sight model. The current section and related appendices contain mathematical proofs, which are additional material provided by this manuscript. They did not appear in the original conference paper [BMVC11].

5.4.1 Matching densities through time

The MAP estimation framework described in Section 5.3.2 is a generic model, in the sense it does not provide a particular form for the likelihood (5.19). This term embodies our assumptions about color constancy over time, we link the likelihood $\Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x}))$ to color density at time t , regardless of position \mathbf{x} .

Consider kernel densities at time t and $t + 1$, written in Eqs (5.8) and (5.9), respectively. We relate probability $\Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x}))$ to density $\tilde{p}_t^{\phi_{t+1}(\mathbf{x})}$, instead of the unknown $\tilde{p}_{t+1}^{\phi_{t+1}(\mathbf{x})}$. Thus, the likelihood term (5.19) becomes

$$\begin{aligned}\Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) &= \prod_{\mathcal{D}} \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x})) \\ &= \prod_{\mathcal{D}} \tilde{p}_t^{\phi_{t+1}(\mathbf{x})}(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} \\ &= \prod_{i=0}^n \prod_{\{\phi_{t+1}=i\}} \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x}\end{aligned}$$

The data term is the negative logarithm of the likelihood,

$$\begin{aligned}E_{\text{data}}[\phi_{t+1}] &= -\log \Pr(\mathbf{f}_{t+1}|\mathbf{f}_t, \phi_t, \phi_{t+1}) \\ &= -\sum_{i=0}^n \int_{\{\phi_{t+1}=i\}} \log(\tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x}))) d\mathbf{x}. \tag{5.20}\end{aligned}$$

Pixels at time $t + 1$ will tend to be included into the best matching region, regarding statistics at time t . To some extent, E_{data} is a "time-consistent" multilabel counterpart of the entropy-based data term (4.20), of [Kim et al., 2005]. Advantageously, unlike in the static image model (5.14), \tilde{p}_t^i does not depend itself on the unknown labeling ϕ_{t+1} . Implicitly, our data term matches \tilde{p}_t^i and \tilde{p}_{t+1}^i , for each region i . The matching is illustrated in Fig. 5.4. It can be proven that it is equivalent to minimize the weighted sum of cross entropies

$$E_{\text{data}}[\phi_{t+1}] = \sum_{i=0}^n |\{\phi_{t+1} = i\}| \text{CE}(p_{t+1}^i[\phi_{t+1}], \tilde{p}_t^i).$$

The proof is given in appendix B.1.3. The interpretation of E_{data} can be pushed further. Using the relationship between the smooth KL divergence and cross entropy in Eq. (5.11), we obtain

$$\text{CE}(p_{t+1}^i, \tilde{p}_t^i) = \widetilde{D_{\text{KL}}}(p_{t+1}^i \| \tilde{p}_t^i) + \text{CE}(p_{t+1}^i, \tilde{p}_{t+1}^i).$$

Hence, minimizing E_{data} has two effects. On the one hand, it pushes p_{t+1}^i to match \tilde{p}_t^i , by means of the smooth KL term. On the other hand, thanks to the cross entropy term between p_{t+1}^i and its smoothed version \tilde{p}_{t+1}^i , it forces p_{t+1}^i to have a compact support, *i.e.* to have a low scattering.

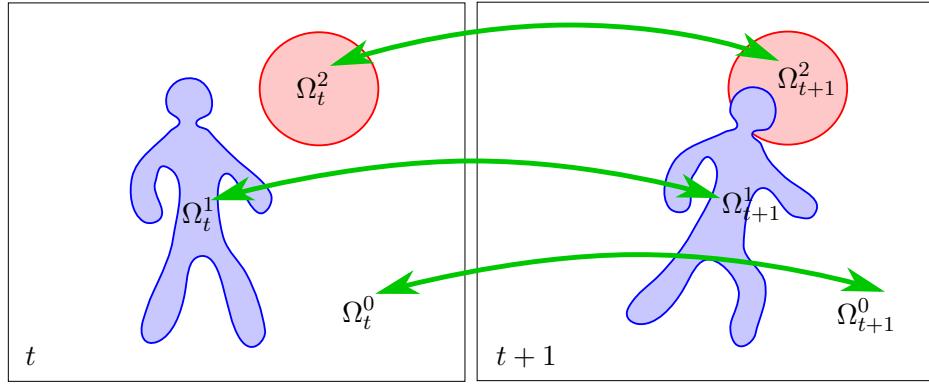


Figure 5.4: Color densities are matched through time, for each object and the background

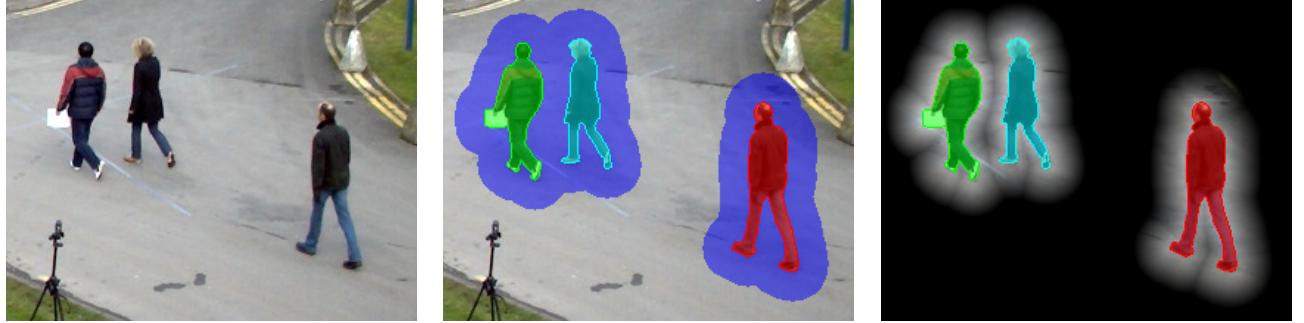


Figure 5.5: Segmented objects endowed with their own short-sight perceptions of background. (Left) Original image, (center) segmented targets with surrounding bands, and (right) background faded to black with respect to its contribution to perceptions.

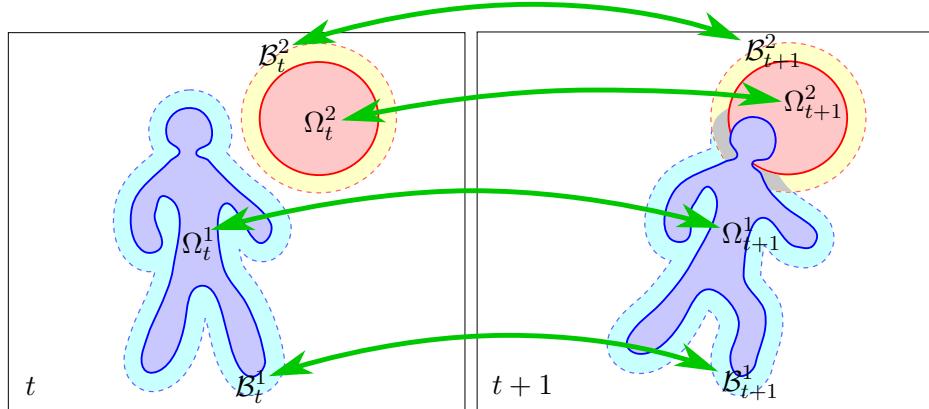


Figure 5.6: In the short-sight model, color densities are matched through time, for each object and its surrounding band

5.4.2 Short-sight perception of background

Exploiting the color density in the whole background raises a major shortcoming. The matter here is the scattering of color statistics, which can happen in cluttered backgrounds, due to the presence of various objects and static parts with different appearances. Real videos do not contain static backgrounds. Any untracked object appearing in the scene gets part of the background. These undesirable objects modify the color distribution of background, even if they are far from the objects of interest Ω_t^i .

To illustrate the undesirable effect it may have on segmentation, let us reconsider the minimization of E_{data} under the perspective of curve evolution, in a single-object setting. Suppose that the boundary between background Ω_{t+1}^0 and object Ω_{t+1}^1 is described by a parameterized curve \mathbf{c}_{t+1} , such that

$$\begin{aligned}\Omega_{t+1}^0 &= \text{out}(\mathbf{c}_{t+1}) \\ \Omega_{t+1}^1 &= \text{in}(\mathbf{c}_{t+1}).\end{aligned}$$

Index $t + 1$ can be dropped for notational simplicity. Curve \mathbf{c} and image \mathbf{f} are considered at time $t + 1$. The data term expressed as a functional of \mathbf{c} is

$$E_{\text{data}}[\mathbf{c}] = - \int_{\text{in}(\mathbf{c})} \log \tilde{p}_t^1(\mathbf{f}(\mathbf{x})) \, d\mathbf{x} - \int_{\text{out}(\mathbf{c})} \log \tilde{p}_t^0(\mathbf{f}(\mathbf{x})) \, d\mathbf{x}.$$

Calculus of variations with respect to \mathbf{c} , that was addressed in Section 4.3, leads to the functional derivative

$$\frac{\delta E_{\text{data}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = [\log \tilde{p}_t^1(\mathbf{f}(\mathbf{c}(u))) - \log \tilde{p}_t^0(\mathbf{f}(\mathbf{c}(u)))] \mathbf{n}(u).$$

Making \mathbf{c} dependent on "algorithmic" time τ - different from frame index t -, the evolution equation minimizing $E_{\text{data}}[\mathbf{c}]$ is

$$\frac{\partial \mathbf{c}(u, \tau)}{\partial \tau} = - \frac{\delta E_{\text{data}}[\mathbf{c}]}{\delta \mathbf{c}(u)} = [\log \tilde{p}_t^0(\mathbf{f}(\mathbf{c}(u, \tau))) - \log \tilde{p}_t^1(\mathbf{f}(\mathbf{c}(u, \tau)))] \mathbf{n}(u),$$

Regardless of curve implementation, which may rely either on parametric contours or level-sets, the curve will expand locally, at point $\mathbf{c}(u, \tau)$, if color $\mathbf{f}(\mathbf{c}(u, \tau))$ is more present in density \tilde{p}_t^1 than in \tilde{p}_t^0 , and shrink in the opposite case. Estimating these densities over entire regions can be a drawback for VOS in real scenes, especially for the background region distribution, which may be cluttered and contain many non-tracked objects. In such case, statistical color data is highly scattered, so that background color density \tilde{p}_t^0 may not be confident. If \tilde{p}_t^0 gets small for nearly all colors, regions are more likely to include background pixels and leak outside actual objects.

To overcome this limitation and obtain reliable background image data, we head towards a background model based on "short-sight perception". To some extent, we adapt the philosophy of local region-based active contours [Lankton and Tannenbaum, 2008, Brox and Cremers, 2009] to the tracking problem. Our approach is also related to the idea of spatial context brought up in [Nguyen et al., 2007]. Instead of considering statistical knowledge in the same extent for all background points, we give more importance to background points as they are closer to objects, and modify E_{data} accordingly. To this purpose, let d_t^i be the Euclidean distance from any background point \mathbf{x} to the nearest point in a given object Ω_t^i . For each $i = 1 \dots n$, let

$$\begin{aligned}d_t^i &: \Omega_t^0 \longrightarrow \mathbb{R}^+ \\ d_t^i(\mathbf{x}) &= \min_{\mathbf{y} \in \Omega_t^i} \|\mathbf{x} - \mathbf{y}\|.\end{aligned}$$

Since the nearest point will always be a boundary point, d_t^i can be equivalently defined as

$$d_t^i(\mathbf{x}) = \min_{\mathbf{y} \in \partial \Omega_t^i} \|\mathbf{x} - \mathbf{y}\|.$$

Let $\mathcal{B}_t^i \subset \Omega_t^0$ be the set of background points which distance to Ω_t^i is lower or equal to a fixed threshold B . For each $i = 1 \dots n$,

$$\mathcal{B}_t^i = \{\mathbf{x} \in \Omega_t^0 \mid d_t^i(\mathbf{x}) \leq B\}.$$

Domain \mathcal{B}_t^i can be thought of a band surrounding Ω_t^i , of width at most B .

As we wish to decrease the contribution of background points as they get farther from objects, we introduce a positive real-valued weighting function $\psi : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, which should be continuous, compactly supported and non-increasing with respect to distance d . We choose piecewise-linear function

$$\psi(d) = \begin{cases} 1 - \frac{d}{B} & \text{if } d \leq B \\ 0 & \text{otherwise.} \end{cases}$$

We consider that background color statistics are relevant only within bands \mathcal{B}_t^i , and thus ignore available knowledge about color appearance in the "far" background $\Omega_t^0 \setminus \bigcup_{i=1}^n \mathcal{B}_t^i$. Points that are at a distance greater than B , to all objects, will not contribute to kernel density estimations nor to the data term.

Since a band \mathcal{B}_t^i is defined for every Ω_t^i , each object has its own local perception of surrounding background and ignores the far background. This principle is depicted in Fig. 5.5. In the right part of the image, background pixels are faded to black proportionally with their distance to the nearest object boundary. The Dirac mixture representing the empirical density of \mathbf{f}_t in band \mathcal{B}_t^i is

$$b_t^i(\mathbf{q}) = \frac{1}{\int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) d\mathbf{x}} \int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) \delta(\mathbf{q} - \mathbf{f}_t(\mathbf{x})) d\mathbf{x}, \quad (5.21)$$

where the contribution of each point is weighted by $\psi(d_t^i(\mathbf{x}))$. Since it is normalized by the sum of weights, it is indeed a generalized PDF over \mathcal{Q} ,

$$\int_{\mathcal{Q}} b_t^i(\mathbf{q}) d\mathbf{q} = 1.$$

The corresponding smooth kernel-density estimation is

$$\tilde{b}_t^i(\mathbf{q}) = \frac{1}{\int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) d\mathbf{x}} \int_{\mathcal{B}_t^i} \psi(d_t^i(\mathbf{x})) G_\sigma(\mathbf{q} - \mathbf{f}_t(\mathbf{x})) d\mathbf{x}, \quad (5.22)$$

Favoring close pixels advantageously prevents the mismatch from being affected by changes on background pixel colors as soon as these changes arise far from the object. Moreover, using the distance weight ψ allows gradual changes in local background representations. When the object moves, background points progressively "enter" into the color distribution of the band.

The short-sight version of data term E_{data} defined in Eq. (5.20), is obtained by replacing the background term with the sum of log-likelihoods on bands. The log-likelihoods are weighted with respect to the distance to the considered object. At time $t + 1$, we shall emphasize that bands \mathcal{B}_{t+1}^i and distances to region d_{t+1}^i depend on the unknown labeling ϕ_{t+1} . To properly write the short-sight data term, we make this dependency explicit using notations $\mathcal{B}_{t+1}^i(\phi_{t+1})$ and $d_{t+1}^i[\phi_{t+1}; \mathbf{x}]$, which gives

$$\begin{aligned} E_{\text{data}}^{\text{SS}}[\phi_{t+1}] = & \\ \sum_{i=1}^n \left\{ - \int_{\{\phi_{t+1}=i\}} \log \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} - \int_{\mathcal{B}_{t+1}^i(\phi_{t+1})} \psi(d_{t+1}^i[\phi_{t+1}; \mathbf{x}]) \log \tilde{b}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} \right\}. & \end{aligned} \quad (5.23)$$

According to eq. (5.22) and (5.23), as background points are considered increasingly far from the objects, the contributions of these points decrease, both in kernel density estimates \tilde{b} at time t and in the energy at time $t+1$. Notice that a single background point may be included into more than one band, and thus intervene in the short-sight background representations of several objects. This is illustrated in Fig. 5.6, which depicts overlapping bands B_{t+1}^1 and B_{t+1}^2 . Equivalently to Eq. (5.23), the short-sight data term can be rewritten in order to highlight the contribution of background points to several bands,

$$\begin{aligned} E_{\text{data}}^{\text{SS}}[\phi_{t+1}] &= - \int_{\{\phi_{t+1}=0\}} \sum_{i=1}^n \psi(d_{t+1}^i[\phi_{t+1}; \mathbf{x}]) \log \tilde{b}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} \\ &\quad - \sum_{i=1}^n \int_{\{\phi_{t+1}=i\}} \log \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x}. \end{aligned} \quad (5.24)$$

If a point \mathbf{x} is assigned to background and does not belong to any band B_{t+1}^i , then

$$\psi(d_{t+1}^i[\phi_{t+1}; \mathbf{x}]) = 0, \quad \forall i \in \{1, \dots, n\},$$

and, consequently, its contribution to $E_{\text{data}}^{\text{SS}}$ is zero.

5.4.3 Optimization using graph cuts

Approximate minimization with graph cuts

In order to minimize energy (5.17), where data term E_{data} can be either the "full-background" term (5.20) or the short-sight term (5.23), we switch to a discrete-space setting, so that an approximate minimization can be performed with the graph-cut framework [Boykov et al., 2001, Boykov and Kolmogorov, 2004, Boykov and Funka-Lea, 2006], which we summarize in the current section.

Consider ϕ and \mathbf{f} as functions defined on a discrete grid $\hat{\mathcal{D}} = \mathcal{D} \cap \mathbb{Z}^2$ of points. In the literature related to Markov Random Fields and graph-cuts, points are rather called *sites*. The set \mathcal{E} of undirected edges is the set of unordered pairs of adjacent sites, according to 4-connectivity,

$$\mathcal{E} = \{\{\mathbf{x}, \mathbf{y}\} \mid \mathbf{x}, \mathbf{y} \in \hat{\mathcal{D}}, \|\mathbf{x} - \mathbf{y}\|_1 = 1\}.$$

In the classical graph-cut framework, the data term is a sum of costs over individual sites, whereas the regularization term is set up as a sum of pairwise costs between neighboring sites,

$$E_{\text{graph-cut}}[\phi] = \sum_{\mathbf{x} \in \hat{\mathcal{D}}} U_{\mathbf{x}}(\phi(\mathbf{x})) + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})). \quad (5.25)$$

In the data term, $U_{\mathbf{x}}(l)$ is the unary cost, of assigning site \mathbf{x} to label l . It depends on image \mathbf{f} . In the regularization term⁷, $V(l_1, l_2)$ is the pairwise cost of assigning site \mathbf{x} to label l_1 and \mathbf{y} to label l_2 , for two neighboring sites \mathbf{x} and \mathbf{y} .

A suitable pairwise cost function V for segmentation is

$$V(l_1, l_2) = \begin{cases} \omega_{\text{reg}} & \text{if } l_1 \neq l_2 \\ 0 & \text{otherwise,} \end{cases} \quad (5.26)$$

where ω_{reg} is a user-defined parameter, controlling the weight of regularization over the data term. Such regularization term is known as the Potts model [Boykov et al., 2001]. If 4-connectivity is chosen, it can be viewed as the discretization of the total length of boundaries of ϕ , as in Eq. (5.16). The length is measured

⁷In the graph-cut literature, the most general form of pairwise cost is indexed by sites, $V_{\mathbf{x}, \mathbf{y}}(\phi(\mathbf{x}), \phi(\mathbf{y}))$, so that the cost can e.g. vary w.r.t $\|\mathbf{x} - \mathbf{y}\|$. Since we use a 4-connected grid, the cost is independent of site locations.

using a city-block (or Manhattan) metric, instead of the Euclidean one.

As soon as ϕ has values in $\{0 \dots n\}$ with $n > 1$ and the Potts model is used for the pairwise term, minimizing $E_{\text{graph-cut}}$ has been proven to be NP-hard [Boykov et al., 2001, Appendix]. The $\alpha - \beta$ -swap algorithm described in [Boykov et al., 2001, Section 4] can be used to find a local minimum of $E_{\text{graph-cut}}$. Starting from an initial labeling, this algorithm iteratively updates ϕ by testing swaps. An $\alpha - \beta$ -swap consists in trying to minimize $E[\phi]$ by exchanging the labels of a subset of α -labeled sites, with a subset of β -labeled sites. The best subsets to swap are determined by finding a minimum cut on a graph built from α -labeled sites and β -labeled sites⁸. If the label swap between the two subsets make $E[\phi]$ decrease, then this swap is kept in ϕ and a next swap is performed. All swaps $(\alpha, \beta) \in \{0, \dots, n\}^2$ s.t. $\beta > \alpha$ are considered.

Minimization of the short-sight data term

As regards the regularization term actually used in energy (5.17), we keep the Potts model (5.26), which is sufficient to ensure smooth object boundaries. Let us focus on the data term. If the "full-background" term (5.20) is considered, the energy can be integrated into the initial graph-cut framework (5.25), where we drop index $t + 1$ for notational simplicity. In what follows, all variables not indexed by time are considered at $t + 1$. Otherwise, time index t is made explicit. The full-background unary cost function is

$$U_{\mathbf{x}}(l) = -\log \tilde{p}_t^l(\mathbf{f}(\mathbf{x})).$$

Conversely, with the short-sight term (5.24), the graph-cut framework (5.25) cannot be used as it is. The problem is that the cost of assigning a pixel \mathbf{x} to the background does not depend on $\mathbf{f}(\mathbf{x})$ only, but on $d^i(\mathbf{x})$, which depends itself on the entire labeling ϕ . Hence, the corresponding data cost $U_{\mathbf{x}}^{\text{SS}}$ can be written as a functional of ϕ . We thus have

$$E^{\text{SS}}[\phi] = \sum_{\mathbf{x} \in \mathcal{D}} U_{\mathbf{x}}^{\text{SS}}[\phi] + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})), \quad (5.27)$$

with

$$U_{\mathbf{x}}^{\text{SS}}[\phi] = \begin{cases} \sum_{i=1}^n -\psi(d^i[\phi; \mathbf{x}]) \log \tilde{b}_t^i(\mathbf{f}(\mathbf{x})) & \text{if } \phi(\mathbf{x}) = 0 \\ -\log \tilde{p}_t^{\phi(\mathbf{x})}(\mathbf{f}(\mathbf{x})) & \text{otherwise.} \end{cases} \quad (5.28)$$

Note that $U_{\mathbf{x}}^{\text{SS}}[\phi] = 0$ if $\phi(\mathbf{x}) = 0$ and \mathbf{x} does not belong to any band $\mathcal{B}^i(\phi)$. To put $E^{\text{SS}}[\phi]$ under form (5.25), and to allow local minimization by a series of $\alpha - \beta$ -swaps, we introduce an approximation of $E^{\text{SS}}[\phi]$. It uses a temporary labeling $\tilde{\phi}$, which represents the current estimate of ϕ_{t+1} at a given iteration of the minimization procedure. An $\alpha - \beta$ swap will determine the best candidate labeling,

$$\phi^* = \underset{\phi}{\operatorname{argmin}} E_{\text{approx}}^{\text{SS}}[\phi, \tilde{\phi}],$$

where the approximate energy $E_{\text{approx}}^{\text{SS}}$ depends both on candidate labeling ϕ and temporary labeling $\tilde{\phi}$,

$$E_{\text{approx}}^{\text{SS}}[\phi, \tilde{\phi}] = \sum_{\mathbf{x} \in \mathcal{D}} U_{\text{approx}, \mathbf{x}}^{\text{SS}}(\phi(\mathbf{x})) + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})). \quad (5.29)$$

The approximate data cost $U_{\text{approx}}^{\text{SS}}$ depends on ϕ at point \mathbf{x} only. Bands and distances to regions are calculated on the fixed temporary labeling $\tilde{\phi}$,

$$U_{\text{approx}, \mathbf{x}}^{\text{SS}}(\phi(\mathbf{x})) = \begin{cases} \sum_{i=1}^n -\psi(d^i[\tilde{\phi}; \mathbf{x}]) \log \tilde{b}_t^i(\mathbf{f}(\mathbf{x})) & \text{if } \phi(\mathbf{x}) = 0 \\ -\log \tilde{p}_t^{\phi(\mathbf{x})}(\mathbf{f}(\mathbf{x})) & \text{otherwise.} \end{cases}$$

⁸The max-flow/min-cut theorem [Cormen et al., 2009, Chapter 26] implies that a minimum cut can be found by maximizing a flow in a network

The minimization procedure is adapted from the original $\alpha - \beta$ -swap algorithm. At each iteration, the best $\alpha - \beta$ swap is determined with respect to the approximate energy $E_{\text{approx}}^{\text{SS}}$. This swap is kept only if it decreases E^{SS} . Doing so, the minimized energy is indeed E^{SS} , and not $E_{\text{approx}}^{\text{SS}}$. The minimization process is summarized in Algorithm 3.

Algorithm 3: Modified $\alpha - \beta$ -swap algorithm for the minimization of the short-sight energy E^{SS}

Input: : labeling ϕ_t , images \mathbf{f}_t and \mathbf{f}_{t+1}

Output: : labeling $\tilde{\phi}$

begin

$$\tilde{\phi} := \phi_t$$

$$\text{cont} := \text{TRUE}$$

while $\text{cont} = \text{TRUE}$ **do**

$$\quad \text{cont} := \text{FALSE}$$

foreach $\text{label pair } (\alpha, \beta)$ **do**

$$\quad \quad \text{Find } \phi^* = \underset{\phi'}{\operatorname{argmin}} E_{\text{approx}}^{\text{SS}}[\phi', \tilde{\phi}] \text{ where } \phi' \text{ is an } \alpha - \beta\text{-swap of } \tilde{\phi}$$

if $E^{\text{SS}}[\phi^*] < E^{\text{SS}}[\tilde{\phi}]$ **then**

$$\quad \quad \tilde{\phi} := \phi^*$$

Update bands \mathcal{B}^i and distances to regions d^i using labeling $\tilde{\phi}$

end

end

end

end

5.4.4 Experiments and discussion

We provide experimental results on both synthetic and real image sequences. To compute kernel density estimates \tilde{p} , we accumulate color values into histograms quantized into 64 levels per channel in RGB space. The standard deviation σ of the Gaussian used in kernel-based estimation was set to 0.75. Labeling in the first frame is an available input. For each sequence, the regularization weight ω_{reg} was tuned to achieve the best segmentation. It allows to study the behavior of the VOS algorithm in the most favorable conditions.

We summarize the two different energies which are minimized, by integrating (5.17) into the graph-cut framework (5.25). Index $t + 1$ is dropped whenever possible. The same regularization term is used for the two energies. The full-background model, which is the baseline, uses (5.20) as the data term,

$$\begin{aligned} E[\phi] &= E_{\text{data}}[\phi] + E_{\text{reg}}[\phi] \\ &= \sum_{\mathbf{x} \in \hat{\mathcal{D}}} -\log \tilde{p}_t^{\phi(\mathbf{x})}(\mathbf{f}(\mathbf{x})) + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})), \end{aligned}$$

whereas the short-sight model uses (5.23) as the data term,

$$\begin{aligned} E^{\text{SS}}[\phi] &= E_{\text{data}}^{\text{SS}}[\phi] + E_{\text{reg}}[\phi] \\ &= \sum_{\mathbf{x} \in \hat{\mathcal{D}}} U_{\mathbf{x}}^{\text{SS}}[\phi] + \sum_{\{\mathbf{x}, \mathbf{y}\} \in \mathcal{E}} V(\phi(\mathbf{x}), \phi(\mathbf{y})), \end{aligned}$$

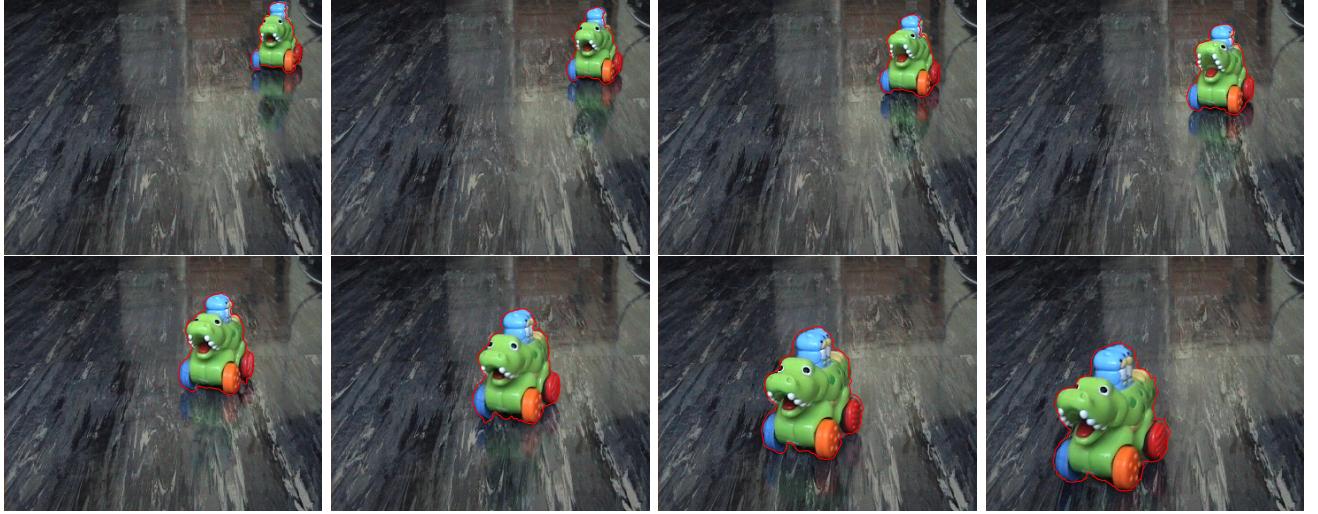


Figure 5.7: Single target tracking in a favorable case: static background, well-separated object and background color distributions

where $U_{\text{data}}^{\text{SS}}$ is defined in Eq. (5.28).

Ground truth reference segmentations were available for sequences shown in figs. 5.7 and 5.8. The accuracy obtained with the short-sight model E^{SS} was measured, with respect to ground truth, using the intersection-over-union (IoU) ratio, defined in Eq. 4.38. The 'Toy crocodile' sequence depicted in Fig. 5.7 holds a single moving object and a static background, having well-separated color distributions. Fig. 5.8 depicts the results obtained on the synthetic 'Moving disks' sequence. With this dataset, we aim to evaluate the segmentation ability of our model in the difficult case where object and background have strongly overlapping color distributions. In addition to color, the background contains non-tracked moving structures similar to the target object in terms of shape and motion (random translations and rotations). For both sequences with ground truth data, the obtained IoU ratio was around 0.98, which corresponds to fairly accurate segmentations.

The purpose of the experiment shown in Fig. 5.9 is to demonstrate the performance of our approach in case of both moving camera and dynamic object⁹. The short-sight modeling of background allows our method to be robust against gradual background changes generated by ego-motion.

Figs. 5.10 and 5.11 depict comparative results of multiple pedestrian segmentation, with $n = 3$, on a sequence taken from the PETS 2009 benchmark database¹⁰, in order to show the improvements made by the short-sight model. As ground-truth pixelwise annotations were not available to compute an IoU measure, the assessment is only visual.

Fig. 5.10 shows the labeling obtained with the full-background model E . The issue of object region leaking into the surrounding background is visible as soon as frame $t = 8$. The energy quickly becomes unable to sufficiently constrain the evolving regions and prevent leaking outside real objects. Without constraints, once the region has included tiny parts of background in a frame, it inevitably propagates into the background in subsequent frames.

Replacing the full-background term by our short-sight model managed to reduce unwanted propagation in the background, as shown in Fig. 5.11. Color distributions in bands \mathcal{B}_t^i are likely to be more compact than in the one

⁹The reference associated to the tennis woman sequence is [Brox and Malik, 2011].

¹⁰Sequence S2-L1 from <http://www.cvg.rdg.ac.uk/PETS2009>.

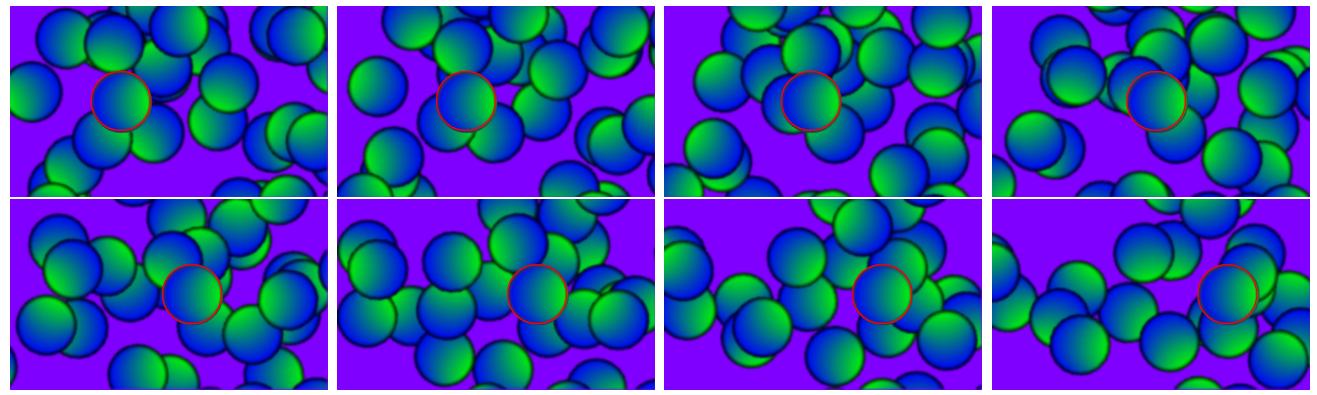


Figure 5.8: Single target tracking in case of strong overlap between moving background and object color distribution



Figure 5.9: Single target tracking in case of ego-motion

of the full background Ω_t^0 . Hence, for each object $i > 0$, the mismatch between \tilde{p}_t^i and \tilde{b}_t^i is likely to be higher than the mismatch between \tilde{p}_t^i and \tilde{p}_t^0 . This advantageously constrains regions within actual objects, as one would obtain by adding shape and motion priors (see for instance results obtained in [Chiverton et al., 2008]). Hence, in comparison with shape prior-based approaches, our method showed potential in the sense the tracking process is basically constrained by a data term without additional application-dependent shape priors.

5.5 Space-time spectral model for object detection in dynamic textured backgrounds

We now tackle unsupervised VOS based on background models. In this context, the detection of new objects appearing in the scene is taken into account, whereas the semantics of objects, and the correspondence between objects at successive frames, is left out. Hence, we turn to a foreground/background labeling problem,

$$\phi_t : \mathcal{D} \longrightarrow \{0, 1\}.$$

The contribution developed in this section is related to the supervision of student Imtiaz Ali. The PhD was linked to the application of tracking pieces of wood flowing over the surface of rivers. The motion of water makes very localized periodic stationary patterns. Hence, we headed towards space-time local frequency analysis, which we found to be well suited to represent spatiotemporal textured patches. Initially developed for VOS in river scenes, it can be applied more broadly to videos containing periodic motions that are localized in small areas. The contribution is referred to as *space-time spectral model*.

Unsupervised VOS based on background models have been reviewed in Section 5.1.2. The design of our space-time spectral background model is, firstly, motivated by the inability of pixelwise color-based background models to represent local periodic motions. Indeed, background models based on pixelwise color distributions, whether they use a parametric representation, like mixtures of Gaussians [Stauffer and Grimson, 2000] or non-parametric one, like the kernel-density estimations in [Elgammal et al., 2002], can account for the different colors appearing at a given point. However, they do not model the temporal evolution of the image. Different local motions can lead to the same color distribution through time. Secondly, background models based on the dynamic texture framework [Doretto et al., 2003] model the entire image as a time-varying process with a small set of basis state vectors. They do not model the local periodicities that are discriminant in the videos we target.

The developed background model has two main properties:

- It is *pixelwise*, in the sense that a reference background model is built at each pixel x . However, unlike background models based on color distributions, it is built from a spatiotemporal neighborhood centered at x . One or several periodic motions are assumed to take place in this neighborhood. Since neighborhoods overlap, it is expected that background properties of neighboring pixels will be similar.
- It is *stationary*. Consider a time range $[t_0, t_0 + P]$, where the period P is large enough to encompass the periods of all local motions. In the spatial neighborhood centered at a given pixel x , it is assumed that the spatiotemporal patterns, formed by local background motions, are the same, regardless of the starting time t_0 . Hence, once the background model has been learnt, it does not need to be updated during the video. Global illumination changes, which amounts to translations in the color space, can happen. The texture representation, if not completely invariant to global illumination changes, is expected to be scarcely affected by them.

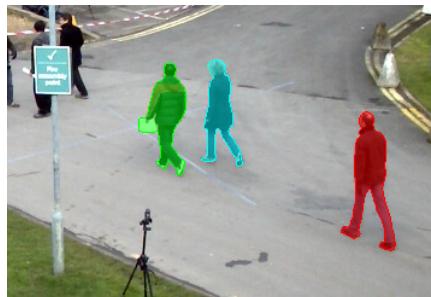
5.5.1 The local Fourier transform

In this section, we review the Fourier transform and its local version in arbitrary dimension n . Given a complex-valued integrable function $g : \mathbb{R}^n \longrightarrow \mathbb{C}$, the n -dimensional Fourier transform of g ,

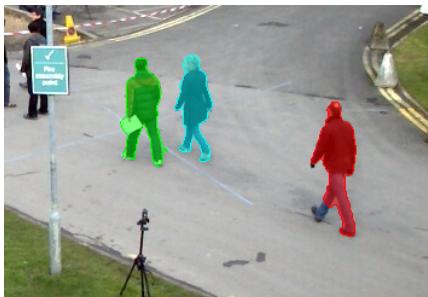
$$\text{FT}\{g\} : \mathbb{R}^n \longrightarrow \mathbb{C},$$



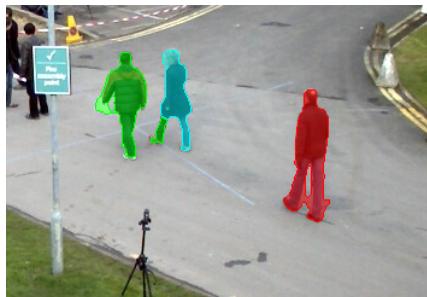
$t = 0$ (full frame)



$t = 0$



$t = 4$



$t = 8$



$t = 12$



$t = 16$



$t = 20$

Figure 5.10: Multiple object VOS with full-background model on PETS 2009 S2-L1 sequence

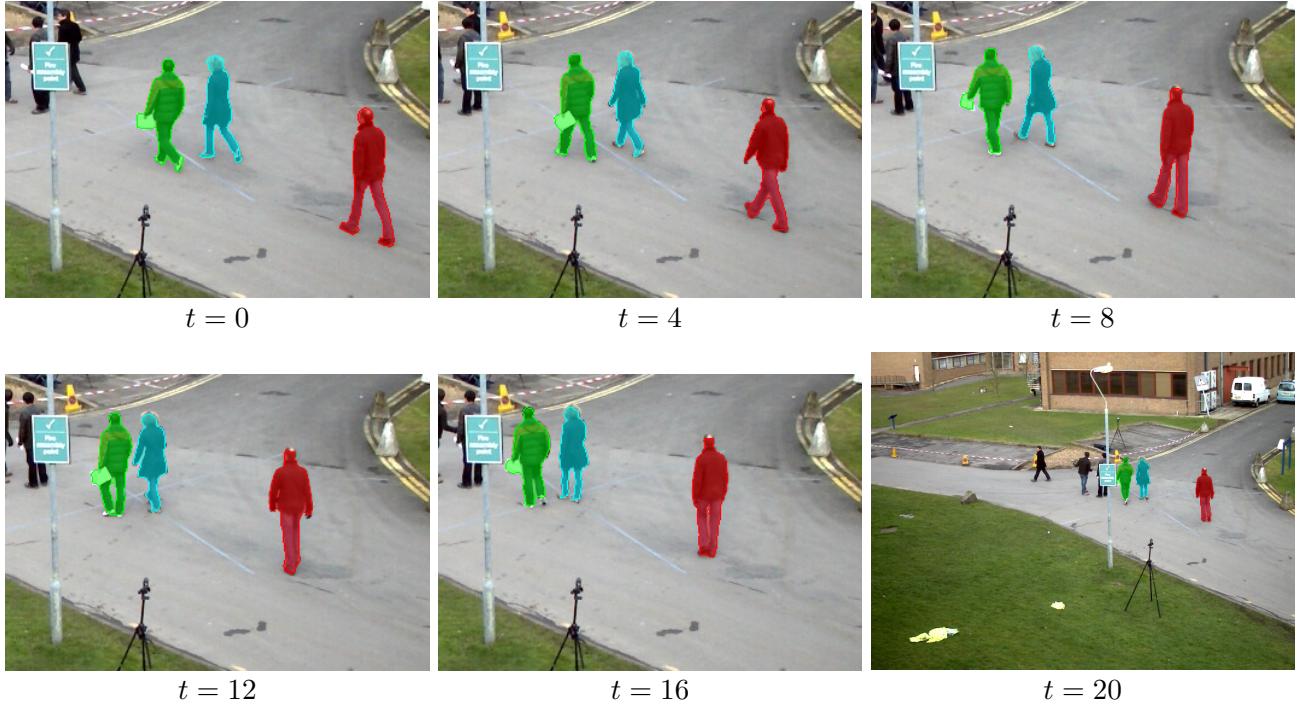


Figure 5.11: Multiple object VOS with short-sight model on PETS 2009 S2-L1 sequence

is defined, for a given frequency vector $\mathbf{u} \in \mathbb{R}^n$, as

$$\text{FT}\{g\}(\mathbf{u}) = \int_{\mathbb{R}^n} g(\mathbf{x}) \exp(-2\pi i(\mathbf{x} \cdot \mathbf{u})) \, d\mathbf{x},$$

where $i^2 = -1$, and $\mathbf{x} \cdot \mathbf{u}$ is a dot product. The Fourier transform is a linear transformation that decomposes a n -dimensional signal into its constituent frequencies. It has the interesting property of being separable, a n -dimensional Fourier transform being equivalent to n 1D Fourier transforms (one for each dimension). The plot of the magnitude, or complex modulus, $|\text{FT}\{g\}(\mathbf{u})|$, versus \mathbf{u} , is the *power spectrum* of signal g .

The Fourier transform is a global decomposition, in the sense that it represents the frequency components of the entire signal. If one wishes to have a frequency decomposition of a section of the signal g around a point \mathbf{x} , one opts for the *local Fourier transform*¹¹. The centered n -dimensional local Fourier transform of g ,

$$\text{LFT}\{g\} : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{C},$$

is defined, at point \mathbf{x} and for a frequency vector $\mathbf{u} \in \mathbb{R}^n$, as

$$\text{LFT}\{g\}(\mathbf{x}, \mathbf{u}) = \int_{\mathbb{R}^n} g(\mathbf{y}) w(\mathbf{y} - \mathbf{x}) \exp(-2\pi i(\mathbf{y} \cdot \mathbf{u})) \, d\mathbf{y}, \quad (5.30)$$

where w is a symmetric window function. Calculating a Fourier transform of a signal on a finite range amounts to "cut off" the signal outside this range. This creates artificial unwanted high-frequency components, which is known as *spectral leakage* [Harris, 1978]. The role of windowing is to reduce the undesirable effect of spectral leakage. Typically, the window function used is a non-negative, smooth, "bell-shaped" curve. One often chooses a Gaussian function centered at zero, or a function with compact support, *i.e.* zero-valued outside of an interval. An example of 1D window with compact support is the centered Hann window - or

¹¹In 1D, it is rather referred to as Short-Time Fourier Transform (STFT)

raised cosine - of radius r ,

$$w_{\text{Hann1}}(x; r) = \begin{cases} \frac{1}{2} + \frac{1}{2} \cos\left(\frac{\pi x}{r}\right) & \text{if } |x| \leq r \\ 0 & \text{otherwise.} \end{cases}$$

Its n -dimensional extension can be obtained by multiplying n 1D Hann windows in each dimension,

$$w_{\text{Hann}}(\mathbf{x}; \mathbf{r}) = \prod_{i=1}^n w_{\text{Hann1}}(x_i; r_i).$$

The use of Fourier analysis for extracting texture features was already addressed in computer vision. For example, the Gabor transform - a Fourier transform windowed by a Gaussian envelope - is used for texture segmentation in [Bovik et al., 1990]. To select appropriate Gabor filters, the power spectrum analysis of the Fourier transform of the textured image is performed [Manjunath and Ma, 1996, Puzicha et al., 1997, Wang et al., 2006]. Local Fourier transform in spatial domain is applied by Zhou *et al* [Zhou et al., 2001] for texture classification and content based image retrieval. Up to our knowledge, it has not been applied to unsupervised VOS with background models.

5.5.2 The discrete 2D+T local Fourier transform

In our background model-based VOS method, we propose to represent space-time textures using the 3D (2D+T) local Fourier transform. Our local space-time texture model can be described with a grayscale image sequence as input, $(f_t)_{t=1 \dots T}$. As will be briefly discussed in the experiments in Section 5.5.4, the role of color is negligible in the modeling of texture. As our texture modeling is based on a discretization of the local Fourier transform, we consider that each image f_t is defined on discrete spatial grid $\hat{\mathcal{D}} \subset \mathbb{Z}^2$. From now on, it is more convenient to handle time in the same way as space. Given a space-time point $\mathbf{p} = (\mathbf{x}, t)$, we use notation $f(\mathbf{p})$ in place of $f_t(\mathbf{x})$.

The discretization of the local Fourier transform (5.30) on the space-time grid is done over a discrete neighborhood. We denote the spatial and temporal extent of this neighborhood by N_s and N_t , respectively. They are user-defined parameters, chosen according to the maximal expected period of spatiotemporal textures. In order the local representation to be centered, N_s and N_t should be odd integers. The spatiotemporal neighborhood centered at space-time point $\mathbf{p} = (x, y, t)$ is the cuboid

$$\mathcal{N}(\mathbf{p}) = \left\{ x - \left\lfloor \frac{N_s}{2} \right\rfloor, \dots, x + \left\lfloor \frac{N_s}{2} \right\rfloor \right\} \times \left\{ y - \left\lfloor \frac{N_s}{2} \right\rfloor, \dots, y + \left\lfloor \frac{N_s}{2} \right\rfloor \right\} \times \left\{ t - \left\lfloor \frac{N_t}{2} \right\rfloor, \dots, t + \left\lfloor \frac{N_t}{2} \right\rfloor \right\}.$$

The Discrete Local Fourier Transform (DLFT), centered at space-time point \mathbf{p} , evaluated at frequency \mathbf{u} , is calculated on $\mathcal{N}(\mathbf{p})$ as

$$\text{DLFT}\{f\}(\mathbf{p}, \mathbf{u}) = \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} f(\mathbf{p}') w(\mathbf{p}' - \mathbf{p}) \exp(-2\pi i(\mathbf{p}' - \mathbf{p}) \cdot \mathbf{u}). \quad (5.31)$$

One may note that, in comparison to the continuous local Fourier transform in Eq. (5.30), a time shift $-\mathbf{p}$ appears in the complex exponential, in make the frequency components invariant to the central position \mathbf{p} . The chosen window is the zero-mean Gaussian function with diagonal covariance matrix

$$\begin{bmatrix} \sigma_s^2 & 0 & 0 \\ 0 & \sigma_s^2 & 0 \\ 0 & 0 & \sigma_t^2 \end{bmatrix},$$

$$w(x, y, t) = \frac{1}{(2\pi)^{\frac{3}{2}} \sigma_s^2 \sigma_t^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_s^2} - \frac{t^2}{2\sigma_t^2}\right).$$

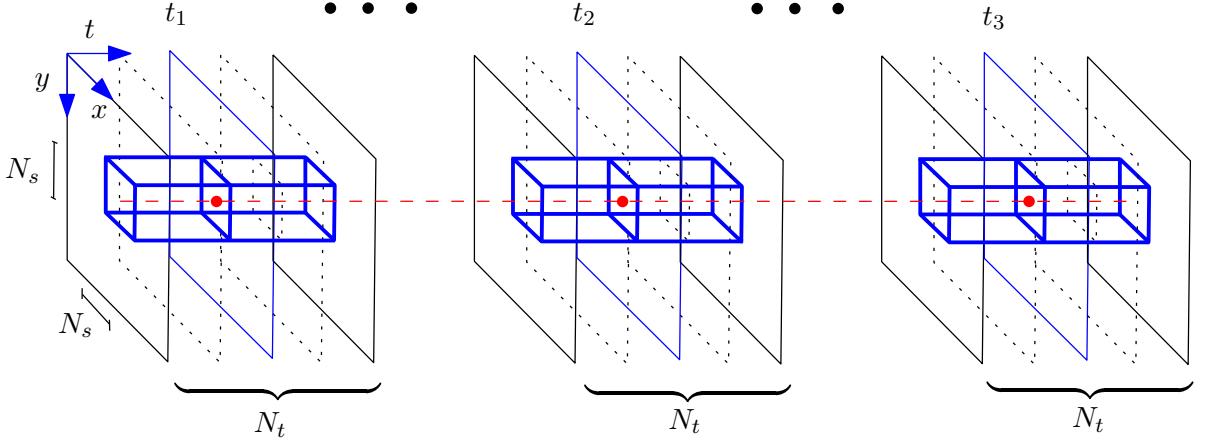


Figure 5.12: An example of sequence containing τ images for learning background. Three spectrum feature vectors $n = 3$ are learned at time instants t_1, t_2 and t_3 during the training period, i.e. $t = 1 \cdots \tau$. The location of pixel is represented by red dots in spatiotemporal window $\mathcal{N}(p)$.

We wish $w(x, y, t)$ to be negligible when $|x|$ or $|y|$ exceeds $\frac{N_s}{2}$, or similarly when $|t|$ exceeds $\frac{N_t}{2}$. It is commonly admitted that the multivariate normal, with diagonal covariance matrix, is almost zero beyond 3 times the standard deviation, in any dimension. Thus, we choose $\sigma_s = \frac{N_s}{6}$ and $\sigma_t = \frac{N_t}{6}$. As a result, we can consider that

$$\sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} w(\mathbf{p}' - \mathbf{p}) \approx 1.$$

It follows that the DLFT at frequency vector $\mathbf{0}$ is real-valued, as the complex exponential in Eq. (5.31) becomes 1. We obtain the weighted mean of the spatiotemporal patch centered at \mathbf{p} ,

$$\text{DLFT}\{f\}(\mathbf{p}, \mathbf{0}) = \sum_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} f(\mathbf{p}') w(\mathbf{p}' - \mathbf{p}).$$

The resolution of the DLFT is such that, in 1D, for an input signal of N samples, only the first N frequency components are significant [Harris, 1978]. Hence, in our 2D+T setting, the DLFT is computed for each frequency vector $\mathbf{u} = [u_1 \ u_2 \ u_3]^\top \in \mathcal{U}$, with

$$\mathcal{U} = \{0, \dots, N_s - 1\} \times \{0, \dots, N_s - 1\} \times \{0, \dots, N_t - 1\}.$$

The sequence of magnitudes of frequency components, at a given position \mathbf{p} , is referred to as *spectrum*,

$$\mathcal{S}(\mathbf{p}) = (|\text{DLFT}\{f\}(\mathbf{p}, \mathbf{u})|)_{\mathbf{u} \in \mathcal{U}}.$$

It can be viewed as a flat feature vector in $\mathbb{R}^{|\mathcal{U}|}$, and considered as the spatiotemporal texture representation of \mathbf{p} in our VOS method. To quantify the match between the space-time texture representations of two points \mathbf{p}_1 and \mathbf{p}_2 , the frequency-wise L^2 distance between their respective spectra will be used,

$$\|\mathcal{S}(\mathbf{p}_1) - \mathcal{S}(\mathbf{p}_2)\|_2^2 = \sum_{\mathbf{u} \in \mathcal{U}} (|\text{DLFT}\{f\}(\mathbf{p}_1, \mathbf{u})| - |\text{DLFT}\{f\}(\mathbf{p}_2, \mathbf{u})|)^2. \quad (5.32)$$

5.5.3 Background model

An input video is split into two portions. The first τ frames of the video are assumed not to contain any object, and are used to learn the background model. We learn K spectra per pixel at time instants $(t_j)_{j=1 \dots K}$,

such that

$$\left\lfloor \frac{N_t}{2} \right\rfloor < t_1 < t_2 < \dots < t_K < \tau - \left\lfloor \frac{N_t}{2} \right\rfloor.$$

Figure 5.12 shows the space-time neighborhoods over which training spectra are computed (in this example, $K = 3$). For our pixelwise and stationary background model, we use the same notation as in the survey of Section 5.1.2. At spatial location \mathbf{x} , it is the set of the K learnt spectra,

$$\text{BM}(\mathbf{x}) = \{\mathcal{S}(\mathbf{x}, t_j)\}_{j=1\dots K}.$$

In this context, the labeling in Eq. (5.1) is considered with $n = 1$. Identification of objects in separate frames is not required. At test time $t > \tau$, a test spectrum, $\mathcal{S}(\mathbf{x}, t)$ is compared to each training spectrum $\text{BM}_j(\mathbf{x}) = \mathcal{S}(\mathbf{x}, t_j)$. If the mismatch to the nearest spectrum, in terms of the L^2 distance defined in (5.32), exceeds a fixed threshold,

$$\min_{j=1\dots K} \|\mathcal{S}(\mathbf{x}, t) - \text{BM}_j(\mathbf{x})\|_2 > \text{th},$$

the labeling $\phi(\mathbf{p}) = \phi_t(\mathbf{x})$ is set to 1. In the next section, we assess the proposed frequency-based model on videos with dynamic backgrounds.

5.5.4 Experiments

Distribution of Fourier coefficients

We expect that neighborhoods with different space-time patterns give dissimilar spectra. To illustrate this, two background pixels \mathbf{x}_1 and \mathbf{x}_2 are marked in a frame of a grayscale video of a river bank, as depicted in Fig. 5.13. Pixel \mathbf{x}_1 is in the vegetation, whereas pixel \mathbf{x}_2 is the moving water. Space-time patterns around \mathbf{x}_1 are more static than the ones around \mathbf{x}_2 . The respective spectra $\mathcal{S}(\mathbf{x}_1, \cdot)$ and $\mathcal{S}(\mathbf{x}_2, \cdot)$, extracted at three different time instants t_1 , t_2 and t_3 , are shown (t_1 , t_2 and t_3 are not consecutive). The size of neighborhood \mathcal{N} is $N_s = 5$ and $N_t = 3$. Firstly, it can be observed that the spectra of a given point are very similar between different times, which supports our claim for a stationary model. Secondly, looking at the distribution of frequency magnitudes of $\mathcal{S}(\mathbf{x}_1, \cdot)$, they are gathered in the low end of the spectrum. Conversely, the distribution of $\mathcal{S}(\mathbf{x}_2, \cdot)$ is spread over a larger frequency range, which expresses a larger space-time motion.

We report a second experiment where we take color into account, and compute the local Fourier transform independently on each color channel, as shown in Fig. 5.14. The video frame is the same as in the previous experiment, except that color data is kept and the position of \mathbf{x}_1 is different. Whether $\mathcal{S}(\mathbf{x}_1, \cdot)$ or $\mathcal{S}(\mathbf{x}_2, \cdot)$ is considered, it can be noticed that the frequency magnitudes are extremely similar across color channels. In addition, looking at the RGB point clouds shown in Fig. 5.15, we observe that colors picked in a small spatial neighborhood, for a large duration, are gathered along the black-white diagonal of the RGB cube, which expresses colors with low saturation. The red, green and blue channels tend to have similar values. This supports the argument for extracting a texture representation on grayscale values, in order to avoid redundancy.

Projection into discriminative subspace

Since the magnitudes of neighboring frequencies are highly related, we expect to have high correlation between several components of the spectra. This leads us to study the relevance of our feature space using a dimensionality reduction technique. We use Linear Discriminant Analysis (LDA), in order to project the feature points onto a subspace that maximizes interclass variance while minimizing intraclass variance. Using the same positions \mathbf{x}_1 and \mathbf{x}_2 as in the experiment depicted reported in Fig. 5.15, we extract three groups of spectra:

- A background spectrum at \mathbf{x}_1
- A background spectrum at \mathbf{x}_2

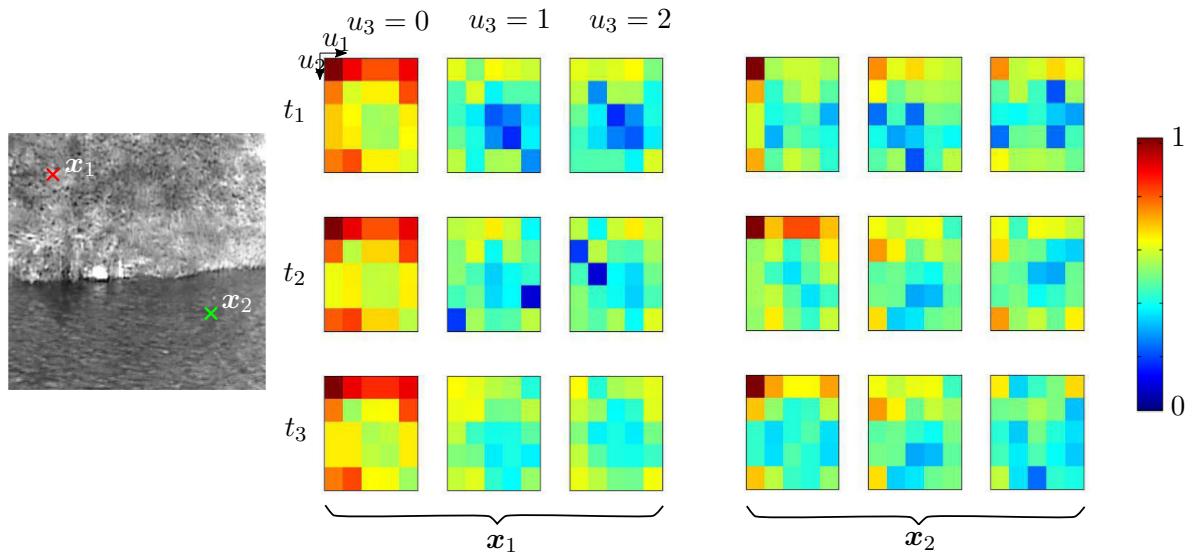


Figure 5.13: Spectra of grayscale image, at two different spatial locations x_1 and x_2 , at three time instants t_1 , t_2 and t_3 . Neighborhood \mathcal{N} has size $5 \times 5 \times 3$

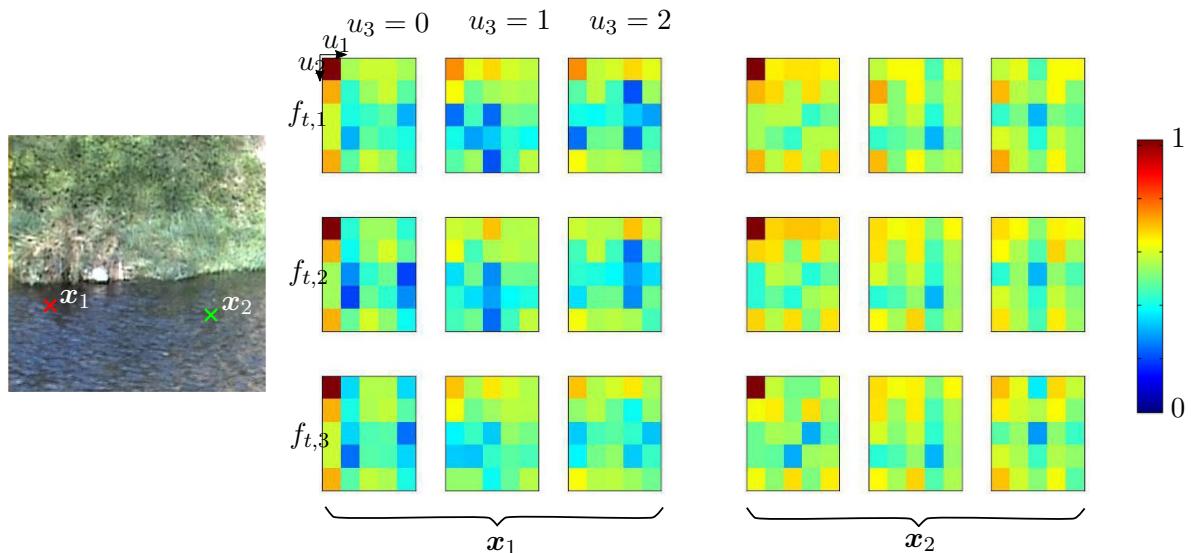


Figure 5.14: Spectra of color image, at two different spatial locations x_1 and x_2 , for Blue ($f_{t,1}$), Green ($f_{t,2}$) and Red ($f_{t,3}$) color channels. Neighborhood \mathcal{N} has size $5 \times 5 \times 3$

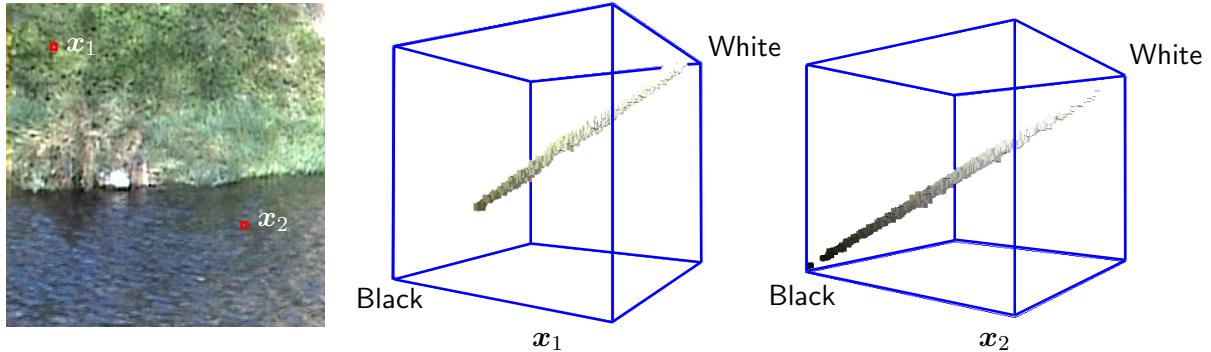


Figure 5.15: Point clouds in RGB space, taken from $5 \times 5 \times 100$ patches centered at x_1 and x_2 .

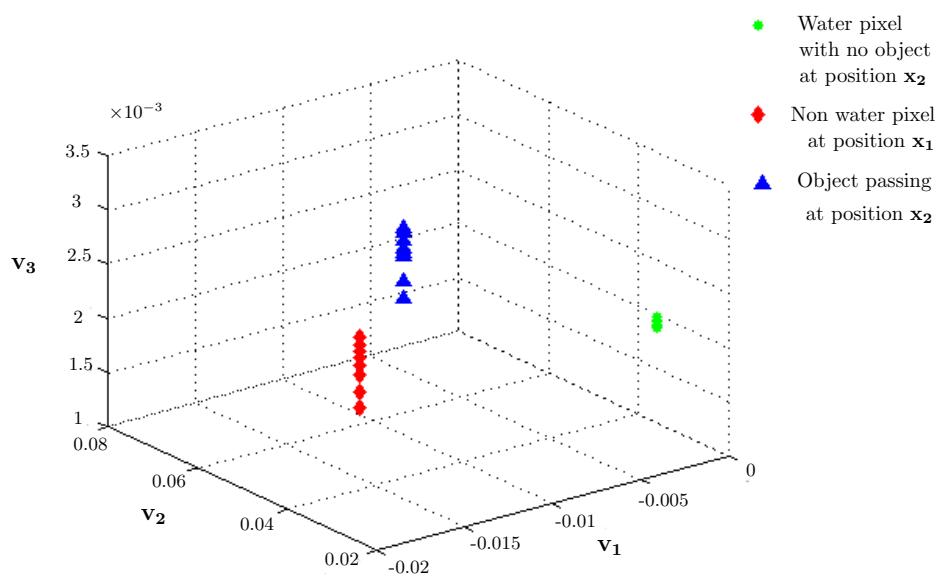


Figure 5.16: Linear discriminant analysis (LDA) for two background pixels x_1 and x_2 and one object pixel (same as in Fig. 5.13). Data is projected onto the first 3 eigenvectors.

- A spectrum at x_2 when an object passes through this position

Each group corresponds to a class. LDA assumes that the intraclass covariance is the same for all classes. We extract 10 spectra for each class, with neighborhood size $5 \times 5 \times 5$, at different time instants. We compute the sample intraclass covariance Σ_{intra} and sample interclass covariance Σ_{inter} . Fig. 5.16 shows the projections of spectrum feature vectors onto the basis (v_1, v_2, v_3) , the three first eigenvectors of $\Sigma_{\text{intra}}^{-1} \Sigma_{\text{inter}}$. It can be observed that the projected data is clustered into very well separated point clouds. It brings validating arguments to our model, in addition to the segmentation results described further.

Object detection: visual assessment

We compare the object detection obtained with our space-time spectral model, against the one obtained with the Gaussian Mixture Model (GMM) of [Stauffer and Grimson, 2000], on real image sequences. Ground truth segmentation was not available for this experiment, hence the comparison is visual. We use the same number K of spectra in our model as the number of Gaussian components in the GMM. Fig. 5.17 illustrates two sequences. The one appearing in the top rows is a home-made video with a bottle floating at the surface (BOTTLE). The one appearing in the bottom rows is taken from the Dyntex database [Péteri et al., 2010] (DUCK). The background of the DUCK sequence contains dark cast shadows of the surroundings, in addition to water ripples. A duck enters in the scene from the top right corner and moves across the scene to the middle of the image plane.

The number of Gaussians per spectrum is set to $K = 5$. The size of neighborhood \mathcal{N} was tuned to $5 \times 5 \times 11$ for both sequences. The learning rate in the GMM was set to 0.05. The obtained foreground mask shows that the space-time spectral model is able to capture the dynamics of the water region, waving grass and leaves in the background. It generates a relevant foreground detection. Conversely, the GMM generates many false detections and misses foreground pixels. It is important to mention that we do not use any regularization on the foreground masks, whether they are obtained with our model or with the GMM.

Reported execution times were obtained on an Intel Core2 Duo 2.66GHz with 4GB RAM, running a C code¹². Computation time of the training phase for the BOTTLE sequence is 89.76s (neighborhood size is $5 \times 5 \times 5$) and 124.04s for the DUCK sequence (neighborhood size is $5 \times 5 \times 11$). At test phase, the detection time per frame is 15.4s for the BOTTLE sequence (frame size is 720×576) and 9.47s for the DUCK sequence (frame size is 352×288).

Object detection: quantitative evaluation

When the approach was developed, no video dataset with full pixelwise object annotations were available to assess the accuracy of object detection. Therefore, we made our own partial annotations on these two videos. We randomly annotated $\approx 15\%$ of the video frames, and computed the IoU score on these frames. We obtained 0.81 on the DUCK sequence and 0.75 on the BOTTLE sequence with our space-time spectral model. We can remark that in the BOTTLE video, the average IoU value is smaller due to the reflections in water that create some false detections. With the GMM, the IoU scores is 0.11 for the DUCK sequence and 0.14 for BOTTLE.

In addition, we generated synthetic sequences from textured backgrounds, with object masks, using samples from the DynTex database, as shown in Fig. 5.18. Despite the limitations of assessing accuracy on synthetic data, it allowed to compute an IoU score on all frames of sequences. In the two sequences shown, a square block of size 30×30 moves from left to right. The block is simultaneously translated and rotated with an angle of 5 degrees clockwise per image in 100 consecutive images. In the WHEAT sequence, the spatial texture of the object is different from the one of the background. In the ESCALATOR sequence, the object

¹²The FFT was computed using the FFTW library, www.fftw.org

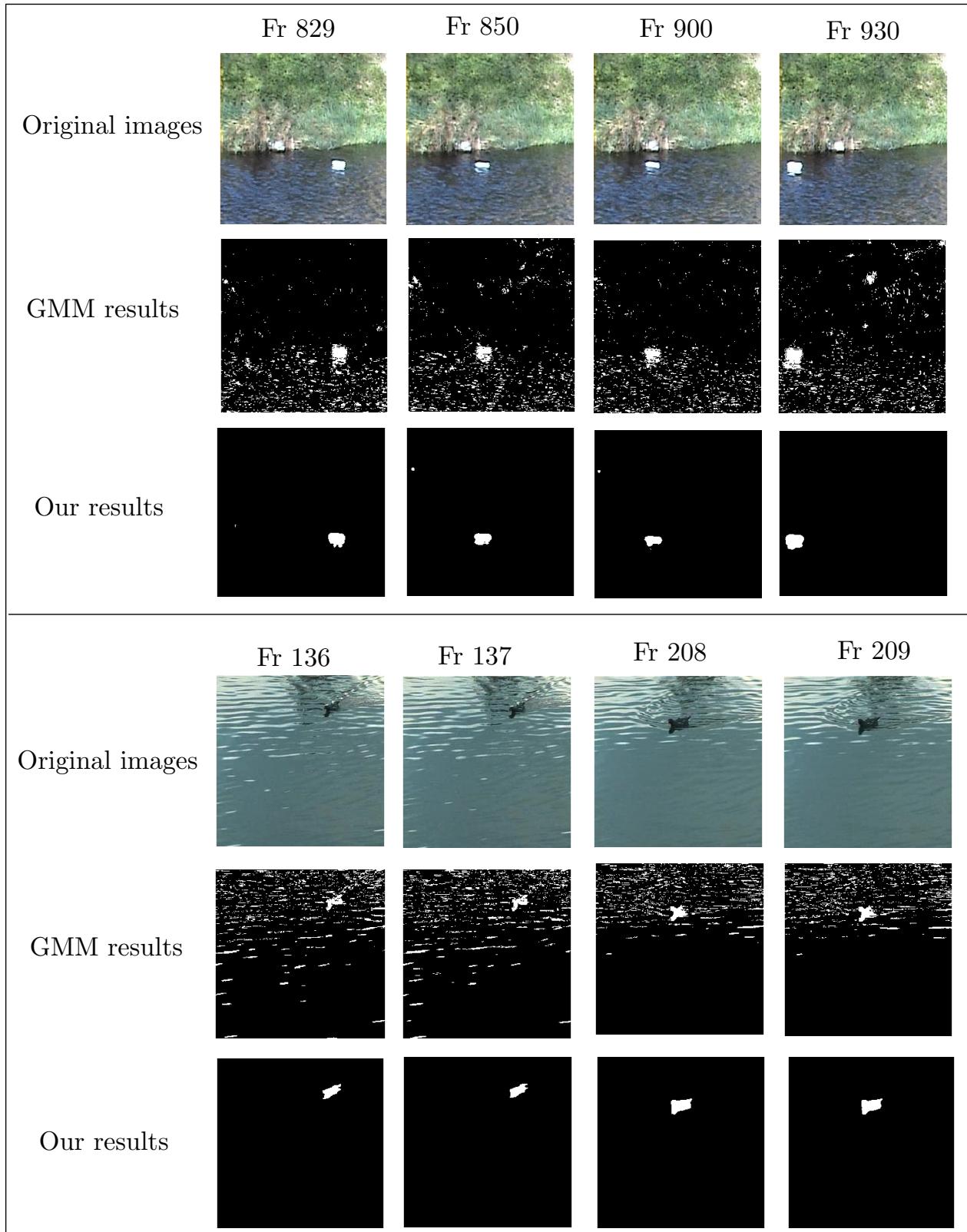


Figure 5.17: Qualitative comparison of object detection on the BOTTLE sequence (top) and DynTex DUCK sequence (bottom)

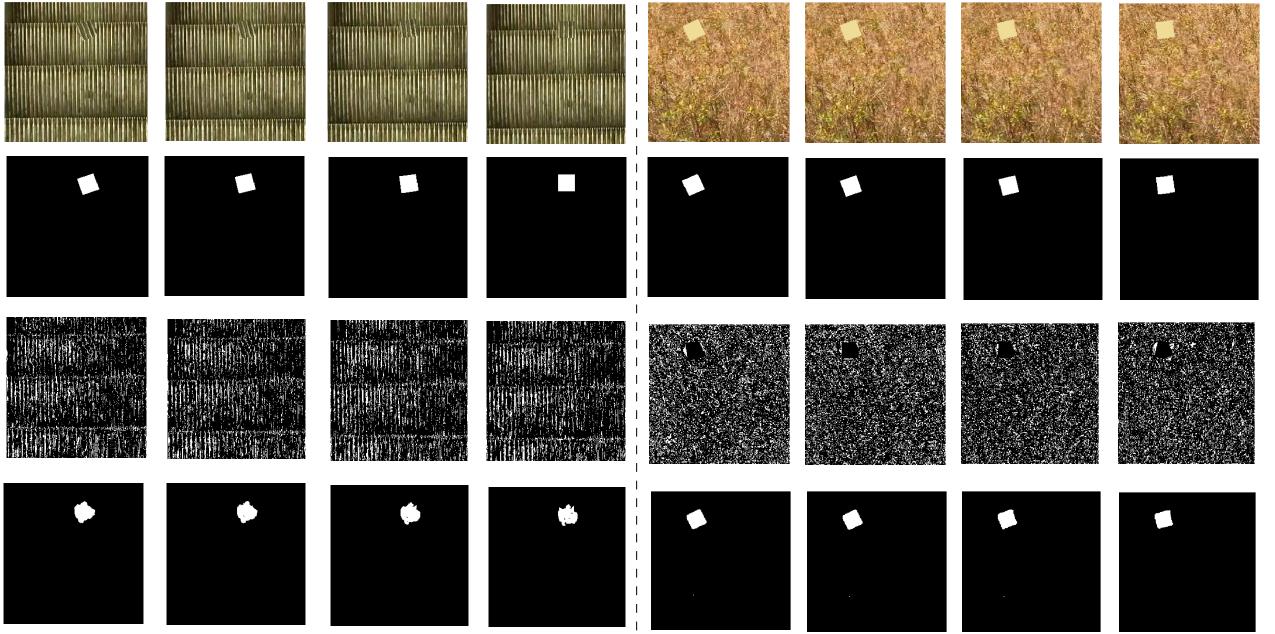


Figure 5.18: *First row* A synthetic square of size 30×30 moving from left to right with (*left*) moving escalator and (*right*) moving wheat field in the background, *Second row* corresponding ground truth images, *Third row* Detection obtained with GMM and *Last row* Detection obtained with our space-time spectral model.

has the same spatial texture as the background, but the texture motion is different between object and background. The background texture moves from top to bottom, whereas the one of the object undergoes rotation. We use different sets of images for training and detection.

The IoU ratio obtained with our space-time spectral model, averaged over all frames, is 0.87 for the ESCALATOR sequence and 0.96 for the WHEAT sequence. The space-time spectral model can be used to detect an object even if it has similar colors as the background. With the GMM, we obtained 0.03 for ESCALATOR and 0.01 for WHEAT, which shows clear superiority of the proposed approach. We give the computation time taken by our method during the training period and object detection. The image size is 256×256 . We compute $K = 8$ spectra for per pixel. Computation time during training period for 193.70s for the ESCALATOR (neighborhood size is $7 \times 7 \times 11$) and 89.76s for the WHEAT sequence (neighborhood size is $5 \times 5 \times 5$). Detection time per frame are 15.4s and 9.47s, respectively. One may note that the detection time taken by the GMM is in the order of 200ms.

Influence of neighborhood size

We finally study the influence of neighborhood size parameters N_s and N_t on the synthetic ESCALATOR sequence. These parameters control the spatial and temporal extent of patterns that can be captured. The obtained segmentations with different values of N_s and N_t are shown in Fig. 5.19. A single frame is shown for each setting. We use odd values from 1 to 9 for N_s , and from 1 to 11 for N_t . The effects of changing the size of the spatiotemporal neighborhood can be observed. When $N_s = 1$, it implies that no spatial neighborhood per pixel is considered, which boils down to extracting purely temporal patterns. The escalator motion in the background is slow and increasing the value of N_t enables the background model to capture the periodicity of the moving escalator.

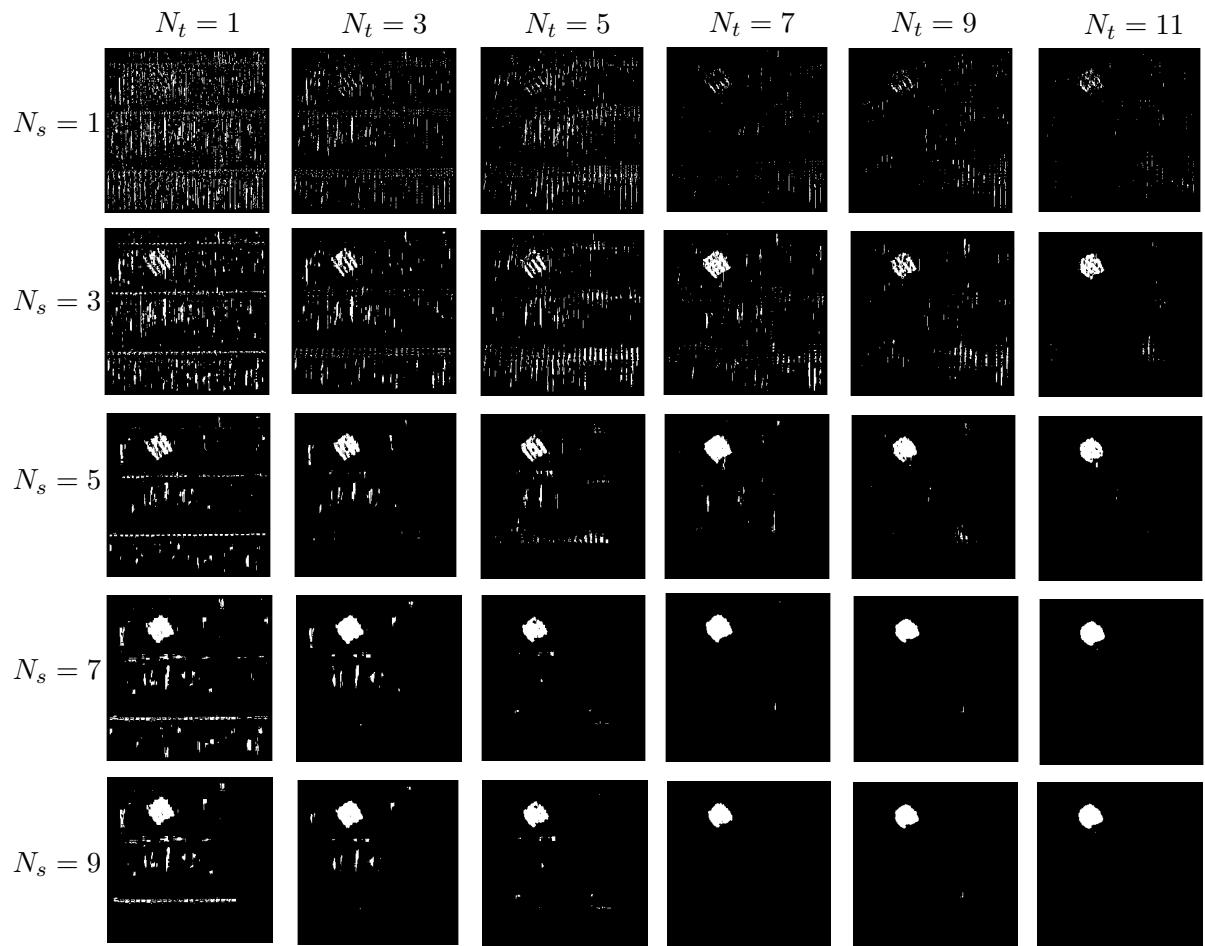


Figure 5.19: Detection results with different values of spatial extent N_s and temporal extent N_t of neighborhood \mathcal{N} .

5.6 Conclusion

5.6.1 Summary of contributions

In the first contribution, multiple VOS was tackled under the angle of joint segmentation and tracking. We started from a Bayesian interpretation of VOS, in order to derive a model based on a probabilistic interpretation. VOS is performed by matching color densities of objects and the background through successive frames. Considering the color density in the whole background raises a major shortcoming, due to the scattering of color statistics, which can happen in cluttered backgrounds. The intuition was that favoring close pixels prevents the object/background mismatch from being affected by changes on background pixel colors as soon as these changes arise far from the object.

Hence, we introduced a *short-sight perception* modeling of background, focusing on the neighborhood of tracked objects to extract consistent statistical data for accurate separation between objects and background. In the energy formulation, the full background was replaced by bands around objects. Within bands, distance weighting allows progressive changes in local background representations. Approximate minimization was performed using a modification of the $\alpha - \beta$ -swap graph cut algorithm.

In the second contribution, we presented a novel frequency-based background model for unsupervised VOS. Such model was dedicated to moving backgrounds containing repetitive space-time patterns. We believe that spatially varying and time repetitive textures in the background are very efficiently modeled using frequency-based method.

Thus, we considered spatiotemporal neighborhoods of pixels, on which we applied the Discrete Local Fourier Transform (DLFT). A pixelwise background model was built by considering the magnitudes of the DLFT spectrum as features. We applied our method for moving object detection in moving backgrounds, where the number of objects was unknown. Experiments were conducted on both synthetic data, generated from the DynTex dataset, and real data. We obtained good accuracy in foreground/background segmentation and outperformed the commonly-used Gaussian Mixture-based background model. In outdoor scenarios, our frequency-based background model leads to better detection and segmentation than the GMM method, which fails to capture time-repetitive background motions.

5.6.2 Possible research tracks

Explicit modeling of shape, occlusion and motion Contributions on video object segmentation were developed over the 2010-2012 period. Back in those days, MRFs and graph-cut-related methods were still a *de facto* standard for formulating and minimizing energies for image/video segmentation. An important part of the literature focused on designing additional energy terms implementing constraints and prior knowledge on the properties of tracked objects [Aeschliman et al., 2010, Brox and Malik, 2011, Papadakis and Bugeau, 2011] and perform approximate minimization using graph cut algorithms.

As a possible extension to the first contribution, discussions we had, at the time, dealt about incorporating priors on the temporal smoothness of shape and motion [Brox and Malik, 2011, Wen et al., 2015], and/or handle occlusions [Papadakis and Bugeau, 2011]. It is likely that the short-sight VOS model, solely based on color distributions, fails when several objects with similar appearance collide. Embedding time-consistency of shape and motion could limit the unwanted leakage of one region into the other object, and thus maintain relevant segmentation. In this case, additional terms could be added into energy (5.24). We might incorporate motion estimation as well, as in [Brox et al., 2006], in order that color density tracking and optical flow benefit from each other.

Regardless of the extensions that collaborators and I wished to investigate at the time, how do these rather old-fashioned kinds of method compete against modern CNN-based methods [Zhao et al., 2019, Chen et al., 2020b] yielding impressive results on challenging datasets [Cordts et al., 2016]? If they cannot compete, is it possible to integrate similar mathematical models in deep convolutional networks? This

constitutes longer-term perspectives, which are discussed in Chapter 7.

Back to the probabilistic framework Considering the detailed mathematical derivation of Sections 5.3.2 and 5.4, unpublished before, it appears that we lost the probabilistic interpretation, by switching from the full-background data term to the short-sight data term.

Indeed, the transformation of the full-background model (5.20) to the short-sight model (5.23) turns out to be in contradiction with a previous assumption that allowed to simplify the likelihood term (5.18) to (5.19). Between these two steps, the assumption was that the likelihood of observing $\mathbf{f}_{t+1}(\mathbf{x})$ depends on \mathbf{f}_t , ϕ_t and next labeling at \mathbf{x} only,

$$\Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}) = \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}(\mathbf{x})),$$

which, in energy (5.20), underlies the fact that each point \mathbf{x} has a unique contribution to the data term, whether it belongs to an object or the background. The transformation from the full model (5.20) to the short-sight model (5.23) invalidates this simplifying assumption, since $\mathbf{f}_{t+1}(\mathbf{x})$ depends on the entire labeling $\mathbf{f}_{t+1}(\mathbf{x})$ if \mathbf{x} belongs to the background. Incidentally, a single background point can have multiple contributions to energy (5.23) if it belongs to several bands. As a consequence, it invalidates (5.19). Whether we should try to keep the probabilistic framework, or remove it in favor of the energy as a starting point is an open question. If the probabilistic framework is to be kept, a possible clue to link (5.18) to (5.19) again, in accordance with the short-sight model, would consist in making the dependence on position \mathbf{x} explicit for background points, which should lead to an expression like

$$\begin{aligned} & \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}, \phi_{t+1}(\mathbf{x}) = 0, \mathbf{x}) \\ &= \begin{cases} 1 & \text{if } \mathbf{x} \notin \bigcup_{i=1}^n \mathcal{B}_{t+1}^i \\ \prod_{i=1}^n \Pr(\mathbf{f}_{t+1}(\mathbf{x})|\mathbf{f}_t, \phi_t, \phi_{t+1}, \phi_{t+1}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_{t+1}^i) & \text{otherwise.} \end{cases} \end{aligned}$$

Further investigations need to be done on this aspect, to study whether it has a relevant probabilistic interpretation, and not just being a mathematical twist to bring extra justification on the short-sight background model.

Chapter 6

The object as a given input: skeleton-based shape matching

Once an object is segmented, one often wishes to assign it to a known object category, by means of feature extraction over its shape and appearance, and feeding the feature representation to a classifier. In this chapter, we focus on the purpose of planar shape classification, *i.e.* the assignment of a planar *query* object to a given class. This classification task is performed by matching the query object, based solely on its shape, to a set of shapes representative of each class. The query object is finally assigned to the class containing the best matches, in a nearest-neighbor fashion.

One of the difficulties in extracting relevant shape features, or *descriptors*, is that real objects should be recognized from their 2D projections on the image plane, as in Fig. 6.1. The planar shape varies according to properties which are intrinsic to the object, such as the relative orientations of articulated body parts, but also extrinsic properties, *i.e.* the point of view from which the object is captured. The ideal shape descriptor should have the following properties:

- discrimination,
- informativity,
- invariance, or at least low sensitivity, to affine transformations (translation, scaling, rotation, symmetry...),
- low sensitivity to small non-linear deformations (noise, articulated motions, occlusions),
- compactness,
- ease of extraction and use,
- ability to distinguish the importance of different parts.

Owing to its efficiency in representing shapes, the skeleton, or medial axis [Blum and Nagel, 1978], was extensively used for shape matching, classification and indexing. Among several advantageous properties over the shape contour itself, it lends itself to the design of shape features with a certain degree of invariance to articulated deformations and reorganization of shape parts [Bai and Latecki, 2008, Sebastian and Kimia, 2005]. A desirable property of a shape descriptor is stability to small contour deformations. In its basic form, the skeleton is intrinsically unstable to contour deformations, as shown in Fig. 6.2. Generating a skeleton that is stable, thin and homotopic to the shape has been the focus of an important part of literature related to computer graphics, discrete geometry and computational geometry. A number of recent methods focus on computing a skeleton that captures the main parts of the shape while being stable with respect to global transformations or small local contour deformations [Chazal and Lieutier, 2005, Jalba et al., 2016,

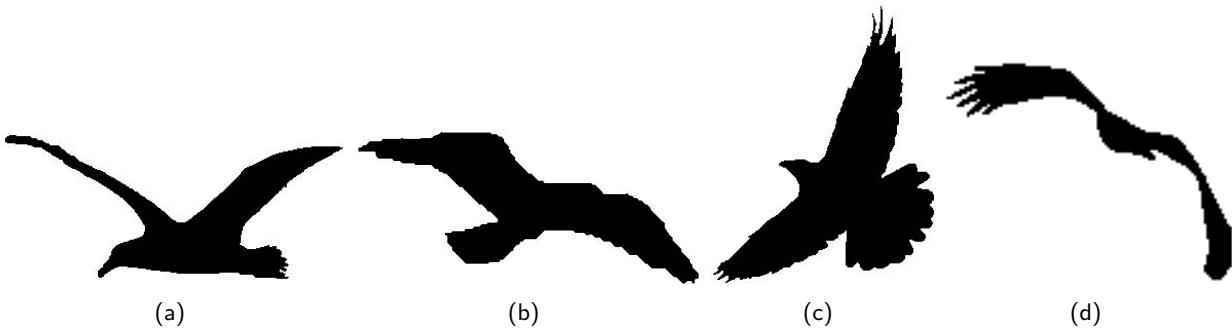


Figure 6.1: Shapes varying according to articulated deformations and change of viewpoint [Yang et al., 2016].

Marie et al., 2016]. These advantageous features are exploited in shape recognition and matching methods [Bai and Latecki, 2008, Macrini et al., 2011b, Sebastian et al., 2004, Yang et al., 2016].

The contributions described in this chapter have their origin in the supervision of PhD student Aurélie Leborgne, in the 2012-2016 period, on skeleton-based shape matching. The first part of Aurelie's PhD tackled skeletons of discrete shapes in \mathbb{Z}^2 . An original skeletonization algorithm was developed, based on the Euclidean distance transform of the shape, generating a skeleton which was homotopic to the shape, thin (one-pixel thickness), and robust to noise. The algorithm, referred to as Digital Euclidean Connected Skeleton (DECS), was published in [ISVC14, JVCIR15]. The contribution on shape matching, described in this chapter, is an extended variant of the initial publication [ICIP16], which used DECS.

Skeleton branches correspond to ridges of the distance map, *i.e.* continuous lines of points that are local maxima of the distance in at least one direction. Extracting these ridges is a non-trivial task on a discrete grid. The *average outward flux* (AOF), used in the Hamilton-Jacobi skeleton [Siddiqi et al., 2002] is used as a ridge detector. In [JVCIR15], we proposed the *ridgeness* measure, without relating it to the AOF. The first main contribution, described in this chapter, is to establish the mathematical relation between these detectors and, extending the work in [Dimitrov et al., 2003], studying various local shape configurations, on which closed-form expressions or approximations of the AOF and ridgeness can be derived. In addition, we conduct experiments to assess the accuracy of skeletons generated using these measures, and study the influence of their respective parameters. The related publication in Section 1.5 is [JMIV19].

In the second contribution, we address shape matching exploiting the natural hierarchy of the skeleton. Using successive smoothing operations and deformation of the initial skeleton, we construct a hierarchical skeleton, in which the importance of branches is quantified. This allows to guide the shape matching algorithm in order to match branches having the same order of importance. To perform matching, our method is founded, on the one hand, on the intrinsic characteristics of the branches, and on the other hand, on the context, represented by their relative positions to other branches. Matching is set as an assignment in a bipartite graph, which is solved using the Hungarian algorithm [Kuhn, 1955, Munkres, 1957]. Related publications in Section 1.5 are [ICIP16, RFIA16].

6.1 Ridge detection in Euclidean distance maps

6.1.1 The Euclidean skeleton

Let $\Omega \subset \mathcal{D}$ be a shape, open subset of the image domain. The Euclidean distance transform

$$D : \Omega \longrightarrow \mathbb{R}^+,$$

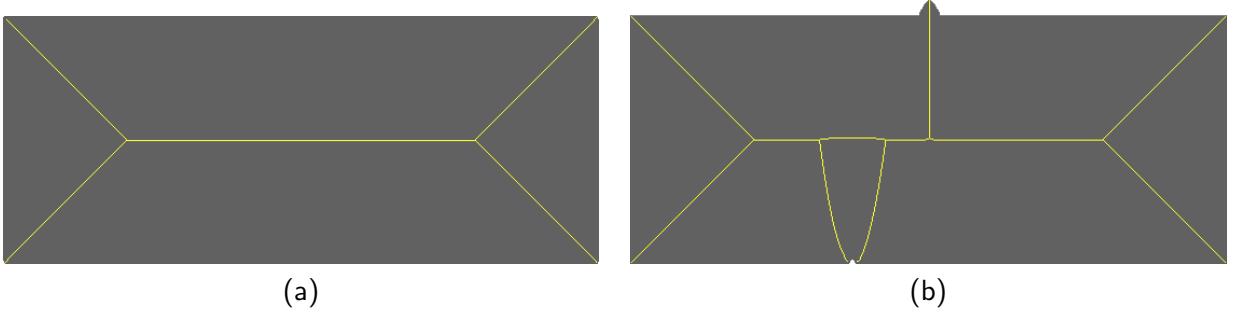


Figure 6.2: (a) Skeleton of a rectangle, (b) Skeleton of the same rectangle to which two perturbations have been added.

at a given shape point x , is the Euclidean distance between x and the nearest point on the shape border $\partial\Omega$,

$$D(x) = \min_{y \in \partial\Omega} \|x - y\|.$$

Several equivalent definitions of the continuous skeleton of Ω exist. Among these, one definition is based on maximally inscribed balls. The Euclidean skeleton, or medial axis, S is the subset of Ω containing centers of maximal balls, or equivalently, balls having at least two distinct contact points on the shape border [Jalba et al., 2016]. The radius of the maximal ball centered at a skeleton point s being $D(s)$, the skeleton is defined as

$$S = \{s \in \Omega \mid \exists p, q \in \partial\Omega, p \neq q, \|s - p\| = \|s - q\| = D(s)\} \quad (6.1)$$

Equivalently, the skeleton can be defined as the set of critical points of the Euclidean distance transform D . Starting from the shape border, if one considers the evolution of a curve in the inward normal direction, the skeleton is the set of locations where fronts collide, namely *shocks* [Siddiqi et al., 1999]. Equivalently, skeleton branches correspond to ridges, or crest lines, of the distance map. In other words, skeleton points are local maxima of D in at least one direction, and ∇D is undefined at these points.

The choice of a skeletonization algorithm depends on the available representation of the shape. When the data available is the border sampled in $\mathbb{R}^{n \in \{2,3\}}$ - typically, a polygon when $n = 2$ or a triangulated mesh when $n = 3$ - one would use Voronoi diagram-based algorithms [Lee, 1982, Brandt and Algazi, 1992, Ogniewicz and Kübler, 1995, Amenta et al., 2001]. On the other hand, when the data available is a discrete shape, i.e. a subset of $\mathbb{Z}^{n \in \{2,3\}}$, it is preferable to choose from thinning procedures [Arcelli and Sanniti di Baja, 1985, Lam et al., 1992, Bertrand and Malandain, 1994, Palágyi, 2008], which iteratively remove border points with topological conditions, and/or distance-based methods [Arcelli and Sanniti di Baja, 1992, Kimmel et al., 1995, Ge and Fitzpatrick, 1996, Hesselink and Roerdink, 2008], which typically detect local maxima of the Euclidean distance transform. Furthermore, distance-based and thinning methods are not mutually exclusive, as some methods combine both aspects, e.g. [Pudney, 1998, Siddiqi et al., 2002]. This is also the case of the methods studied in this section.

The extraction of ridges of the Euclidean distance map, which is the focus of this section, dates as far back as [Arcelli and Sanniti di Baja, 1992]. The problem was initially addressed in a discrete *ad hoc* manner, in which a set of discrete kernels were designed to extract ridges. Later, it was formulated in a consistent, continuous framework in the Hamilton-Jacobi skeleton [Siddiqi et al., 2002], where the *Average Outward Flux* (AOF), a measure of local divergence of the distance map, was used to distinguish skeleton points from non-skeleton points.

The AOF measure was combined with an homotopy-preserving thinning process. Starting from the border, points are iteratively removed by ascending order of AOF, in absolute value. Points with strong

AOF being located on local maxima of the distance map, the obtained skeleton is consistent in the Euclidean sense. Numerous works build upon the Hamilton-Jacobi skeleton, such as 3D centerline extraction [Bouix et al., 2005], shock graphs for shape matching [Sebastian et al., 2004] and the curvature-density correction of [Torsello and Hancock, 2006]. Theoretical values of the AOF, known as *flux invariants*, were studied in [Dimitrov et al., 2003], for a number of local configurations of planar shapes.

In contribution [JVCIR15], we introduced the *ridgeness* measure. In this method, candidate skeleton points are extracted by filtering the Euclidean distance map with a negative Laplacian of Gaussian (LoG) kernel. Local maxima of the distance map have thus strong ridgeness. In [JVCIR15], we applied hard thresholding to remove points with insufficient ridgeness. This results in significantly lower complexity in comparison to iterative thinning, but to the detriment of connectivity, as hard thresholding may disconnect skeleton branches. In order to guarantee homotopy to the shape, a reconnection step was added, based on a criterion combining ridgeness and centers of maximal balls.

To generate a skeleton relevant for recognition tasks, a common processing step is the pruning and/or hierarchization of skeleton branches [Bai et al., 2007, Yang et al., 2016], as will be reviewed with more details in Section 6.3.1. As regards distance-based skeletonization algorithms, the ridge detection step is crucial, as it affects the amount of branches to be pruned afterwards. In their respective methods, the AOF and the ridgeness are thresholded at some stage. Points selected by this thresholding step are retained as candidate skeleton points. A loose thresholding will retain many points, resulting in a possibly high amount of undesirable branches, while an excessive thresholding might remove significant branches. Both measures have a parameter, related to their spatial extent, which impacts the detection. Thus, a principled way of determining the threshold, both according to the spatial parameter and the desired degree of branching, by means other than simple empirical study, would be of significant value.

We believe that studying the behavior of the skeleton measures can provide insights on how to choose their thresholds appropriately. We therefore conduct an analytical study of the measures on a set of local *theoretical* shape configurations (regular skeleton points, endpoints, etc). We make multiple contributions. First, we establish the mathematical relation between the AOF and our ridgeness measure. Then, for the AOF, we provide mathematical derivations extending the flux invariants of Dimitrov *et al* [Dimitrov et al., 2003]. We express this particular contribution more explicitly in Section 6.1, once the AOF is defined. As regards the ridgeness, we provide completely new invariants. Some configurations, like ligatures [Katz and Pizer, 2003, Macrini et al., 2011a, Pizer et al., 2003] - connections between skeleton branches - are often problematic in skeleton extraction and skeleton-based shape matching. Unlike other local shape configurations, the case of ligature was not studied in [Dimitrov et al., 2003]. Therefore, we provide an analysis of AOF and ridgeness measures for this particular case. Finally, we report the experiments that we conducted on a shape dataset, in which we vary parameters and thresholds to corroborate our derivations, and compared the performances of AOF-based and ridgeness-based skeletons.

The AOF and the ridgeness are local detectors of ridges of D . As they imply first and second order differentiation, at a point \mathbf{x} where D is twice differentiable - which is true almost everywhere - the Laplacian, divergence of gradient and Hessian matrix are linked as follows:

$$\text{div } \nabla D(\mathbf{x}) = \Delta D(\mathbf{x}) = \text{tr}(\nabla^2 D)(\mathbf{x})$$

6.1.2 Average outward flux

In a given region B , the outward flux of a vector field \mathbf{v} is the amount by which vectors of \mathbf{v} point towards the exterior of B . Naturally, the outward flux of ∇D is close to zero in regions located on linear slopes of D (non-skeleton points), whereas it becomes highly negative on ridges of D (skeleton points). [Siddiqi et al., 2002] defined the skeleton likeliness as the AOF of ∇D , i.e. the outward flux in region $B(\mathbf{x})$ centered at \mathbf{x} ,

normalized by the length of the boundary of $B(\mathbf{x})$:

$$\text{aof}(\mathbf{x}) = \frac{1}{|\partial B(\mathbf{x})|} \int_{\partial B(\mathbf{x})} \nabla D \cdot \mathbf{n} \, ds \quad (6.2)$$

where ds is an element of the boundary $\partial B(\mathbf{x})$ of the region $B(\mathbf{x})$ and \mathbf{n} is the outward normal along this boundary. Notice that the boundary $\partial B(\mathbf{x})$ can contain critical points of D , where D is not differentiable. However, the subset of $\partial B(\mathbf{x})$ where ∇D is undefined is Lebesgue-measurable [Royden and Fitzpatrick, 2010, Chapter 2], and has measure zero¹, hence $\nabla D \cdot \mathbf{n}$ is integrable in the sense of Lebesgue. Via the divergence theorem,

$$\begin{aligned} \text{aof}(\mathbf{x}) &= \frac{1}{|\partial B(\mathbf{x})|} \int_{B(\mathbf{x})} \operatorname{div} \nabla D(\mathbf{y}) \, d\mathbf{y} \\ &= \frac{1}{|\partial B(\mathbf{x})|} \int_{B(\mathbf{x})} \Delta D(\mathbf{y}) \, d\mathbf{y}. \end{aligned} \quad (6.3)$$

In [Siddiqi et al., 2002], region B is chosen as a ball of constant radius r . In the 2D case, using Eqs. (6.2) and (6.3), the AOF is written as an integral over the circle of radius r , spanned by angle θ :

$$\text{aof}(\mathbf{x}, r) = \frac{1}{2\pi} \int_0^{2\pi} \nabla D \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \, d\theta \quad (6.4)$$

The AOF is highly negative for skeleton points and close to zero for non-skeleton points.

6.1.3 Ridgeness

The n -dimensional Laplacian of Gaussian (LoG) filter is

$$\Delta G_\sigma(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \sigma^{n+2}} \left(\frac{\|\mathbf{x}\|^2}{\sigma^2} - n \right) \exp \left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2} \right),$$

which is the Laplacian of a zero-mean n -dimensional Gaussian, with isotropic covariance matrix $\Sigma = \sigma^2 \mathbf{I}$, where \mathbf{I} is the $n \times n$ identity matrix. In [JVCIR15], we defined the ridgeness of a 2D distance transform, at a given scale σ , as the negative of the LoG-filtered distance transform:

$$\text{rdg}(\mathbf{x}, \sigma) = -(D * \Delta G_\sigma)(\mathbf{x}), \quad (6.5)$$

where $*$ is the convolution operator over \mathbb{R}^2 . Assuming that the distance function is extended over the entire domain \mathbb{R}^2 ,

$$\text{rdg}(\mathbf{x}, \sigma) = - \int_{\mathbb{R}^2} D(\mathbf{y}) \Delta G_\sigma(\mathbf{x} - \mathbf{y}) \, d\mathbf{y}. \quad (6.6)$$

As D is twice differentiable almost everywhere, rdg can be equivalently defined as

$$\text{rdg}(\mathbf{x}, \sigma) = - \int_{\mathbb{R}^2} \Delta D(\mathbf{y}) G_\sigma(\mathbf{x} - \mathbf{y}) \, d\mathbf{y}. \quad (6.7)$$

The ridgeness is high for skeleton points and close to zero for non-skeleton points. As for the AOF, the set of points where ΔD is undefined is Lebesgue-measurable and has measure zero². Hence, the integral in Eq. (6.7) is well defined in the sense of Lebesgue.

As the ridgeness measure implies convolution of D with an infinite support kernel, the distance map is extended outside shape Ω , so that it is defined everywhere. We thus consider the signed distance transform

$$D(\mathbf{x}) = \begin{cases} \min_{\mathbf{y} \in \partial \Omega} \|\mathbf{y} - \mathbf{x}\| & \text{if } \mathbf{x} \in \Omega \\ -\min_{\mathbf{y} \in \partial \Omega} \|\mathbf{y} - \mathbf{x}\| & \text{if } \mathbf{x} \notin \Omega \end{cases} \quad (6.8)$$

¹It is a set of isolated points

²It is a set of portions of curves

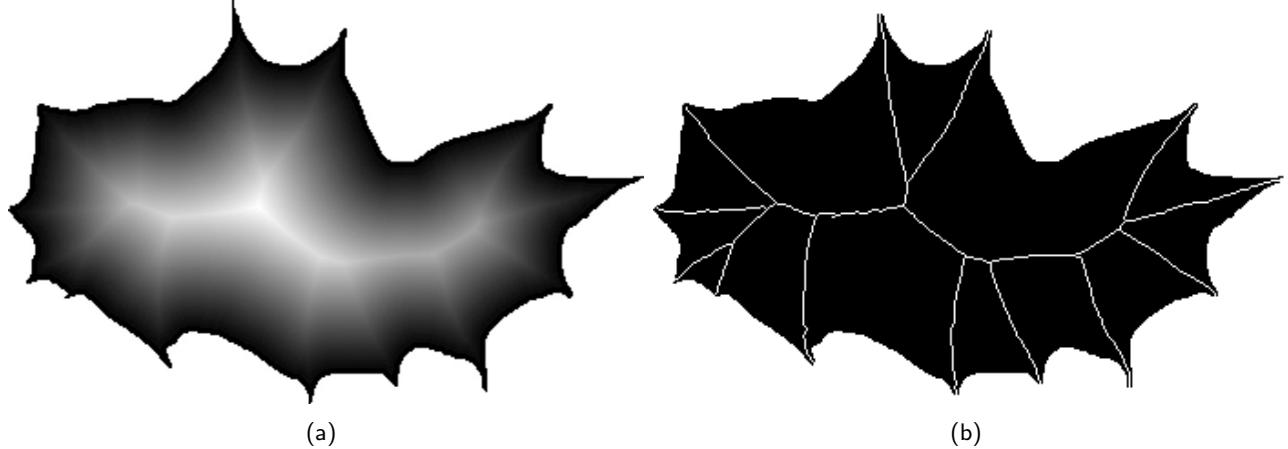


Figure 6.3: (a) Euclidean Distance Map of a leaf shape, (b) Skeleton of this leaf shape.

The signed distance satisfies condition

$$\|\nabla D\| = 1, \quad (6.9)$$

except at skeleton points.

6.1.4 Computing the skeleton

Let us now switch to a discrete domain $\hat{\mathcal{D}} = \mathcal{D} \cap \mathbb{Z}^2$. We are given a binary input image containing the discrete shape $\hat{\Omega} \subset \hat{\mathcal{D}}$. The discrete shape can be intuitively understood as

$$\hat{\Omega} = \Omega \cap \mathbb{Z}^2.$$

The Euclidean distance map D in $\hat{\Omega}$ is computed thanks to the steerable algorithm of [Coeurjolly and Montanvert, 2007, Meijster et al., 2000], which operates in two linear passes in each dimension. Overall, the complexity is $O(|\hat{\Omega}|)$. An example of Euclidean distance transform of a tree leaf is presented in Fig. 6.3(a).

The 8-connected neighborhood of a point $x \in \hat{\mathcal{D}}$ is

$$\mathcal{N}_8(x) = \{y \in \hat{\mathcal{D}} \mid \|y - x\|_\infty = 1\}$$

For every point x in the outer 8-connected border of $\hat{\Omega}$, i.e. the set of background pixels with at least one 8-connected neighbor in $\hat{\Omega}$, D is set to 0. Eventually, D is extended below 0 in the background, according to Eq. (6.8), within a radius of $r + 1$ for the AOF-based skeleton, and $3\sigma + 1$ for the ridgeness-based skeleton. We thus obtain a truncated signed distance function, which is smooth on the object contour, avoiding border artifacts on the AOF and ridgeness maps.

In Algorithm 4, line 10, boundary $\partial\hat{\Omega}$ is discretized as the 8-connected inner border, i.e. the subset of pixels in $\hat{\Omega}$ having at least one 8-connected neighbor in the background.

The skeletonization procedure, described in Algorithm 4, performs flux-ordered thinning, using a max-heap [Cormen et al., 2009, Chapter 6], relying on a criterion based on simple points [Bertrand and Malandain, 1994], so that thinness and homotopy to the input shape are maintained. Endpoints such that $aof(x, r) < th_{aof}$ are automatically kept as skeleton points. As soon as a point p has been processed, propagation is performed on its 8-connected neighborhood. The procedure generates a discrete 8-connected skeleton $\hat{S} \subset \hat{\Omega}$.

This algorithm theoretically operates in $O(|\hat{\Omega}| \log |\hat{\Omega}|)$ iterations - due to the fact that thinning is ordered with respect to the AOF measure - but is close to $O(|\hat{\Omega}|)$ in practice. Skeletons generated with this procedure

Algorithm 4: AOF-ordered max heap-based thinning [Siddiqi et al., 2002]

Input:
 $\hat{\Omega} \subset \hat{\mathcal{D}}$: discrete shape
 $r \in \mathbb{R}^+$: radius

Output:
 $\hat{S} \subset \hat{\Omega}$: skeleton

Variables

\mathcal{H} : max-heap sorted w.r.t aof
 $E \subset \mathbb{Z}^2$: set of skeleton endpoints

begin

$\hat{S} := \hat{\Omega}$
 $E := \emptyset$

10 **foreach** point $x \in \partial\hat{\Omega}$ **do**
 | insert(x, \mathcal{H})

end

16 **while** notEmpty(\mathcal{H}) **do**
 | $x := \text{extractTopElement}(\mathcal{H})$
 | **if** isSimple(x) **then**
 | | **if** isEndpoint(x) and aof(x, r) $\leq \text{th}_{\text{aof}}$ **then**
 | | | Add x to E
 | | **else**
 | | | Remove x from \hat{S}
 | | | **foreach** neighbor $y \in \mathcal{N}_8(x) \cap \hat{S}$ **do**
 | | | | **if** isSimple(y) and $y \notin E$ **then**
 | | | | | insert(y, \mathcal{H})
 | | | | **end**
 | | | **end**
 | | **end**
 | **end**
 end

end

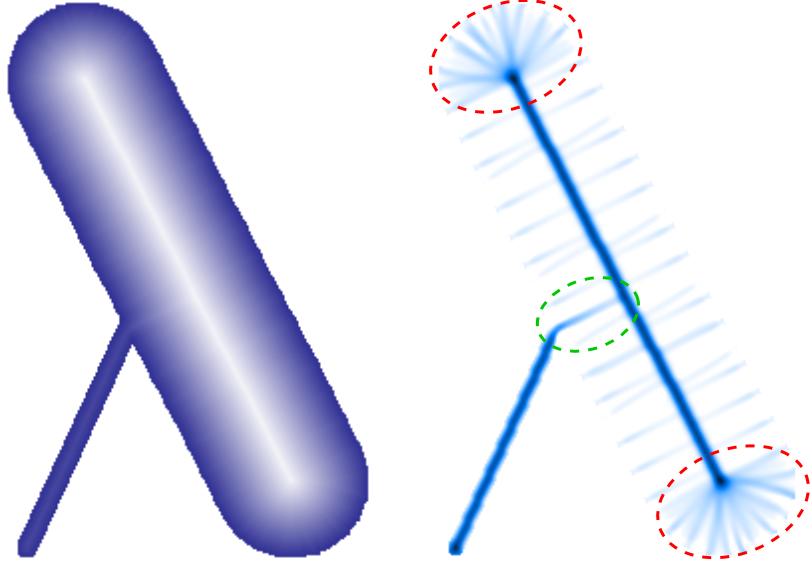


Figure 6.4: Shortcomings of ridges on ligatures and branch extremities. (Left) Distance map and (right) Ridgeness map. Ligatures create undesirably weak ridges (dashed green ellipse) whereas branch extremities create undesirably strong ridges (dashed red ellipses)

are studied in Section 6.2.7.

In our initial DECS (Digital Euclidean Connected Skeleton) method [JVCIR15], the skeleton construction uses two thresholds on the ridgeness map, namely $\text{th}_{\text{rdg-low}}$ and $\text{th}_{\text{rdg-high}}$. Hard thresholding is performed with respect to $\text{th}_{\text{rdg-low}}$, chosen slightly above 0, in order to remove all points that are unlikely to be skeleton points, in linear time. The initial purpose was to avoid the log-linear complexity of Algorithm 4. However, small branches connected by weak ligatures, described in Section 6.2.1, can be lost, as skeleton ligature points can have very weak ridgeness. This impediment drastically reduces the range of $\text{th}_{\text{rdg-low}}$ for which an accurate skeleton can be obtained. After thresholding, a thinning pass is performed. Finally, the thin skeleton is pruned with a criterion using, among others, the second threshold $\text{th}_{\text{rdg-high}}$. In [JMIV19], we argued that the ridgeness-based skeleton can be generated more simply, in the same way as the AOF-based one. Indeed, a similar homotopy-preserving iterative thinning can be applied, with a single threshold th_{rdg} . Specifically, in the ridgeness-ordered thinning procedure, \mathcal{H} is a min-heap sorted w.r.t to ridgeness, and the condition in line 16 should be replaced by

$$\text{isEndpoint}(\mathbf{p}) \text{ and } \text{rdg}(\mathbf{p}, \sigma) \geq \text{th}_{\text{rdg}} \quad (6.10)$$

The skeleton obtained is shown in Fig. 6.3(b).

6.2 Theoretical values for AOF and ridgeness

6.2.1 Why studying local shape configurations?

We now introduce our first contribution on the backdrop of the limitations of existing work. In [Siddiqi et al., 2002], no value is given for th_{aof} , which is used for marking skeleton endpoints in Algorithm 4. In [Bouix et al., 2005, pp 220], th_{aof} is selected using an empirical approach only, such that 25 – 40% of the AOF map has values below it. They report a value of -5.0 for all experiments. In [Dimitrov et al., 2003, pp 840], a threshold value is given with respect to the minimum *object angle*³ allowed for skeleton endpoints, but no explicit formula, involving the AOF parameter r , is given. Similarly,

³The notion of the *object angle* is explained in Section 6.2.2

in [JVCIR15], we only provided an empirical approach to determine threshold(s) on ridgeness. Our claim here is that the thresholds can be chosen by taking into account theoretical values on specific shape configurations, and can be expressed with respect to their respective parameters r and σ .

Theoretical values of the AOF were calculated in [Dimitrov et al., 2003, Torsello and Hancock, 2006] for an infinitesimal r , i.e. as a limit when r tends to 0. They were not calculated for a general r . Moreover, they were calculated for skeleton points only, but not at locations neighboring skeleton points. We extend the work in [Dimitrov et al., 2003] by providing invariants for any r . For some particular types of skeleton points, we perform further extension by generalizing the measure to points near skeleton points. Doing so, we formalize the variation of AOF as the considered points get farther from the skeleton. Since we establish the relation between AOF and ridgeness, we are able to provide equivalent results for the ridgeness.

In addition, we provide a model for *ligature* skeleton points, induced by connections between branches, which was not studied in [Dimitrov et al., 2003]. In general, skeleton points are located on significant ridges of D . However, there exist non-skeleton points with undesirably high - in absolute value - AOF or ridgeness (typically, points in branch extremities) and, conversely, skeleton with undesirably low AOF or ridgeness (typically, ligature points). This phenomenon is depicted in Fig. 6.4. Note that the presence of undesirably strong ridges near branch extremities is amplified by discretization artifacts. Ligature points are problematic [Katz and Pizer, 2003, Macrini et al., 2011a, Pizer et al., 2003], as they are weak ridges of D , that should nevertheless be kept as part of the skeleton.

We use a novel approach to study the various skeleton points, inspired by the classifications in [Dimitrov et al., 2003, Giblin and Kimia, 2004, Pizer et al., 2003]. We consider the following types:

- regular skeleton points
- peak points
- end points
- ligature points
- junction points

We calculate theoretical values of the AOF in Eq. (6.4), and ridgeness in Eq. (6.6) for local shape configurations corresponding to these types of skeleton points. For some configurations, integrals can be calculated explicitly, while other configurations require approximations of D so that closed-form expressions can be obtained. A useful property, that will be used subsequently, is the rotation-invariance of the AOF and ridgeness. To be more precise, $aof(\mathbf{x}, r)$ and $rdg(\mathbf{x}, \sigma)$ do not change if the shape is rotated with center \mathbf{x} . The following derivations are valid for any orientation of the skeleton branch under study. As will be derived, the AOF and ridgeness have the advantageous property of being independent of the local thickness of the shape, i.e. of the absolute value of $D(\mathbf{x})$. They rather depend on the local geometry of the shape borders. To begin with, we establish the link between the two measures.

Proposition 1. *The AOF and ridgeness at point \mathbf{x} are related as follows:*

$$rdg(\mathbf{x}, \sigma) = -\frac{1}{\sigma^4} \int_0^\infty \rho^2 \exp\left(-\frac{\rho^2}{2\sigma^2}\right) aof(\mathbf{x}, \rho) d\rho \quad (6.11)$$

Proof. The proof is given in Appendix C.1.1. □

In what follows, we omit the second parameter for AOF and ridgeness. Thus it is assumed that

$$\begin{aligned} aof(\mathbf{x}) &= aof(\mathbf{x}, r) \\ rdg(\mathbf{x}) &= rdg(\mathbf{x}, \sigma) \end{aligned}$$

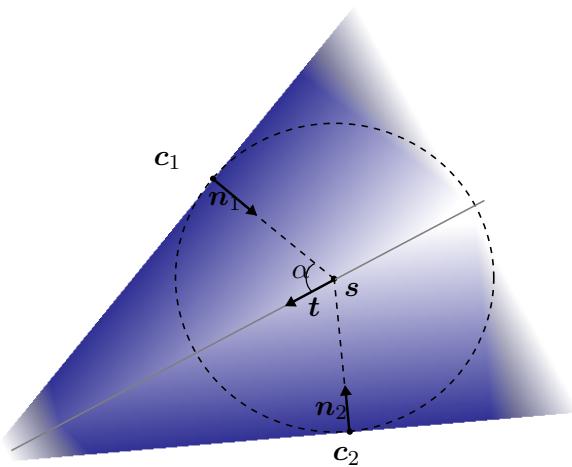


Figure 6.5: Regular skeleton point.

6.2.2 Regular skeleton point

Let $c : [0, 1] \rightarrow \mathcal{D}$ be a parametrization of the shape contour $\partial\Omega$. The parameterization is continuously differentiable and positively oriented, so that the unit tangent and inward normal are defined everywhere (see Eqs (4.1) and (4.2), respectively). Consider a skeleton point s , as the center of a maximal disk tangent to the contour at two points c_1 and c_2 . We denote by n_1 and n_2 the unit normal vectors at c_1 and c_2 , respectively. This regular skeleton point configuration is illustrated in Fig. 6.5.

The distance between a point x and a line with origin p and unit direction vector v is $(x - p) \cdot v^\perp$. Thus, if we locally approximate parts of the contour around c_1 and c_2 with straight lines, distance D in the neighborhood of s is

$$D(x) = D(s) + \min((x - s) \cdot n_1, (x - s) \cdot n_2) \quad (6.12)$$

The unit direction of the skeleton branch is

$$t = \frac{n_1^\perp - n_2^\perp}{\|n_1 - n_2\|}$$

Let α be the *object angle*, as introduced in [Dimitrov et al., 2003], which is half the angle formed by the two inward unit normal vectors. Note that $n_1 \cdot n_2 = \cos(2\alpha)$ and $\|n_1 - n_2\| = 2 \sin \alpha$. When the two contour parts are parallel, the object angle is $\frac{\pi}{2}$. For every point x located on the skeleton, i.e. $\exists k$ s.t. $x = s + kt$, D is not differentiable. However, we can still derive the expressions of the gradient and Laplacian of D , using identity $\min(x, y) = \frac{x + y - |x - y|}{2}$

$$\begin{aligned} \nabla D(x) &= \frac{1}{2} (n_1 + n_2 - \text{sgn}((x - s) \cdot (n_1 - n_2))(n_1 - n_2)) \\ \Delta D(x) &= -\delta((x - s) \cdot (n_1 - n_2)) \|n_1 - n_2\|^2 \end{aligned} \quad (6.13)$$

where δ is the Dirac distribution, implying that the gradient and Laplacian should be understood in the sense of distributions (weak derivatives).

As will be derived, the AOF and ridgeness at skeleton point s depend on object angle α . In what follows, we calculate the AOF and ridgeness measures for any point x in the vicinity of s . As we will see, in absolute value, both are decreasing functions of $|(\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp|$, the distance between x and the nearest point on the skeleton.

Average outward flux

Proposition 2. *The AOF at a point x in the neighborhood of a regular skeleton point s , with object angle α , is*

$$\text{aof}_{\text{regular}}(\mathbf{x}) = \begin{cases} -\frac{2 \sin \alpha}{\pi r} \sqrt{r^2 - ((\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp)^2} & \text{if } |(\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp| < r \\ 0 & \text{otherwise} \end{cases} \quad (6.14)$$

Proof. The proof is given in Appendix C.1.2. \square

As a geometric interpretation, notice that $2\sqrt{r^2 - ((\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp)^2}$ is the length of the line segment resulting from the intersection of the disk and the skeleton branch. As a particular case, when the point is the skeleton point s ,

$$\text{aof}_{\text{regular}}(\mathbf{s}) = -\frac{2}{\pi} \sin \alpha \quad (6.15)$$

as found in [Dimitrov et al., 2003]. A notable property is that the AOF at a regular skeleton point is independent of r . Moreover, it is independent of the position s , and thus constant along the skeleton branch. The most salient regular skeleton point is obtained when the two contour parts are parallel, i.e. $\alpha = \frac{\pi}{2}$, which gives $\text{aof}_{\text{regular}}(\mathbf{s}) = -\frac{2}{\pi}$.

Ridgeness

Proposition 3. *The ridgeness at a point x in the neighborhood of a regular skeleton point s , with object angle α , is*

$$\text{rdg}_{\text{regular}}(\mathbf{x}) = \frac{\sqrt{2\pi} \sin \alpha}{\pi \sigma} \exp\left(-\frac{((\mathbf{x} - \mathbf{s}) \cdot \mathbf{t}^\perp)^2}{2\sigma^2}\right) \quad (6.16)$$

Proof. The proof is given in Appendix C.1.3. \square

It is easy to see from Eq. (6.16) that the ridgeness decreases with a Gaussian profile as x gets farther from the skeleton branch. As a particular case, when the point is the skeleton point s ,

$$\text{rdg}_{\text{regular}}(\mathbf{s}) = \frac{\sqrt{2\pi}}{\pi \sigma} \sin \alpha. \quad (6.17)$$

As the AOF, the ridgeness is independent of position s and is thus constant along the skeleton branch. The highest ridgeness value appears when the two contour parts are parallel, which leads to $\text{rdg}_{\text{regular}}(\mathbf{s}) = \frac{\sqrt{2\pi}}{\pi \sigma}$.

6.2.3 Peak point

If Ω is a disk, there exist only one skeleton point s at its center, which is a local maximum of D . The distance function is then

$$D(\mathbf{x}) = D(\mathbf{s}) - \|\mathbf{s} - \mathbf{x}\| \quad (6.18)$$

which is non-differentiable at s . Otherwise, for any $\mathbf{x} \neq \mathbf{s}$,

$$\begin{aligned} \nabla D(\mathbf{x}) &= \frac{\mathbf{s} - \mathbf{x}}{\|\mathbf{s} - \mathbf{x}\|} \\ \Delta D(\mathbf{x}) &= -\frac{1}{\|\mathbf{s} - \mathbf{x}\|} \end{aligned} \quad (6.19)$$

This case is depicted in Fig. 6.6(a). It is of little practical use in itself, as the shape to be skeletonized is rarely a disk. However, in Section 6.2.4, we derive the more general endpoint case from the current case. For calculating both AOF and ridgeness at x , we switch to a polar coordinate system, centered at x s.t. $\mathbf{s} = \mathbf{x} + [\mathcal{R} \cos \beta, \mathcal{R} \sin \beta]^\top$, and show that aof and rdg are decreasing functions (in absolute value) of distance $\mathcal{R} = \|\mathbf{x} - \mathbf{s}\|$.

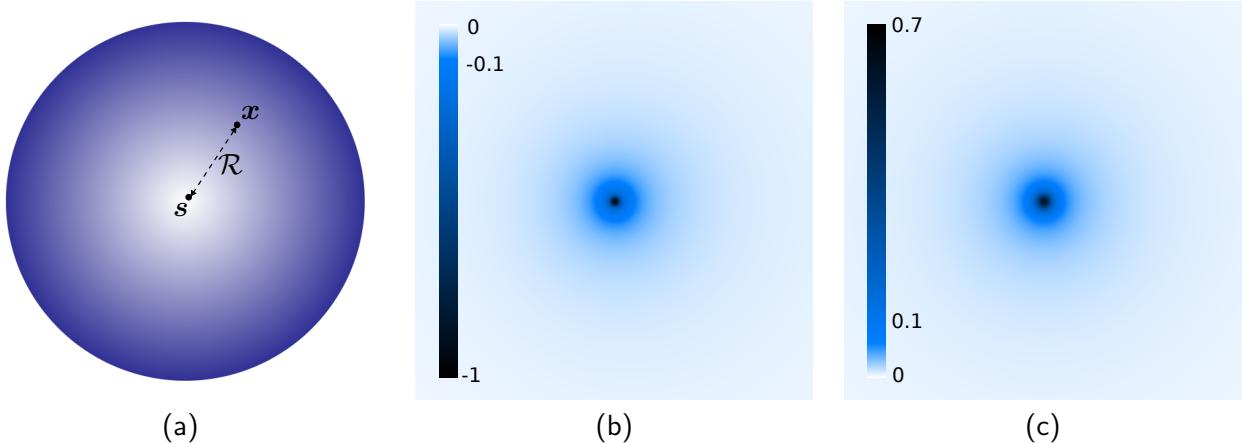


Figure 6.6: Peak point. (a) Distance (b) Average outward flux with $r = 2$ (c) Ridgeness with $\sigma = 2$

Preliminary notes on elliptic integrals

We define special functions that arise when deriving the AOF and ridgeness of points in the neighborhood of a peak point. Given an argument $\psi \in [0, \frac{\pi}{2}]$ and a modulus $k \in [0, 1]$, $F(\psi, k)$ and $E(\psi, k)$ are Legendre's *incomplete elliptic integrals* of the first and second kind [Carlson, 2010, p. 486], respectively, defined as

$$\begin{aligned} F(\psi, k) &= \int_0^\psi \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta, \\ E(\psi, k) &= \int_0^\psi \sqrt{1 - k^2 \sin^2 \theta} d\theta. \end{aligned} \quad (6.20)$$

A particular case arises when $\psi = \frac{\pi}{2}$, which leads to the so-called *complete elliptic integrals* of the first and second kind, respectively,

$$\begin{aligned} K(k) &= F\left(\frac{\pi}{2}, k\right), \\ E(k) &= E\left(\frac{\pi}{2}, k\right). \end{aligned} \quad (6.21)$$

These integrals have no closed-form expressions. They can be numerically evaluated using Landen's transformation, related to the arithmetic-geometric mean [Carlson, 2010, p. 493]. Moreover, closed-form approximations and bounds for them have been extensively studied [Alzer and Qiu, 2004, Anderson et al., 1992, Carlson, 1966, Neuman, 2011].

Average outward flux

Using polar coordinates centered at x and Eqs. (6.4) and (6.19), we obtain the following result:

Proposition 4. *The AOF at a point x , at distance $\mathcal{R} = \|x - s\|$ from a peak skeleton point s is*

$$\text{aof}_{\text{peak}}(x) = \frac{1}{\pi} \int_0^\pi \frac{-r + \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta. \quad (6.22)$$

Proposition 5. *The AOF at a point x , at distance $\mathcal{R} = \|x - s\|$ from a peak skeleton point s can be expressed using complete elliptic integrals as*

$$\text{aof}_{\text{peak}}(x) = \frac{1}{\pi r} ((\mathcal{R} - r)K(k) - (\mathcal{R} + r)E(k)) \quad (6.23)$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$.

Proof. The proofs for the two previous propositions are given in Appendix C.1.5. \square

Let M be the generalized mean of two real numbers a and b , also known as the power mean,

$$M_p(a, b) = \begin{cases} \left(\frac{a^p + b^p}{2}\right)^{\frac{1}{p}} & \text{if } p \neq 0 \\ \sqrt{ab} & \text{if } p = 0. \end{cases}$$

Special cases include the geometric mean ($p = 0$) and the arithmetic mean ($p = 1$). According to [Carlson, 1966, Vamanamurthy and Vuorinen, 1994], the following inequality holds,

$$M_0(a, b) < M_1(a, b) < M_p(a, b) \text{ with } p > 1. \quad (6.24)$$

It can be noticed that k is the ratio between the geometric and arithmetic means of r and \mathcal{R} , which verifies, according to Eq. (6.24),

$$\frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r} \leq 1$$

It appears that the second-order Taylor expansion of D gives a suitable approximation to aof_{peak} as soon as \mathcal{R} is large enough. Let \tilde{D} be the second-order Taylor approximation of D in the neighborhood of \mathbf{x} :

$$\begin{aligned} \tilde{D}(\mathbf{y}) &= D(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^\top \nabla D(\mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathbf{x})^\top \nabla^2 D(\mathbf{x})(\mathbf{y} - \mathbf{x}) \\ D(\mathbf{y}) &= \tilde{D}(\mathbf{y}) + O(\|\mathbf{y} - \mathbf{x}\|^3) \end{aligned} \quad (6.25)$$

The approximate AOF is

$$\widetilde{\text{aof}}(\mathbf{x}) = \frac{1}{2\pi} \int_0^{2\pi} \nabla \tilde{D} \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \quad (6.26)$$

Proposition 6. *The second-order approximation of the AOF at a point \mathbf{x} , at distance $\mathcal{R} = \|\mathbf{x} - s\|$ from a peak skeleton point s is*

$$\widetilde{\text{aof}}_{\text{peak}}(\mathbf{x}) = -\frac{r}{2\mathcal{R}}$$

Proposition 7. *The AOF at \mathbf{x} is asymptotically equivalent to its second-order approximation*

$$\text{aof}_{\text{peak}}(\mathbf{x}) \sim \widetilde{\text{aof}}_{\text{peak}}(\mathbf{x}) \text{ (as } \|\mathbf{x} - s\| \rightarrow +\infty\text{)}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.6 \square

We observe from Fig. 6.7 that this approximation is accurate as soon as $\mathcal{R} \gg r$.

Ridgeness

We rewrite Eq. (6.7) in the polar-coordinate system centered at \mathbf{x} ,

$$\text{rdg}(\mathbf{x}) = - \int_0^\infty \int_0^{2\pi} \rho G_\sigma(\rho) \Delta D \left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta d\rho, \quad (6.27)$$

where $G_\sigma(\rho)$ is a shorthand notation for

$$G_\sigma(\rho \cos \theta, \rho \sin \theta) = \frac{1}{2\pi\sigma^2} \exp \left(-\frac{\rho^2}{2\sigma^2} \right).$$

Combining Eqs. (6.27) and (6.19), we obtain the following result:

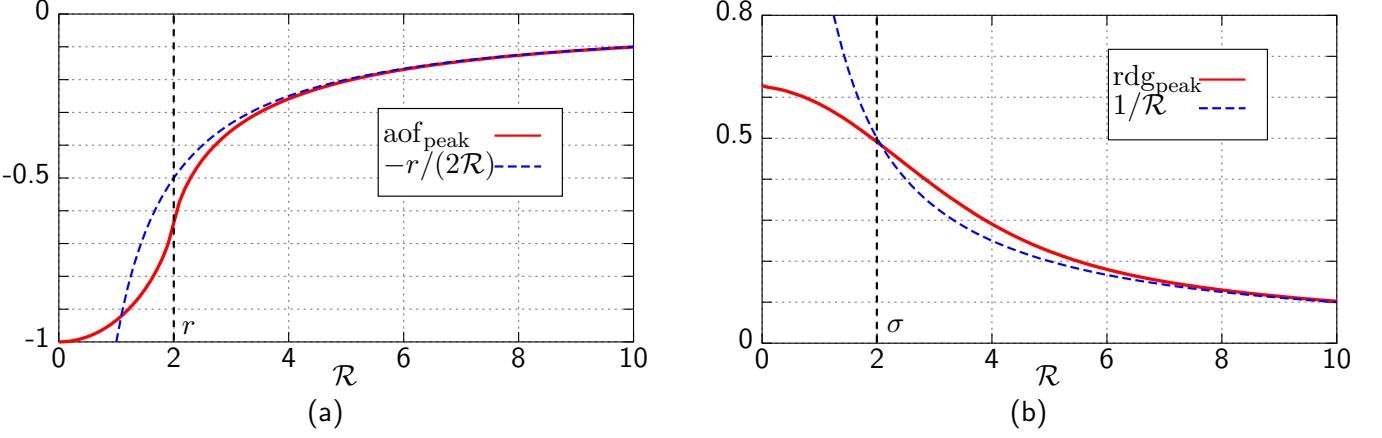


Figure 6.7: (a) Average outward flux and corresponding bounds at distance \mathcal{R} from a peak skeleton point, versus \mathcal{R} (with $r = 2$). (b) Ridgeness and corresponding bounds at distance \mathcal{R} from a peak skeleton point, versus \mathcal{R} (with $\sigma = 2$)

Proposition 8. *The ridgeness at a point x , at distance $\mathcal{R} = \|x - s\|$ from a peak skeleton point s is*

$$\text{rdg}_{\text{peak}}(x) = 2 \int_0^\infty \rho G_\sigma(\rho) \int_0^\pi \frac{1}{\sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R}\cos\theta}} d\theta d\rho. \quad (6.28)$$

Proposition 9. *The ridgeness at a point x , at distance $\mathcal{R} = \|x - s\|$ from a peak skeleton point s can be expressed using the complete elliptic integral of the first kind as*

$$\text{rdg}_{\text{peak}}(x) = \frac{2}{\pi\sigma^2} \int_0^\infty \frac{\rho}{\mathcal{R} + \rho} K\left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}\right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right) d\rho. \quad (6.29)$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.7. \square

No closed-form expression can be found for Eq. (6.29). In Fig. 6.7, numerical integration was performed to plot rdg_{peak} versus \mathcal{R} . As for the AOF, the second-order Taylor expansion of \tilde{D} gives a suitable approximation to rdg_{peak} as soon as \mathcal{R} is large enough. The approximate ridgeness is obtained using Eqs (6.25) and the polar transformation of Eq. (6.6):

$$\widetilde{\text{rdg}}(x) = - \int_0^\infty \int_0^{2\pi} \rho \Delta G_\sigma(\rho) \tilde{D}\left(x + \begin{bmatrix} \rho \cos\theta \\ \rho \sin\theta \end{bmatrix}\right) d\theta d\rho, \quad (6.30)$$

where $\Delta G_\sigma(\rho)$ is a shorthand notation for

$$\Delta G_\sigma(\rho \cos\theta, \rho \sin\theta) = \frac{1}{\pi\sigma^4} \left(\frac{\rho^2}{2\sigma^2} - 1 \right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right),$$

Proposition 10. *The second-order approximation of the ridgeness at a point x , at distance $\mathcal{R} = \|x - s\|$ from a peak skeleton point s is*

$$\widetilde{\text{rdg}}_{\text{peak}}(x) = \frac{1}{\mathcal{R}}.$$

Proposition 11. *The ridgeness at x is asymptotically equivalent to its second-order approximation*

$$\text{rdg}_{\text{peak}}(x) \sim \widetilde{\text{rdg}}_{\text{peak}}(x) \text{ (as } \|x - s\| \rightarrow +\infty\text{)}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.8. \square

As for the AOF, we observe from Fig. 6.7 that this approximation is accurate as soon as $\mathcal{R} \gg \sigma$.

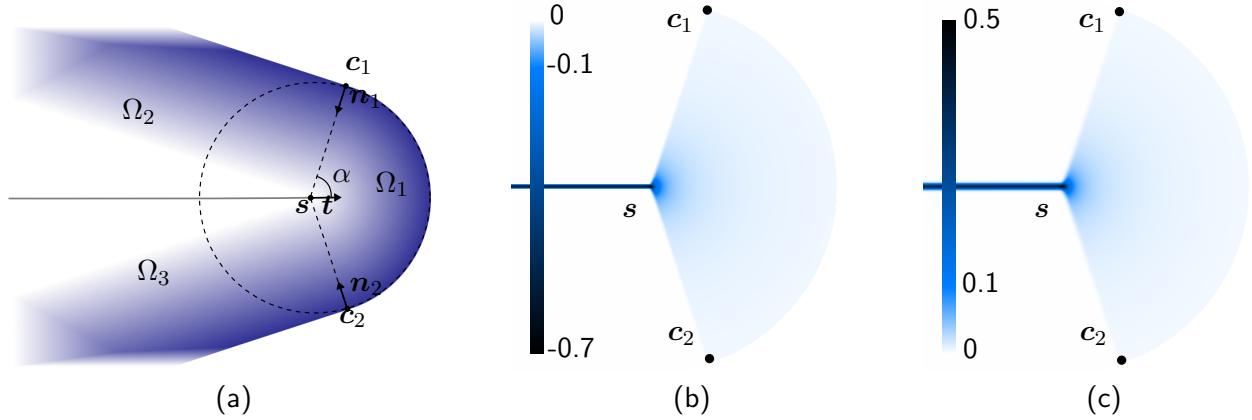


Figure 6.8: Endpoint. (a) Distance (b) Average outward flux with $r = 2$ (c) Ridgeness with $\sigma = 2$

6.2.4 Endpoint

The endpoint configuration, illustrated in Fig. 6.8(a), is described as a mix of properties of the regular skeleton point in Section 6.2.2 and the peak point in Section 6.2.3. As in the case of the regular skeleton point, the skeleton branch forms an object angle α , which is half the angle formed by the two inward unit normal vectors \mathbf{n}_1 and \mathbf{n}_2 . The skeleton branch has unit tangent vector

$$\mathbf{t} = \frac{\mathbf{n}_1^\perp - \mathbf{n}_2^\perp}{\|\mathbf{n}_1 - \mathbf{n}_2\|}.$$

We assume that the branch extremity forms an arc of angle 2α with center s . In what follows, we focus on deriving the AOF and ridgeness at s , as functions of α . The shape branch is split into 3 open subregions. Ω_1 is the region bounded by line segments sc_1 , sc_2 and the arc from c_1 to c_2 . Ω_2 is the region above the skeleton branch and on the left of line sc_1 , whereas Ω_3 is the region below the skeleton branch and on the left of the line sc_2 . In the current case, we consider a piecewise definition of the distance,

$$D(\mathbf{x}) = \begin{cases} D(s) - \|\mathbf{x} - s\| & \text{if } \mathbf{x} \in \Omega_1 \\ D(s) + (\mathbf{x} - s) \cdot \mathbf{n}_1 & \text{if } \mathbf{x} \in \Omega_2 \\ D(s) + (\mathbf{x} - s) \cdot \mathbf{n}_2 & \text{if } \mathbf{x} \in \Omega_3, \end{cases} \quad (6.31)$$

and its resulting gradient, which is undefined on the common boundaries of Ω_1 , Ω_2 and Ω_3 ,

$$\nabla D(\mathbf{x}) = \begin{cases} \frac{s - \mathbf{x}}{\|s - \mathbf{x}\|} & \text{if } \mathbf{x} \in \Omega_1 \\ \mathbf{n}_1 & \text{if } \mathbf{x} \in \Omega_2 \\ \mathbf{n}_2 & \text{if } \mathbf{x} \in \Omega_3. \end{cases} \quad (6.32)$$

Average outward flux

Proposition 12. *The AOF at the endpoint s of a skeleton branch, with object angle α , is⁴*

$$\text{aof}_{\text{end}}(s) = -\frac{1}{\pi}(\alpha + \sin \alpha). \quad (6.33)$$

Proof. The proof is given in Appendix C.1.9. □

For any point $\mathbf{x} \in \Omega_1$ s.t. the disk of radius r and center \mathbf{x} is fully included within Ω_1 , $\text{aof}_{\text{end}}(\mathbf{x}) = \text{aof}_{\text{peak}}(\mathbf{x})$. It can be observed from Fig. 6.8(b) that the AOF in Ω_1 has a behavior similar to the one obtained for the peak point case. Its absolute value decreases in $O(1/\|\mathbf{x} - s\|)$.

⁴A similar result was already stated in [Dimitrov et al., 2003]

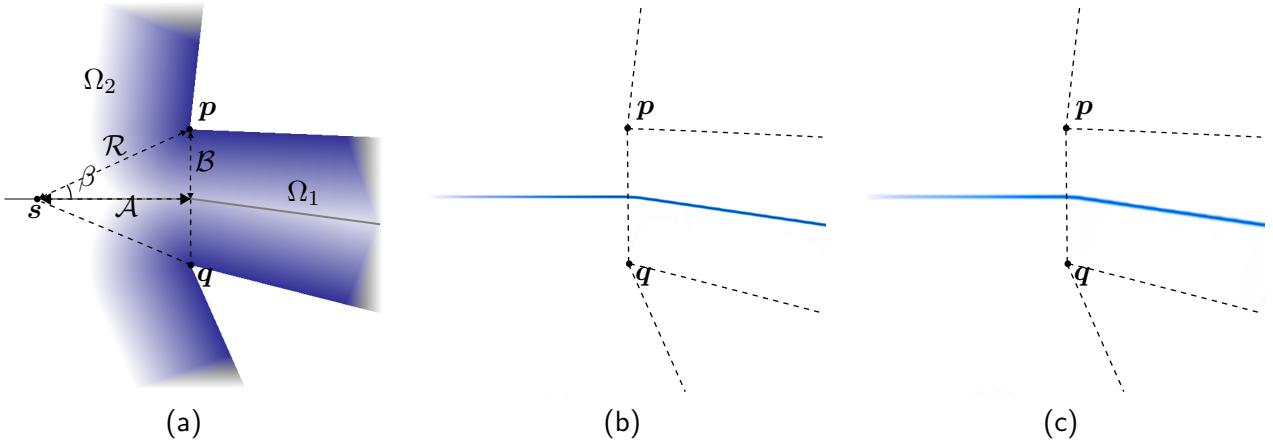


Figure 6.9: Ligature point. (a) Distance (b) Average outward flux with $r = 2$ (c) Ridgeness with $\sigma = 2$ (color scales for the AOF and ridgeness are similar to the ones in Fig. 6.8)

Ridgeness

Proposition 13. *The ridgeness at the endpoint s of a skeleton branch, with object angle α , is*

$$\text{rdg}_{\text{end}}(s) = \frac{\sqrt{2\pi}}{2\pi\sigma}(\alpha + \sin \alpha) \quad (6.34)$$

Proof. The proof is given in Appendix C.1.10. \square

Again, it can be observed from Fig. 6.8(c) that the ridgeness in Ω_1 has a behavior similar to the one obtained for the peak point case.

6.2.5 Ligature point

In Fig. 6.9(a), a thin branch connects to a thick branch, which creates a ligature. As partially described in Section 6.1.4, a *ligature* is a skeleton branch created by the junction of two shape branches. Unlike a regular skeleton branch, it does not arise from a shape branch itself. The junction creates two corners p and q . Let ℓ be the line passing through p and q . Let us denote the two branches by Ω_1 and Ω_2 , on the right and left of ℓ , respectively. The ligature is included into Ω_2 and is located on the bisector of p and q , regardless of the orientation of branch Ω_1 . Its unit tangent vector is

$$\mathbf{t} = \frac{(\mathbf{p} - \mathbf{q})^\perp}{\|\mathbf{p} - \mathbf{q}\|}.$$

Assuming that \mathbf{t} is directed towards Ω_2 , any ligature point verifies

$$\mathbf{s} = \frac{\mathbf{p} + \mathbf{q}}{2} + \mathcal{A}\mathbf{t}$$

with $\mathcal{A} \geq 0$. Midpoint $(\mathbf{p} + \mathbf{q})/2$ is referred to as the *ligature junction*. Let \mathcal{B} be the half-thickness of branch Ω_1 at the junction,

$$\mathcal{B} = \frac{\|\mathbf{p} - \mathbf{q}\|}{2}.$$

Let us assume that the thickness of Ω_2 is much greater than \mathcal{B} . For any point in the neighborhood of a ligature point s ,

$$D(\mathbf{x}) = \min(\|\mathbf{x} - \mathbf{p}\|, \|\mathbf{x} - \mathbf{q}\|), \quad (6.35)$$

and, for the ligature point itself,

$$D(\mathbf{s}) = \|\mathbf{s} - \mathbf{p}\| = \|\mathbf{s} - \mathbf{q}\| = \sqrt{\mathcal{A}^2 + \mathcal{B}^2}.$$

We calculate the AOF and ridgeness of a ligature point \mathbf{s} , assuming that distance \mathcal{A} is reasonably small compared to the thickness of Ω_2 . In other words, \mathbf{s} is far enough from the opposite border of Ω_2 , so that \mathbf{p} and \mathbf{q} are considered as the only local borders. As will be derived, AOF and ridgeness are decreasing functions, in absolute value, of distance \mathcal{A} . Hence, in what follows, the distance to borders \mathcal{R} , and angle β both depend on \mathcal{A} :

$$\begin{aligned} \mathcal{R}(\mathcal{A}) &= \sqrt{\mathcal{A}^2 + \mathcal{B}^2} \\ \beta(\mathcal{A}) &= \tan^{-1} \frac{\mathcal{B}}{\mathcal{A}} \end{aligned} \quad (6.36)$$

Average outward flux

As an additional requirement, the following AOF is valid only if $\mathcal{A} \geq r$ and $\mathcal{B} \geq r$.

Proposition 14. *The AOF at ligature point \mathbf{s} , at a distance \mathcal{A} from the ligature junction, is*

$$\text{aof}_{\text{ligature}}(\mathbf{s}) = \frac{1}{\pi} \int_0^\pi \frac{r - \mathcal{A} \cos \theta - \mathcal{B} \sin \theta}{\sqrt{r^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2r(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)}} d\theta. \quad (6.37)$$

Proposition 15. *The AOF at ligature point \mathbf{s} , at a distance \mathcal{A} from the ligature junction, can be expressed with complete and incomplete elliptic integrals as*

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{\pi r} \left[(\mathcal{R} + r) \left(2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \right. \\ &\quad \left. - (\mathcal{R} - r) \left(2K(k) - F\left(\frac{\beta}{2}, k\right) - F\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \right] \end{aligned}$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$, and $\mathcal{R} = \mathcal{R}(\mathcal{A})$ and $\beta = \beta(\mathcal{A})$.

Using the Taylor expansion \tilde{D} of Eq. (6.25), we can calculate an approximate AOF.

Proposition 16. *The second-order approximation of the AOF at ligature point \mathbf{s} , at a distance \mathcal{A} from the ligature junction is*

$$\widetilde{\text{aof}}_{\text{ligature}}(\mathbf{s}) = \frac{1}{\sqrt{\mathcal{A}^2 + \mathcal{B}^2}} \left(\frac{r}{2} - \frac{2\mathcal{B}}{\pi} \right).$$

Proposition 17. *The AOF at ligature point \mathbf{s} is asymptotically equivalent to its second-order approximation*

$$\text{aof}_{\text{ligature}}(\mathbf{s}) \sim \widetilde{\text{aof}}_{\text{ligature}}(\mathbf{s}) \text{ (as } \mathcal{A} \rightarrow +\infty\text{).}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.12. □

Ridgeness

Since the ridgeness is calculated on an infinite domain, it should be assumed that the following expressions are accurate if \mathcal{A} and the thickness of Ω_2 are large enough, so that $\mathcal{A} > n\sigma$ (usually, $n = 3$). Using a transformation of definition (6.6) in the polar-coordinate system centered at \mathbf{x} ,

$$\text{rdg}(\mathbf{x}) = - \int_0^\infty \int_0^{2\pi} \rho \Delta G_\sigma(\rho) D \left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta d\rho, \quad (6.38)$$

we obtain the following result:

Proposition 18. *The ridgeness at ligature point s , at a distance \mathcal{A} from the ligature junction, is*

$$\text{rdg}_{\text{ligature}}(s) = -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^\pi \sqrt{\rho^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2\rho(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)} \, d\theta d\rho. \quad (6.39)$$

Proposition 19. *The ridgeness at ligature point s , at a distance \mathcal{A} from the ligature junction can be expressed using complete and incomplete elliptic integrals of the second kind as*

$$\begin{aligned} \text{rdg}_{\text{ligature}}(s) &= \frac{4}{\pi \sigma^4} \int_0^\infty \left(1 - \frac{\rho^2}{2\sigma^2}\right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \\ &\quad (\mathcal{R} + \rho) \left(2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right)\right) d\rho, \end{aligned} \quad (6.40)$$

with $k = \frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}$ and $\mathcal{R} = \mathcal{R}(\mathcal{A})$ and $\beta = \beta(\mathcal{A})$, as defined in Eq. (6.36).

Proof. The proofs for the two previous propositions are given in Appendix C.1.13. \square

Using the Taylor expansion \tilde{D} of Eq. (6.25), we can calculate an approximate ridgeness.

Proposition 20. *The second-order approximation of the ridgeness at ligature point s , at a distance \mathcal{A} from the ligature junction is*

$$\widetilde{\text{rdg}}_{\text{ligature}}(s) = \frac{1}{\sqrt{\mathcal{A}^2 + \mathcal{B}^2}} \left(\mathcal{B} \frac{\sqrt{2\pi}}{\sigma\pi} - 1 \right)$$

Proposition 21. *The ridgeness at ligature point s is asymptotically equivalent to its second-order approximation*

$$\text{rdg}_{\text{ligature}}(s) \sim \widetilde{\text{rdg}}_{\text{ligature}}(s) \text{ (as } \mathcal{A} \rightarrow +\infty\text{).}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.14. \square

The AOF and ridgeness are illustrated in Figs. 6.9b and 6.9c, respectively. Note that their color scales are the same as for Figs. 6.8b and Fig. 6.8b. In accordance with Propositions 16 and 20, it can be observed that the AOF and ridgeness become weaker as the considered ligature point gets farther from the ligature junction.

6.2.6 Junction point

We consider a simplified model of a junction of n branches, such that the n corners $\{\mathbf{p}_i\}_{i=1\dots n}$ formed by the branches are all equidistant to the junction skeleton point s , as depicted in Fig. 6.10. We focus on the AOF and ridgeness at junction point s . Note that any point located on a line segment between s and the midpoint of two successive corners, $(\mathbf{p}_i + \mathbf{p}_{i+1})/2$, is a ligature point. The distance from s to any corner is denoted by \mathcal{R} . In the neighborhood of s , the distance is:

$$\begin{aligned} D(\mathbf{x}) &= \min_{i=1\dots n} D_i(\mathbf{x}) \\ D_i(\mathbf{x}) &= \|\mathbf{x} - \mathbf{p}_i\|. \end{aligned} \quad (6.41)$$

Switching to polar coordinates, corners are defined as

$$\mathbf{p}_i = s + [\mathcal{R} \cos \beta_i, \mathcal{R} \sin \beta_i]^\top \quad (6.42)$$

where β_i is the absolute angle formed by the line from s to \mathbf{p}_i and the horizontal axis. We denote by α_i the relative angle formed by s and the two successive corners \mathbf{p}_i and \mathbf{p}_{i+1} , thus

$$\alpha_i = \beta_{i+1} - \beta_i.$$

In the subsequent parts of this section, we show that the AOF and ridgeness only depend on the spatial layout of the corners and distance \mathcal{R} - which is linked to the thicknesses of the branches - but does not depend on the geometry of the branch borders.

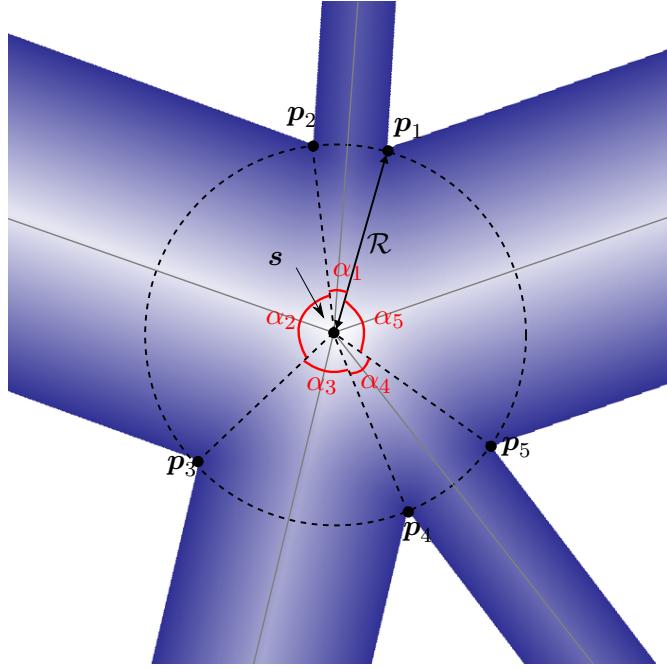


Figure 6.10: Junction point

Average outward flux

Hereafter, we assume that $\mathcal{R} > r$.

Proposition 22. *The AOF at junction point s at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1...n}$ is*

$$\text{aof}_{\text{junction}}(s) = \frac{1}{\pi} \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} \frac{r - \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta. \quad (6.43)$$

Proposition 23. *The AOF at junction point s at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1...n}$ can be expressed with complete and incomplete elliptic integrals as*

$$\begin{aligned} \text{aof}_{\text{junction}}(s) = & \frac{1}{\pi r} \sum_{i=1}^n \left[(\mathcal{R} + r) \left(E(k) - E \left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k \right) \right) \right. \\ & \left. - (\mathcal{R} - r) \left(K(k) - F \left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k \right) \right) \right] \end{aligned} \quad (6.44)$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$.

Proof. The proofs for the two previous propositions are given in Appendix C.1.15. \square

Using the Taylor expansion \tilde{D} of Eq. (6.25), we can calculate an approximate AOF.

Proposition 24. *The second-order approximation of the AOF at junction point s , at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1...n}$, is*

$$\widetilde{\text{aof}}_{\text{junction}}(s) = -\frac{1}{\pi} s_2 + \frac{r}{2\mathcal{R}} \left(1 - \frac{1}{2\pi} s_1 \right)$$

where s_1 and s_2 are the sums of angles α_i and their halves, respectively:

$$\begin{aligned} s_1 &= \sum_{i=1}^n \sin \alpha_i \\ s_2 &= \sum_{i=1}^n \sin\left(\frac{\alpha_i}{2}\right) \end{aligned} \quad (6.45)$$

This result should be put in perspective with the invariant obtained in [Dimitrov et al., 2003] for junction points. Indeed, the term $-\frac{1}{\pi}s_2$ was also found by them. We extend their result with an additional term taking r into account.

Proposition 25. *The AOF at junction point s is asymptotically equivalent to its second-order approximation*

$$\text{aof}_{\text{junction}}(s) \sim \widetilde{\text{aof}}_{\text{junction}}(s) \text{ (as } \mathcal{R} \rightarrow +\infty\text{).}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.16. \square

Ridgeness

Again, it should be assumed that the following expressions are accurate if the thickness of the junction is large enough, i.e. $\mathcal{R} > n\sigma$ (usually, $n = 3$). Starting from the polar LoG-based expression of the ridgeness of Eq (6.38), it follows that:

Proposition 26. *The ridgeness at junction point s at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1\dots n}$ is*

$$\text{rdg}_{\text{junction}}(s) = -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} \sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta} \, d\theta \, d\rho. \quad (6.46)$$

Proposition 27. *The ridgeness at junction point s at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1\dots n}$ can be expressed using complete and incomplete elliptic integrals of the second kind as*

$$\text{rdg}_{\text{junction}}(s) = \frac{4}{\pi\sigma^4} \int_0^\infty \left(1 - \frac{\rho^2}{2\sigma^2}\right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \sum_{i=1}^n (\mathcal{R} + \rho) \left(E(k) - E\left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k\right)\right) d\rho, \quad (6.47)$$

$$\text{with } k = \frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}.$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.17. \square

Proposition 28. *The second-order approximation of the ridgeness at junction point s , at distance \mathcal{R} from n corners forming angles $(\alpha_i)_{i=1\dots n}$, is*

$$\widetilde{\text{rdg}}_{\text{junction}}(s) = \frac{1}{\sigma\sqrt{2\pi}} s_2 - \frac{1}{\mathcal{R}} \left(1 - \frac{1}{2\pi} s_1\right)$$

with s_1 and s_2 as defined in Eq. (6.45).

Proposition 29. *The ridgeness at junction point s is asymptotically equivalent to its second-order approximation*

$$\text{rdg}_{\text{junction}}(s) \sim \widetilde{\text{rdg}}_{\text{junction}}(s) \text{ (as } \mathcal{R} \rightarrow +\infty\text{)}$$

Proof. The proofs for the two previous propositions are given in Appendix C.1.18. \square

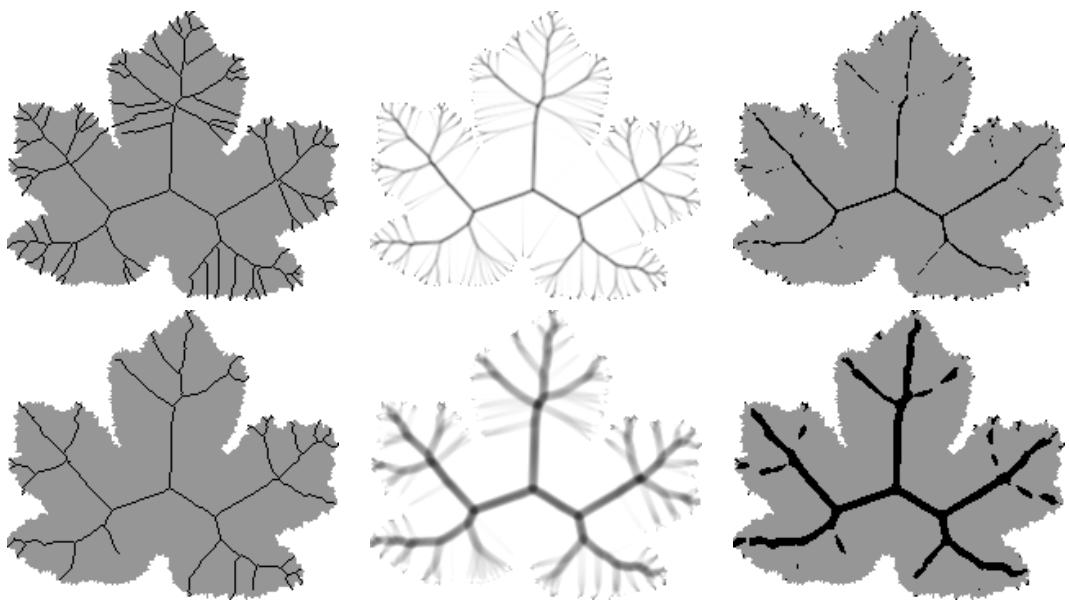


Figure 6.11: AOF-based skeleton for shape with irregular border. Left: final skeleton, Center: AOF, Right: thresholded AOF. Top row: $r = 1$, bottom row: $r = 4$

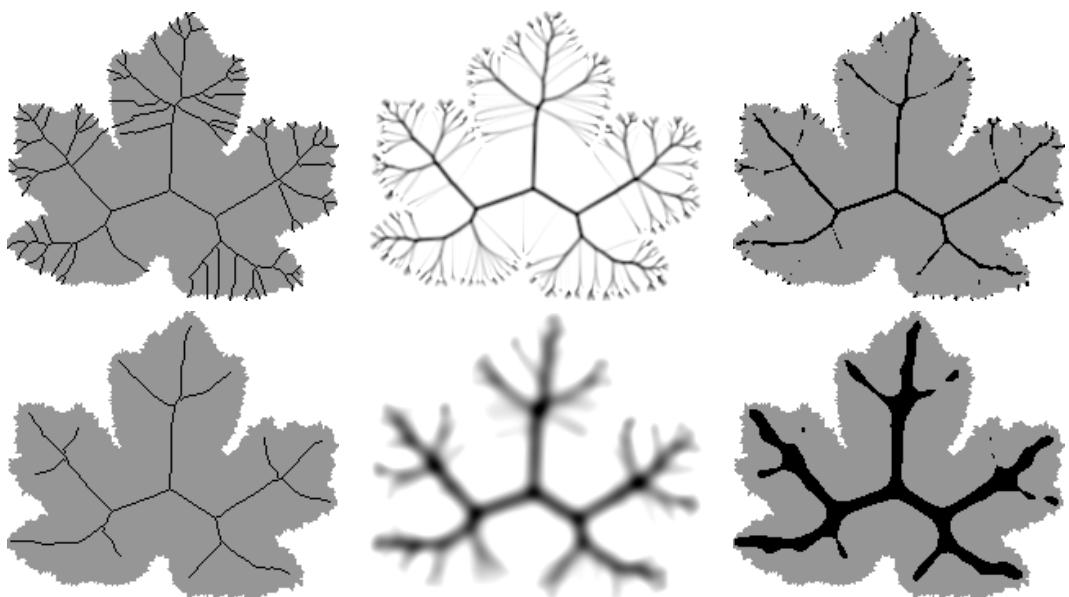


Figure 6.12: Ridgeness-based skeleton for shape with irregular border. Left: final skeleton, Center: ridgeness, Right: thresholded ridgeness. Top row: $\sigma = 1$, bottom row: $\sigma = 4$

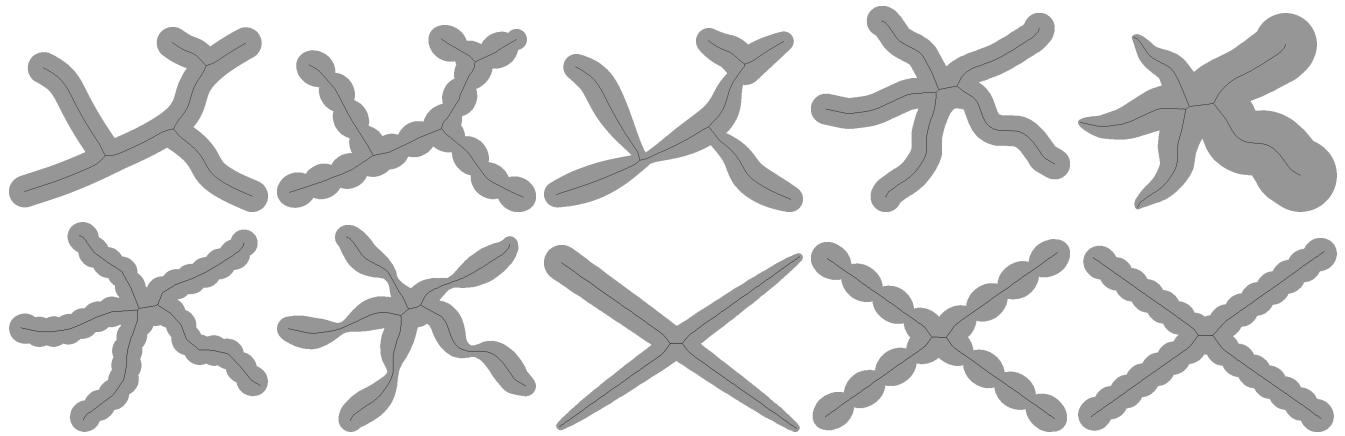


Figure 6.13: Overview of the synthetic shape dataset

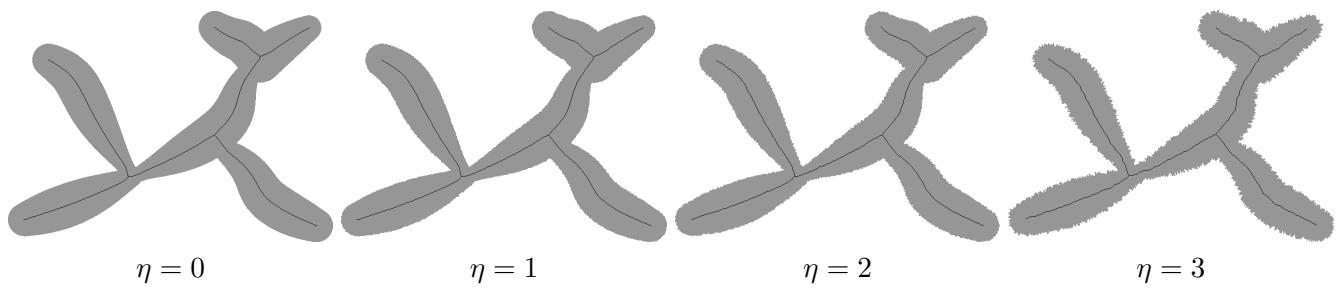


Figure 6.14: Synthetic shape n°5 at different noise levels η with AOF-based skeletons (ridgeness-based skeletons are visually equivalent and are not depicted)

6.2.7 Experiments

Influence of scale parameters: analysis on regular skeleton points

We performed several numerical experiments to corroborate our derivations and to assess the applicability of the theoretical AOF and ridgeness values. We first give a short overview of the influence of parameters r and σ , and their respective thresholds, on the final skeleton. The choice of thresholds th_{aof} and th_{rdg} in the AOF-based and ridgeness-based thinning procedure is crucial, as they control the amount of pixels that will be retained as skeleton endpoints, i.e. starting points for branches, according to lines 10 and 11 in Algorithm 4. Thresholds should be chosen as far as possible in the light of the previously derived analytical expressions.

In [Dimitrov et al., 2003, pp 840], it was suggested that th_{aof} be chosen with respect to a minimal object angle. However, no explicit formula was provided, nor was a relation established with respect to a particular theoretical shape configuration. Following their suggestion, it seems natural to derive a threshold according to a minimal object angle with respect to the regular skeleton point configuration described in Section 6.2.2, as it is the type of skeleton point most commonly encountered. Object protusions generating branches with an object angle below this minimal angle should be considered as insignificant. Choosing the threshold according to the minimal object angle gives a clear geometrical interpretation of what a significant object part is. Hence, Eqs. (6.15) and (6.17) were used as a basis,

$$\begin{aligned} th_{aof} &= -\frac{2}{\pi} \sin \alpha_0, \\ th_{rdg} &= \frac{\sqrt{2\pi}}{\pi\sigma} \sin \alpha_0, \end{aligned} \tag{6.48}$$

where α_0 is the minimal object angle that a shape part should form in order to generate a skeleton branch. Following [Dimitrov et al., 2003, pp 840], we used $\alpha_0 = \frac{\pi}{6} = 30^\circ$ in the current experiment. AOF and ridgeness-based skeletons were computed on a shape with moderate noise, such that, at a fine scale, protusions and indentations on the shape border are expected to generate branches. Results are depicted in Figs. 6.11 and 6.12. The left, center and right columns contain skeletons, AOF/ridgeness and thresholded AOF/ridgeness maps, respectively. For each measure, two scales $r, \sigma \in \{1, 4\}$ are tested and thresholds are set accordingly. Note that th_{aof} only depends on α_0 and is thus left unchanged when r varies. Conversely, th_{rdg} is set according to α_0 and σ . Note that the color scale of the AOF map, in the center column of Fig. 6.11, is inverted so that it can be easily interpreted and compared to the ridgeness map.

In the right columns of Figs. 6.11 and 6.12, black pixels correspond to all p for which $aof(p, r) \leq th_{aof}$ or $rdg(p, \sigma) \geq th_{rdg}$. Note that this thresholding does not correspond to the final skeleton, as it has gaps and is not thin. A visual inspection shows that a skeleton branch emanates from each connected component of these selected pixels. For both AOF and ridgeness, the amount of connected components of thresholded pixels diminishes as the scale is increased. Simultaneously, the thickness of the central connected components, arising from the most significant shape parts, increases. This corroborates the expressions of AOF and ridgeness of points near regular skeleton branches, in Eqs. (6.14) and (6.16). Let p be a point in the vicinity of the skeleton branch and d its distance to the nearest regular skeleton point. We derive the conditions according to which p is selected as a candidate skeleton point, i.e. $aof(p, r) \leq th_{aof}$ or $rdg(p, \sigma) \geq th_{rdg}$, with respect to d and a given α , the object angle of the considered branch. We assume that $\alpha \in [\alpha_0, \frac{\pi}{2}]$. Regarding the AOF, according to Eq. (6.14), p satisfies $aof(p, r) \leq th_{aof}$ if

$$-\frac{2}{\pi r} \sin \alpha \sqrt{r^2 - d^2} \leq -\frac{2}{\pi} \sin \alpha_0,$$

which implies

$$d \leq r \sqrt{1 - \frac{\sin^2 \alpha_0}{\sin^2 \alpha}}. \tag{6.49}$$

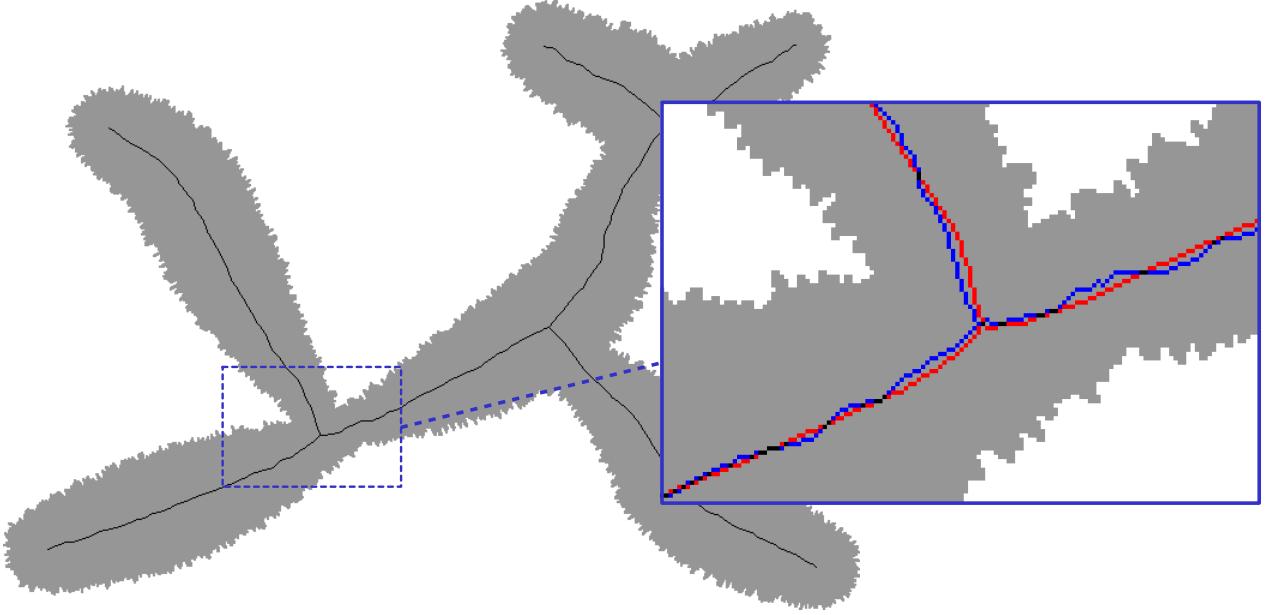


Figure 6.15: Accuracy of AOF-based skeleton for synthetic shape n°5 at noise level $\eta = 3$. In the zoomed part, **blue** pixels belong to the computed skeleton, whereas **red** pixels belong to the ground truth skeleton

Similarly, regarding the ridgeness, plugging the inequality $\text{rdg}(\mathbf{p}, \sigma) \geq \text{th}_{\text{rdg}}$ into Eq. (6.16) leads to

$$\frac{\sqrt{2\pi}}{\pi\sigma} \sin \alpha \exp\left(-\frac{d^2}{2\sigma^2}\right) \geq \frac{\sqrt{2\pi}}{\pi\sigma} \sin \alpha_0,$$

which implies

$$d \leq \sigma \sqrt{-2 \log\left(\frac{\sin \alpha_0}{\sin \alpha}\right)}. \quad (6.50)$$

According to Eqs (6.49) and (6.50), at a fixed object angle α , the distance d below which pixels will be thresholded as candidate skeleton points increases as r or σ gets larger, in agreement with our observation.

Quantitative study of accuracy

We study and compare quantitatively the accuracy of skeletons generated using the AOF-based and ridgeness-based thinning procedures, under variations of parameters and thresholds. Quantifying the accuracy of the skeletonization algorithms requires images where the expected structures of skeletons are known. For this purpose, we created a dataset of 20 synthetic shapes. Various curved centerlines were first manually generated. These centerlines were then dilated by using circular masks with smoothly varying radii along their entire length. This allows expected skeleton branches to be known beforehand. The expected skeleton branches correspond to the initial centerlines, except in junction areas, which thus need to be corrected. For each shape, the ground-truth reference skeleton was generated by correcting these junction areas using those of the AOF-based skeleton with $r = 2$ and th_{aof} value selected as in Eq. (6.48). An overview of this dataset is shown in Fig. 6.13.

In order to study the influence of contour noise on the choice of parameters r and σ , and their respective thresholds, the shapes were corrupted with additive white Gaussian noise at different intensities. We achieved this by moving contour points along their unit normal vector, with an offset randomly drawn from a zero-mean Gaussian with standard deviation η . A particular shape of the dataset at noise levels $\eta \in \{0, 1, 2, 3\}$ is depicted in Fig. 6.14. Note that $\eta = 0$ corresponds to the initial uncorrupted shapes, from which the

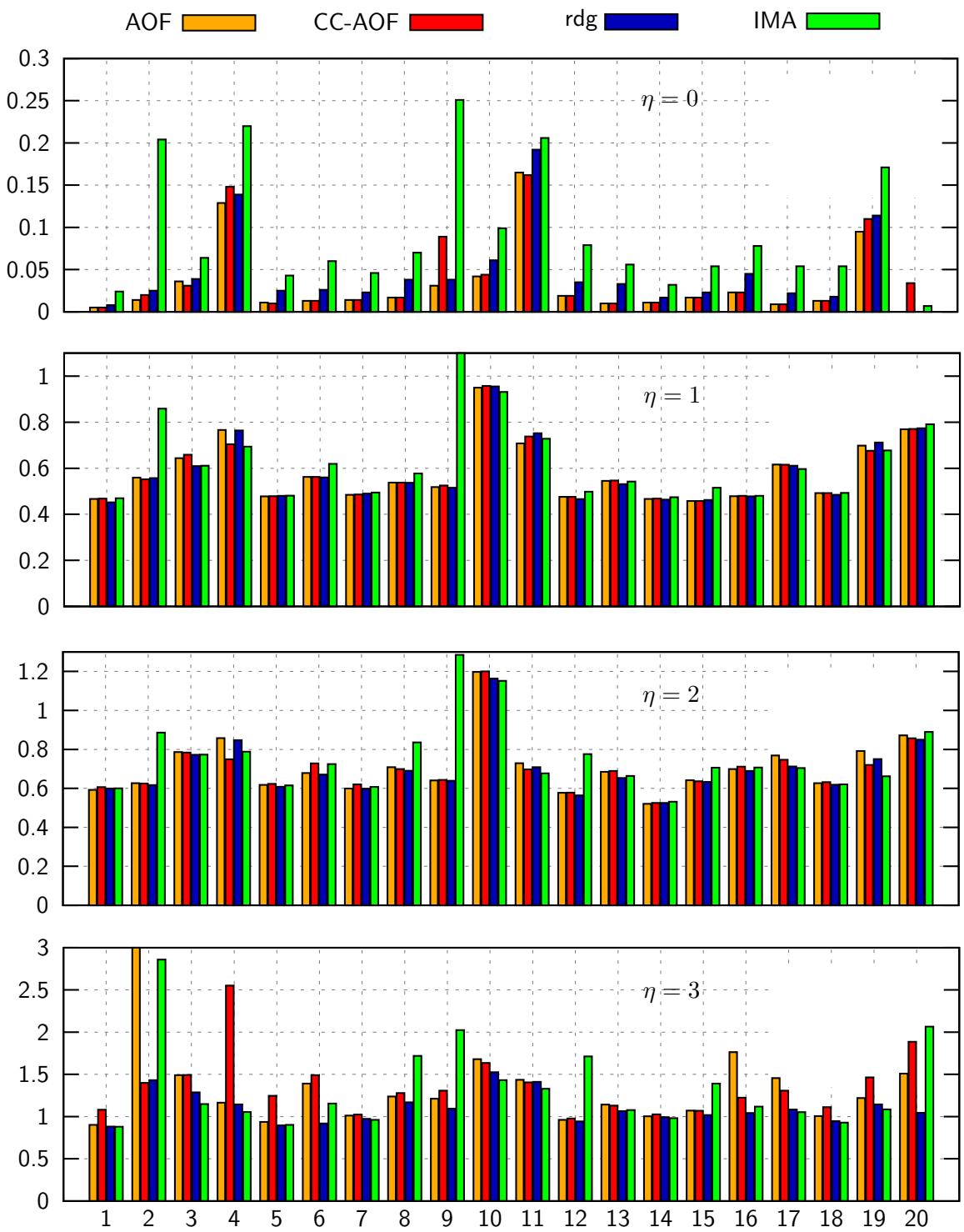


Figure 6.16: Modified Hausdorff distance between ground truth skeleton and computed skeleton, for the individual 20 synthetic shapes, at 4 different noise levels

	AOF	CC-AOF	rdg	IMA
$\eta = 0$	0.034	0.040	0.046	0.094
$\eta = 1$	0.584	0.582	0.583	0.632
$\eta = 2$	0.711	0.703	0.695	0.760
$\eta = 3$	1.333	1.346	1.101	1.359

Table 6.1: Modified Hausdorff distance between ground truth skeleton and computed skeleton, averaged over the 20 synthetic shapes, at 4 different noise levels. For each noise level, the lowest and highest MHD are highlighted in blue and red, respectively

ground-truth skeletons are extracted.

Accuracy is measured based on the similarity between the extracted skeleton and the ground-truth skeleton. We use the Modified Hausdorff distance (MHD) in the Euclidean sense:

$$\text{MHD}(P, Q) = \max \left\{ \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|p - q\|, \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \|q - p\| \right\}$$

where P and Q are non-empty subsets of \mathbb{Z}^2 (the extracted skeleton and the ground-truth skeleton). The discrepancy between the extracted and ground-truth skeleton is illustrated in Fig. 6.15.

In addition to the AOF-based and ridgeness-based skeletons, we report results obtained with the corrected AOF method of [Torsello and Hancock, 2006], as well as the Integer Medial Axis [Hesselink and Roerdink, 2008]. On the one hand, the AOF arises from the divergence of ∇D , or equivalently, the curvature of the front propagating along ∇D [Siddiqi et al., 2002]. According to [Torsello and Hancock, 2006], the error in calculating the AOF is related to the pixel resolution but is also proportional to the curvature. Hence, they developed a method that alleviates the contribution of the curvature to the error, by taking into account variations of curvature density. This led to the correction of curvature density effects on the AOF, that is subsequently referred to as CC-AOF. On the other hand, the Integer Medial Axis (IMA) algorithm is based on a discrete modeling of the shape. In addition to D , it uses the *feature transform*, which maps each shape point to the set of closest boundary points:

$$\text{FT}(x) = \{y \in \partial\Omega \mid \|x - y\| = D(x)\}$$

The AOF and ridgeness-based skeletonization methods include pruning natively. The pruning level is controlled by th_{aof} and th_{rdg} , respectively. Similarly, the IMA integrates pruning in the criterion used to select skeleton points. This criterion implies the distance between feature transform points of neighboring shape points. Three pruning modes are proposed, depending on the form of the function of this distance: constant pruning, linear pruning and square-root pruning. Constant and linear pruning criteria depend on a parameter γ , which is varied in the experiments⁵.

Radius r and scale σ were both varied from 1 to 5 with a step of 0.1. Threshold th_{aof} was varied from -1 to 0 with a step of 0.02, whereas threshold th_{rdg} was varied from 0 to 1 with a step of 0.02. For the IMA, the best results were obtained with the constant pruning mode, with parameter γ varying from 10 to 50. For each pair $(r, \text{th}_{\text{aof}})$ (and correspondingly, $(\sigma, \text{th}_{\text{rdg}})$ and γ), the AOF, CC-AOF, ridgeness and IMA skeletons were generated from the 20 shapes at the 4 different noise levels.

⁵For the CC-AOF, we used the skeleton module by F.-X. Dupé integrated in D. Tschumperlé’s CImg library: <https://github.com/dtschump/CImg>. For the IMA, we used our own C++ translation of the Java implementation available at <http://wimhesselink.nl/imageproc/skeletons>

In Fig. 6.16, the MHD is graphically represented on a per-shape basis. For each shape at each noise level, we retained the configurations of $(r, \text{th}_{\text{aof}})$, $(\sigma, \text{th}_{\text{rdg}})$ and γ that resulted in the most accurate skeleton. It is not straightforward to bring out a clear trend from Fig. 6.16, except that the IMA skeleton gives lower accuracy than the three other ones at noise level $\eta = 0$. CC-AOF seems to give the best results at noise level $\eta = 0$, whereas the ridgeness-based skeleton seems to deal better with noisy shapes. Note that the y-scale in Fig. 6.16 is different across noise levels. To get an overall view of the performances, results listed in Fig. 6.16 are averaged in Table 6.1. On noisy shapes, it is observed that the ridgeness-based skeleton outperform AOF-based ones. It is slightly more accurate at noise level $\eta = 2$ and significantly better at noise level $\eta = 3$. This is expected from the LoG filtering embedded in the ridgeness measure, which integrates regularization of the distance map into the ridge-detection process.

The previous experiments consider the skeletonization algorithms with their most favorable parameter tuning, but does not report their behavior with respect to the parameters. In Fig. 6.17, the MHD is averaged over the 20 shapes, for each couple $(r, \text{th}_{\text{aof}})$ of the AOF (top row) and CC-AOF (middle row) and each couple $(\sigma, \text{th}_{\text{rdg}})$ of the ridgeness (bottom row), at different noise levels. The IMA having only one parameter, equivalent plots could not be obtained, hence we did not include it into this study. Notice that, in the top and middle rows, values of th_{aof} increase downwards. First, it can be seen from the general appearance of the MHD surfaces that accuracy smoothly evolves with respect to parameters and thresholds. For the AOF and ridgeness, large Regions of Accurate Skeletons (RAS), with characteristic shapes, are observed. Unsurprisingly, as a general trend, accuracy falls as the noise level increases. In each plot, the area above the RAS corresponds to over-pruned skeletons, generated with AOF and ridgeness maps that were thresholded too hard. In this case, the skeleton is almost empty, all candidate skeleton points being filtered out. Conversely, the area below the RAS is related to under-pruned noisy skeletons with undesirable branches, due to loose thresholding. For the ridgeness, hyperbola-shaped RAS are observed, indicating that the optimal threshold is an inverse function of scale σ , which supports, among others, our derivations that led to Eqs (6.17) and (6.34). As claimed in [Torsello and Hancock, 2006], the correction of curvature density effects, as a postprocessing step in the CC-AOF, makes the AOF significantly less sensitive to parameter tuning. The interpretation is that it filters out noisy branches while reinforcing the AOF on desired branches. No area of empty skeletons can be observed, unlike in the AOF.

6.3 Matching with skeletons

We now address skeleton-based shape classification. Suppose we are given two shapes $\Omega^{(1)}$ and $\Omega^{(2)}$, one being the query of unknown category, and the other one being an example of a known class. To measure the dissimilarity between the two shapes, the idea is to determine the best assignment between their parts, as illustrated in Figure 6.18. The dissimilarity will be a function of the sum of costs of assigning each part of $\Omega^{(1)}$ to a part of $\Omega^{(2)}$. The skeleton is used to provide a representation of the different shape parts. It tends to have the desirable properties listed at the beginning of the current chapter, such as low sensitivity to affine transformations and small non-rigid deformations.

6.3.1 The need for hierarchizing branches

The advantage of the skeleton for matching shape parts is that there is a natural correspondence between skeleton branches and shape parts. However, the skeleton has an important shortcoming in this regard, as it is sensitive to noise on the shape contour. Even if a perturbation of the object border only modifies a limited part of the skeleton, this change can be significant. Consider Fig. 6.2 again. The details of the border, which can be negligible in comparison to the entire shape, can thus generate long branches, as highlighted in [Pizer et al., 1987]. Moreover, the length of a branch is not proportional to the size of the perturbation from which it is derived. Instead, it is proportional to the thickness of the shape at the place of the perturbation. This is the reason why it is relevant to attribute more or less importance to skeleton branches, in planar shape matching. The principle is that important branches come from the general appearance of the shape, whereas

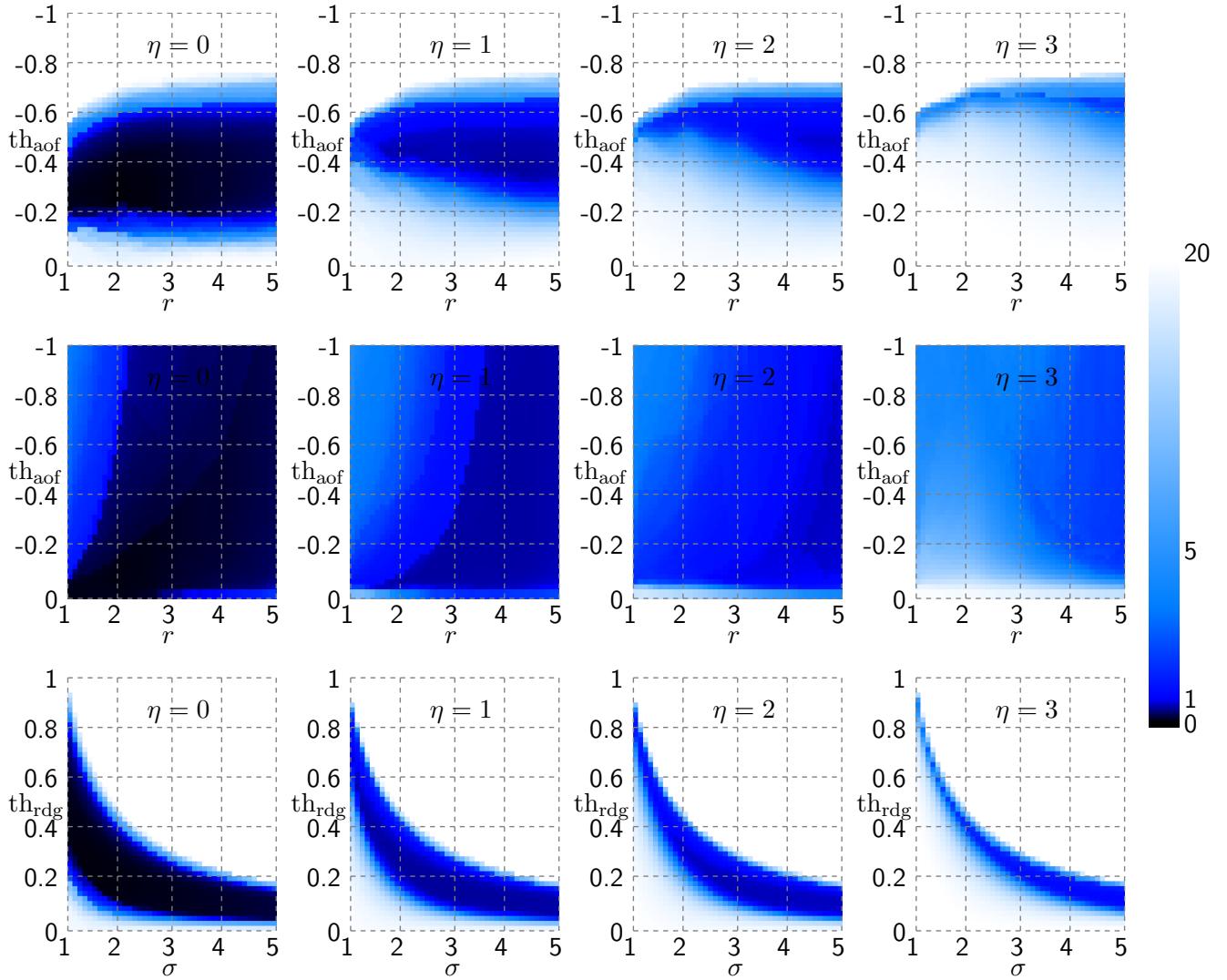


Figure 6.17: Modified Hausdorff distance between ground truth skeleton and computed skeleton, averaged over all shapes, versus parameter and threshold, at 4 different noise levels. Top row: AOF, middle row: CC-AOF, bottom row: ridgeness

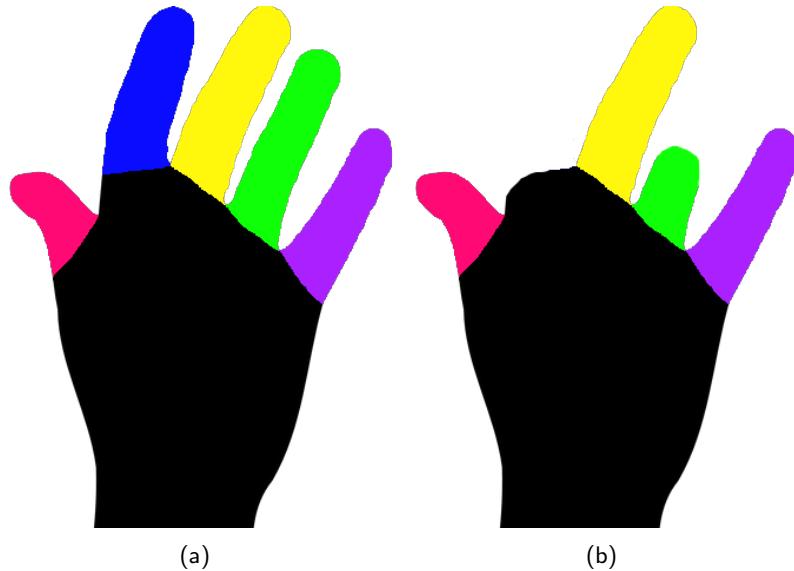


Figure 6.18: Matching of the different parts of two hands.

branches arising from small border details are negligible.

In the literature, hierarchization of skeleton branches was addressed in various ways. Some authors use kinship relationships. In this category of methods, the hierarchy is represented by a parent-child relationship between neighboring branches. The "parent" branch is the one of greatest importance, among neighboring branches. To do this, [Ogniewicz and Kübler, 1995] assign a degree of importance to each skeleton point. As written before Eq. 6.1, each skeleton point is the center of a disk having at least two tangent contour points. The greater the distance between these two points, the more important the branch is. In [Pizer et al., 1987], the authors have the same view of the skeleton hierarchy. To give a measure of importance to each branch, they note that, by smoothing the shape successively and recalculating the skeleton on each smoothed shape, the branches gradually disappear. However, there are still many problems to be solved, such as the loss of the topology of the shape during smoothing or a bad correspondence between the successive skeletons.

Another way of hierarchizing the skeleton is the use of shock graphs, developed in [Siddiqi et al., 1999] and used in [Sebastian et al., 2004], in particular. The system of shocks comes from the process of evolution of a front, starting from the border of the shape, such as the grassfire transform [Blum, 1967], and evolving at constant speed, in the direction normal to the curve. In other words, a shock graph is a dynamic interpretation of a skeleton in which shocks are singular points formed by colliding fronts. More recently, researchers in [Macrini et al., 2008, Macrini et al., 2011a, Macrini et al., 2011b] worked on bone graphs. The main idea here is that the whole skeleton is not necessary to encode the shape. In fact, certain parts of the branches encode the prominent parts of the shape contour. On the other hand, the concavities of the border located on each side of the prominent parts are responsible of special points of the skeleton called ligatures in which the skeleton points are grouped by segment. Since the skeleton pixels belong to a ligature or encode the prominent parts of the shape, to create a DAG, the authors chose meaningful names that refer to the anatomy. A bone is associated with a protuberance on the shape contour and a ligament connects two bones. A ligament is therefore an oriented edge. The attribute of each edge represents the position where the child bone is connected to the parent bone.

Multiscale approaches were also addressed. The hierarchy can also be seen as a sequence of skeletons, in which each skeleton corresponds to a representation of the shape at a certain scale. In this context, [Borgefors et al., 2001] construct a sequence of skeletons by downsampling the image iteratively. As the size of the image decreases, details disappear gradually. Unlike [Pizer et al., 1987], the multiscale representation

is built by successive downsampling, instead of smoothing. The skeleton is recalculated on each image and corresponding parts are matched across scales. Thus, the more the skeleton parts persist and the more important they are. In [Yang et al., 2016], the authors use a polygon representation of the contour of the shape, in which each convex vertex is the origin of a skeleton branch, as in [Bai et al., 2007]. They simplify this polygon progressively by removing the vertices one by one using a criterion based on length and angle. Thus, the corresponding skeleton is simplified iteratively.

Finally, other approaches imply the notion of hierarchy by using pruning, such as in the λ -medial axis [Chazal and Lieutier, 2005, Chaussard et al., 2011], which defines the *weak feature size* (wfs) as the minimum distance between $\mathcal{D} \setminus \Omega$ (the complement of the shape) and the critical points of the distance function of Ω . The obtained λ - medial axis is a subset of the skeleton when $\lambda < \text{wfs}$. Details of the skeleton are removed as λ is increased. In addition, the authors in [Shen et al., 2011] created the Bending Potential Ratio (BPR) to give a value of importance to each branch. This measure is based on the importance of the contour segments taking into account contextual information. Based on the polygonal approximation of [Bai et al., 2007], [Montero and Lang, 2012] select skeleton pixels with equidistant border points on two sides of the shape, which also creates a hierarchy of skeletons. Similarly, in [Yang et al., 2016, Liu et al., 2013], a polygonal approximation is considered. The most prominent convex border points are extracted, as these are potentially useful to remove noise. In order to know if a branch coming from a salient point can be removed without modifying the general appearance of the shape, the authors use a measure of global importance (percentage of reconstruction of the branch used in [Bai et al., 2007, Attali et al., 1995]) and local importance (length of the part of the branch that reconstructs a part of the shape by itself). As for [Serino and Sanniti di Baja, 2016], they evaluate the importance of each branch by taking into account the loss between the initial shape and the reconstructed shape from the skeleton to which branches are successively removed.

6.3.2 Skeleton matching

We provide a summary of the review on skeleton-based matching, which was presented in Aurélie Leborgne's PhD manuscript. For this detailed survey, the reader can refer to [Leborgne, 2016, Chapter 2].

Skeleton matching can be reformulated as a problem of attributed graph matching. From a general perspective, graph matching methods can be divided into two broad categories [Riesen et al., 2010]: exact graph matching and inexact (or error-tolerant) graph matching. Exact matching is related to the problem of (sub)graph isomorphism, *i.e.* detecting identical (sub)structures of two graphs and their corresponding attributes. This category assumes the existence of only noise-free objects, which is never in practice in image applications. Hence, the majority of methods for shape recognition uses inexact graph matching [Bunke and Allermann, 1983], in which error tolerance is part of the process. Most skeletal graph matching methods exploit hierarchy, which can be represented by:

- arcs in a directed graph: the direction of the arc induces a hierarchy;
- values in an attributed graph: a value represents the hierarchical level of the branch.

On the one hand, many methods based on Directed Acyclic Graphs (DAG) use the Graph Edit Distance (GED), which can be viewed as an extension of Levenshtein's distance on strings [Levenshtein, 1966]. The principle of GED is to define a cost for each editing operation (addition, removal or substitution), and to determine the sequence of editing operations to transform one graph into another with minimal total cost. The GED can be applied to various types of graphs (see the survey in [Gao et al., 2010]). It was used in particular on the above-mentioned shock graphs [Siddiqi et al., 1999, Sebastian et al., 2004].

Another important category of error-tolerant matching methods is recursive matching. Overall, this mechanism consists in iteratively constructing a solution by adding node assignments that are compatible with those already present in the partial solution. The search is generally guided by a similarity value. More precisely, given two DAGs, the general idea is to recursively match the nodes. This type of algorithm was used in particular on the

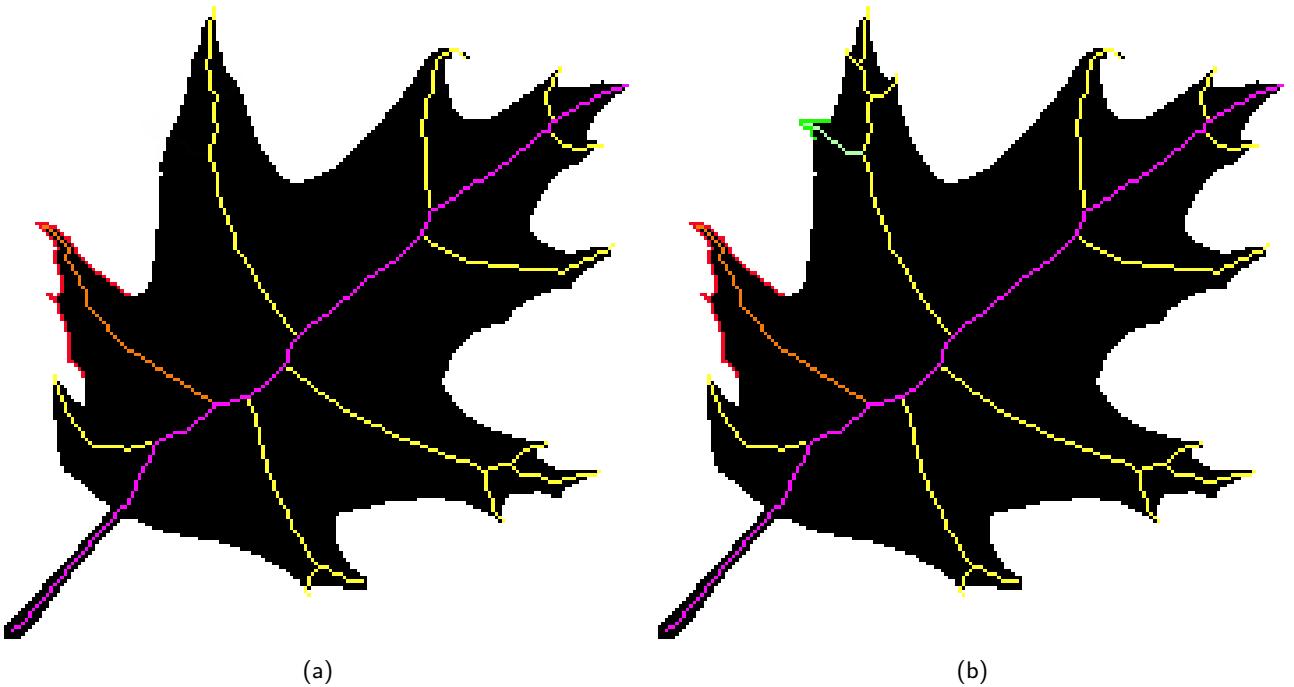


Figure 6.19: Branch sequences with variable weights. (a) Skeleton of a leaf, (b) Skeleton of the same leaf with additional details.

bone graphs that we previously described [Macrini et al., 2008, Macrini et al., 2011a, Macrini et al., 2011b]) in which the nodes represent the bones and the arcs represent the ligatures.

On the other hand, skeletons can be transformed into Attributed Relational Graphs (ARG). Dissimilarity can be calculated using one-to-one matching, which should be used when, in addition to the dissimilarity value, a strict correspondence between vertices is needed. In [Bai and Latecki, 2008, Shen et al., 2013], the authors match ARGs using the Hungarian algorithm [Kuhn, 1955, Munkres, 1957], which solves the assignment problem in a valued bipartite graph in polynomial time. Otherwise, dissimilarity can be calculated using many-to-many matching, like in [Keselman et al., 2003, Demirci et al., 2006]. The Dynamic Time Warping (DTW) applied on a specific ordering of terminal skeleton points in [Yasseen et al., 2016] is another example of many-to-many matching.

6.4 Building a weighted skeleton

The development of a matching method based on a hierarchical skeleton was originally motivated by extending the path-similarity method [Bai and Latecki, 2008], which has been reviewed in Section 6.3.1, on two aspects: decreasing its complexity, and integrating hierarchy. Regarding the first aspect, an in-depth study of the complexity of [Bai and Latecki, 2008] will be provided in Section 6.5.5, in comparison to the one of our contribution. Regarding the second aspect, we present in the current section how skeleton hierarchy is embedded into our framework.

6.4.1 Branches and branch sequences

When matching two shapes using their skeletons, it seems natural to pair skeleton branches with the same order of importance, or *weight*. We actually study the importance of *branch sequences*, rather than branches. Consider the leaf, in Fig. 6.19, which is a typical object where the hierarchy naturally arises. The main nerve of the leaf, drawn in pink, running from the tail to the apex, is a *branch sequence* (BS). A precise

graph-theoretical formulation of branches and BSs is given below. For now, we give an intuitive explanation of these two types of skeleton elements. A branch links endpoints or junctions, without passing through any junction, while a BS, made up of several branches, can have intermediate junctions.

Quantification of the importance of each BS can be applied to skeletons because each comes from a protusion of the shape border, of variable importance. Each BS corresponds to a specific shape part. Consider the two shapes in Fig. 6.19, which are very similar. The skeleton of shape in Fig. 6.19(b) has an extra branch (drawn in green), due to additional shape details (the green portion of the shape contour). The BSs corresponding to these details should have very little weight for matching. Obviously, the orange skeleton branch should be considered as more important than the green one in the matching process.

To determine the weight of BSs, we propose a multiscale representation of the shape, where shape details are gradually smoothed out as the scale increases. Consequently, corresponding skeleton branches are gradually removed. To quantify the weight of a BS, we use the scale at which it disappears. A BS has greater importance as it persists across a larger number of scales. We should emphasize the use of BSs instead of branches in the hierarchization of the skeleton. Considering the example in Fig. 6.19, one can imagine that secondary leaf protusions will disappear as the shape is iteratively smoothed, and that the main nerve will persist. Branches composing this main nerve should have the same order of weight, and therefore they are expected to belong to the same BS.

We use a time-varying undirected graph representation, referred to as *skeleton graph*

$$\text{SG}^t = (\text{SG}_V^t, \text{SG}_E^t, \phi^t).$$

We omit time superscript for notational convenience. SG_V is the set of skeleton vertices $\{\mathbf{v}_j\}_{j=1\dots|\text{SG}_V|}$, where each $\mathbf{v}_j \in \mathbb{R}^2$ (but could be restricted to \mathbb{Z}^2). SG_E is the set of undirected edges, i.e. a set of two-sets of vertices.

The initial skeleton graph SG^0 is built from the initial discrete skeleton $\hat{S}(0)$. Recall that $\hat{S}(0)$ is a subset of the discrete image domain. Its elements live in \mathbb{Z}^2 and it does not contain any adjacency relationship between skeleton pixels, whereas SG store adjacency relations explicitly in graph edges, between vertices that are allowed to evolve in \mathbb{R}^2 . Since $\hat{S}(0)$ is 8-connected and of unit thickness, the construction of SG is trivial. A vertex is created at each pixel in $\hat{S}(0)$, and an edge is created for each pair of 8-connected pixels.

The labeling is an integer set-valued function of edges,

$$\phi : \text{SG}_E \longrightarrow 2^{\mathbb{N}},$$

whose purpose will be to indicate from which branch(es) the edge comes, and thus to group branches in BSs. The initial labeling ϕ^0 is arbitrary, provided that two different branches have different labels, and that the labeling is the same for all edges of that branch. As explained below, to allow the future construction of the hierarchical skeleton, the labels of the initial branches will be transferred to merged branches when removing branches. The labels of the initial labeling ϕ^0 are partly stored in the branches of skeleton SG^t at any time t .

The neighborhood $\eta_j \subset \text{SG}_V$ is the set of skeleton vertices adjacent to \mathbf{v}_j ,

$$\eta_j = \{\mathbf{v}_k \in \text{SG}_V \mid \{\mathbf{v}_j, \mathbf{v}_k\} \in \text{SG}_E\}.$$

We assume that no isolated skeleton vertex exists, so $|\eta_j| \geq 1$. The type of skeleton vertex \mathbf{v}_j can be determined from its degree, the number of adjacent vertices:

- If $|\eta_j| = 1$, \mathbf{v}_j is a terminal vertex.
- If $|\eta_j| = 2$, \mathbf{v}_j is a regular skeleton vertex.

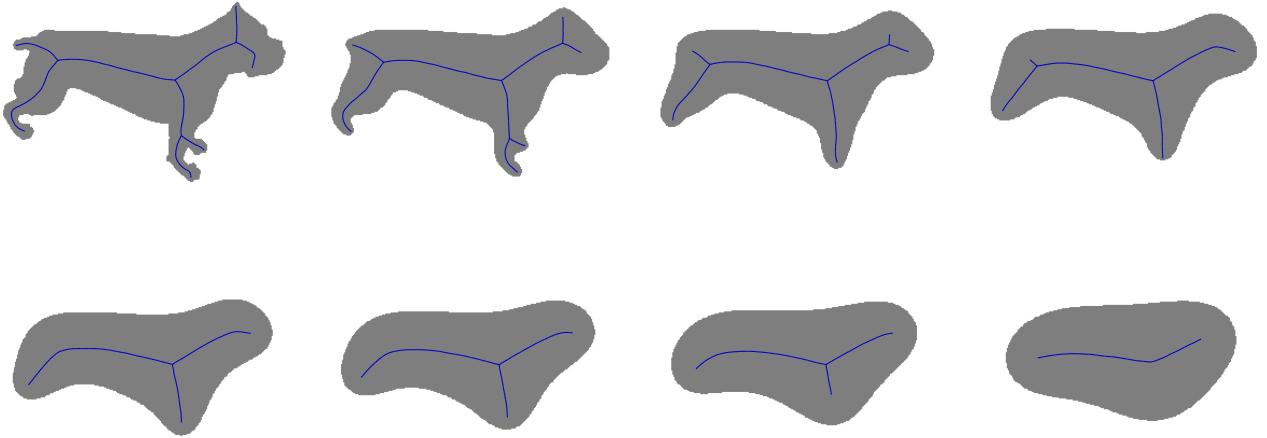


Figure 6.20: Evolution of shape and skeleton under the area-preserving smoothing flow

- If $|\eta_j| \geq 2$, v_j is a junction vertex.

We also assume that vertices of the skeleton graph form a regular unit sampling of the continuous skeleton S . Hence, if two skeleton vertices v_j and v_k are adjacent, it is expected that

$$\|v_j - v_k\| \approx 1.$$

In our discrete setting, a *branch* b is a simple path linking two non-regular skeleton vertices, *i.e.* a sequence of $n \geq 2$ vertices

$$b = (v_{s_1}, v_{s_2}, \dots, v_{s_{n-1}}, v_{s_n}),$$

such that v_{s_1} and v_{s_n} are terminal or junction vertices, whereas v_{s_2} to $v_{s_{n-1}}$ are regular skeleton vertices. Notice that a junction vertex belongs to at least three branches. Note also that a branch may be composed of two terminal or junction vertices only, and the edge linking them (in which case, there is no regular vertex between them).

A *branch sequence* (BS) μ is a simple path linking two non-regular skeleton vertices, *i.e.* a sequence of $n \geq 2$ vertices

$$\mu = (v_{s_1}, v_{s_2}, \dots, v_{s_{n-1}}, v_{s_n}),$$

such that v_{s_1} or v_{s_n} is a terminal vertex, the other one being a terminal or junction vertex. Vertices v_{s_2} to $v_{s_{n-1}}$ are regular skeleton vertices or junction vertices. A BS can be equivalently defined as a sequence of branches connected two-by-two by their end vertices. Incidentally, a branch belongs to only one BS.

6.4.2 The continuous skeleton under area-preserving shape smoothing

Let us leave the graph-based representation for a while, and study the deformation of the continuous space-time skeleton $S(t)$ in the case the shape contour evolves in time, according to a smoothing flow. As illustrated in Fig. 6.20, the skeleton has its branches progressively pruned as the contour is increasingly smooth. We consider an area-preserving regularization flow, making the shape contour converge to a circle. The method is thus restricted to simply-connected shapes, whose contours can be represented by a unique simple closed curve. The evolution of the shape through time builds an area-preserving scale-space [Sapiro and Tannenbaum, 1995], which is relevant for studying a multiscale representation of the

shape, while keeping the proportion⁶.

The evolution of the shape takes place in a continuous space-time setting. The shape contour of the given shape is described by \mathbf{c} , a time-varying C^2 simple closed curve, parameterized over range $[0, 1]$,

$$\begin{aligned}\mathbf{c} &: [0, 1] \times \mathbb{R}^+ \longrightarrow \mathcal{D} \\ \mathbf{c}(u, t) &= [x(u, t) \ y(u, t)]^\top.\end{aligned}$$

In what follows, $\mathbf{c}(\cdot, t)$ denotes the entire curve at fixed time t ,

$$\mathbf{c}(\cdot, t) : [0, 1] \longrightarrow \mathcal{D},$$

whereas $\mathbf{c}(u, t)$ is a position. The time-varying shape is $\Omega(t) = \text{in}(\mathbf{c}(\cdot, t)) \subset \mathcal{D}$.

Function $D : \mathcal{D} \times \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ is the time-varying signed Euclidean distance transform of the shape,

$$D(\mathbf{x}, t) = \begin{cases} \min_{u \in [0, 1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \\ -\min_{u \in [0, 1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{out}(\mathbf{c}(\cdot, t)). \end{cases} \quad (6.51)$$

The continuous skeleton at time t , denoted $S(t)$, is defined by plugging Eq. (6.51) into Eq. (6.1). It is the set of centers of balls having at least two contact points on $\mathbf{c}(t)$,

$$S(t) = \{\mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \mid \exists u_1, u_2 \in [0, 1], u_1 \neq u_2, \|\mathbf{x} - \mathbf{c}(u_1, t)\| = \|\mathbf{x} - \mathbf{c}(u_2, t)\| = D(\mathbf{x}, t)\}. \quad (6.52)$$

Notice that S is a set-valued function of t . The shape contour \mathbf{c} evolves according to the area-preserving flow [Gage, 1986, Dolcetta et al., 2002],

$$\frac{\partial \mathbf{c}(u, t)}{\partial t} = \left(\kappa(u, t) - \frac{2\pi}{L[\mathbf{c}(\cdot, t)]} \right) \mathbf{n}(u, t), \quad (6.53)$$

where \mathbf{n} is the inward unit normal, $L[\mathbf{c}]$ is the Euclidean length of \mathbf{c} and κ is the signed curvature (see Eqs (4.2), (4.3) and (4.4), respectively). Conservation of the area is proven in [Dolcetta et al., 2002],

$$\frac{dA[\mathbf{c}(\cdot, t)]}{dt} = 0,$$

where A was defined in Eq. (4.5). As the curve converges to a circle whose area is equal to the area of the initial shape, the flow maximizes the isoperimetric quotient

$$\text{IQ}[\mathbf{c}] = \frac{4\pi A[\mathbf{c}]}{L[\mathbf{c}]^2}, \quad (6.54)$$

which is equal to 1, and maximal, when \mathbf{c} is a circle. As a stopping criterion, we use $\text{IQ}[\mathbf{c}] \geq 1 - \epsilon$, with small positive constant ϵ .

Putting together Eqs (6.51), (6.52) and (6.53), the time-varying skeleton $S(t)$ should be estimated using the system

$$\begin{cases} \frac{\partial \mathbf{c}(u, t)}{\partial t} = \left(\kappa(u, t) - \frac{2\pi}{L[\mathbf{c}(\cdot, t)]} \right) \mathbf{n}(u, t), \\ D(\mathbf{x}, t) = \begin{cases} \min_{u \in [0, 1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \\ -\min_{u \in [0, 1]} \|\mathbf{x} - \mathbf{c}(u, t)\| & \text{if } \mathbf{x} \in \text{out}(\mathbf{c}(\cdot, t)) \end{cases} \\ S(t) = \{\mathbf{x} \in \text{in}(\mathbf{c}(\cdot, t)) \mid \exists u_1, u_2 \in [0, 1], u_1 \neq u_2, \|\mathbf{x} - \mathbf{c}(u_1, t)\| = \|\mathbf{x} - \mathbf{c}(u_2, t)\| = D(\mathbf{x}, t)\}. \end{cases} \quad (6.55)$$

⁶as opposed to, for example, the mean curvature flow that can be deduced from Eq. (4.14), which makes the contour shrink to a single point

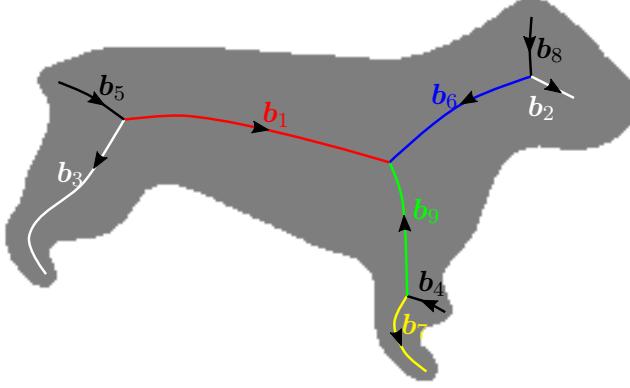


Figure 6.21: Set of continuous skeleton branches. Arrows indicate the orientation of parameterization over range $[0, 1]$

Ideally, we could calculate or, at least, approximate $\frac{dS(t)}{dt}$, so that explicit time-discretization could be applied, and estimate $S(t + \delta t)$ from $S(t)$, with an appropriately chosen time step δt . However, we lack the mathematical framework to derive, or even approximate, a closed-form expression for $\frac{dS(t)}{dt}$. The same remark holds for $\frac{dD(x, t)}{dt}$.

A naive method to overcome this issue would consist in computing a discrete skeleton \hat{S} from scratch, using the algorithm described in Section 6.1.4, after updating the distance map D at each iteration of the evolution flow (6.53). This would yield a complexity log-linear with respect to the shape area, $O(|\hat{\Omega}| \log |\hat{\Omega}|)$, multiplied by the number of iterations of (6.53). In order to decrease the complexity and exploit the temporal regularity of $c(\cdot, t)$, we thus developed a deformable skeleton, allowing to approximate $S(t + \delta t)$ given $S(t)$.

6.4.3 The continuous deformable skeleton

We introduce a continuous parameterization of $S(t)$, as a set of curves

$$SB(t) = \{b_j(\cdot, t)\}_{j=1 \dots |SB(t)|}. \quad (6.56)$$

In comparison to the skeleton graph SG, the continuous parameterization SB allows us to derive the motion equation of branch curves. It is relevant to describe the motion of skeleton at the continuous branch level, rather than on individual skeleton vertices, since we can highlight the variational energy-based interpretation. This motion equation will finally be implemented on the skeleton graph in Section 6.4.4.

We denote by $b_j(\cdot, t)$ the entire j^{th} branch at time t , whereas $b_j(u, t)$ denotes a particular curve point. At a given time t , we know that branch b_j is fully included into $\Omega(t)$,

$$b_j(\cdot, t) : [0, 1] \longrightarrow \Omega(t).$$

Each branch being an open curve, its endpoints are different,

$$b_j(0, t) \neq b_j(1, t).$$

The number of branches $|SB(t)|$ is an integer-valued, piecewise-constant, function of time. It is non-increasing, since no branch can be added by the smoothing flow.

We will refer to *terminal endpoint* an endpoint which is not connected to any other endpoint. Otherwise, it is a *junction endpoint*. Note that if the two endpoints of a branch b_j are terminal, it means that the skeleton

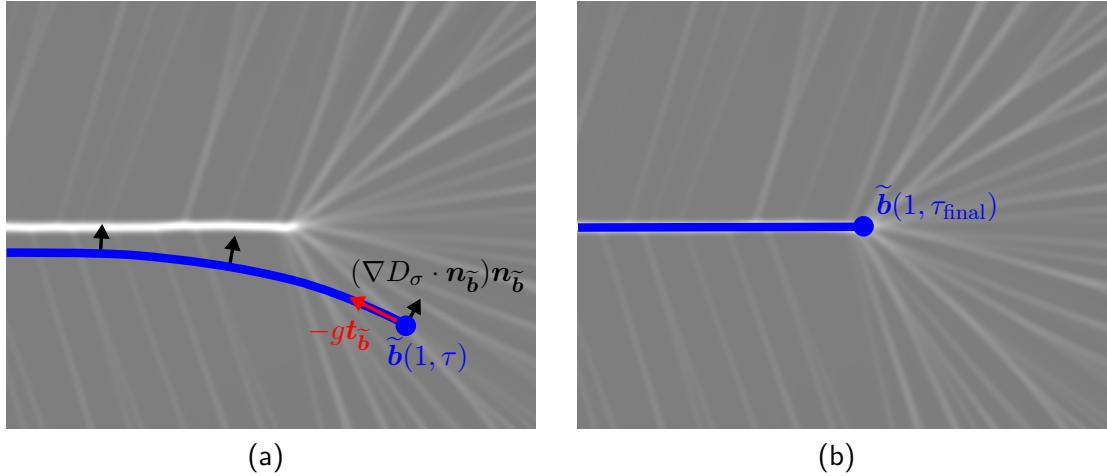


Figure 6.22: Deformable skeleton: a branch \tilde{b} under deformation (a) at time instant $\tau = 0$, with motions made up of normal components in black and tangential components in red. (b) at final time τ_{final}

is a single curve b_j . This restriction is not inherent to the definition of SB, but to the constraint that we only handle simply-connected shapes. In a more general setting, SB could contain cycles.

If a branch b_j is connected to another branch at, for example, its endpoint $b_j(0, t)$, we have

$$\exists k \in \{1, \dots, |\text{SB}(t)|\}, k \neq j \quad b_j(0, t) = b_k(0, t) \text{ or } b_j(0, t) = b_k(1, t).$$

It can be, equivalently, connected to another branch at its endpoint $b_j(1, t)$. A non-terminal endpoint will be usually connected to two branches. An example is shown in Fig 6.21. Endpoint $b_1(0, t)$ is connected to endpoint $b_3(0, t)$ and $b_5(1, t)$, whereas $b_3(1, t)$ and $b_5(0, t)$ are terminal endpoints.

Consider that shape contour c is evolved according to (6.53), for a duration $\tilde{\delta}t$, leading to an updated contour $c(\cdot, t + \tilde{\delta}t)$. Duration $\tilde{\delta}t$ is supposed to be greater than the time step δt . An updated distance map $D(\cdot, t + \tilde{\delta}t)$ is calculated. As the distance, the ridgeness function defined in Eq. (6.5) is also extended in time,

$$\text{rdg}(\mathbf{x}, t) = -(D * \Delta G_\sigma)(\mathbf{x}, t),$$

where $*$ is the convolution in space. Each branch $b_j(\cdot, t)$ is located on a ridge of $D(\cdot, t)$. The problem is to derive $\text{SB}(t + \tilde{\delta}t)$, by deforming branches in $\text{SB}(t)$, so that they fit to the ridges of the updated distance $D(\cdot, t + \tilde{\delta}t)$.

Nested evolution equation for evolving branches

We introduce a nested evolution equation, in parallel to the evolving system (6.55). The nested evolution equation deforms a given branch $b_j(\cdot, t)$ to $b_j(\cdot, t + \tilde{\delta}t)$. We introduce an auxiliary time variable τ , and an auxiliary branch \tilde{b} varying with respect to τ , such that

$$\begin{aligned} \tilde{b}(\cdot, 0) &= b_j(\cdot, t), \\ \tilde{b}(\cdot, \tau_{\text{final}}) &= b_j(\cdot, t + \tilde{\delta}t), \end{aligned}$$

where τ_{final} is the auxiliary time at which the nested evolution will be stopped. Informally, the temporary branch \tilde{b} lives in a "subtime" of t . The evolution is illustrated in Fig 6.22, where the background is the updated ridgeness $\text{rdg}(t + \tilde{\delta}t)$. At subtime τ_{final} , the auxiliary branch \tilde{b} stabilizes on the nearest ridge.

To achieve this deformation, we use a *skeleton cost function* $g : \mathcal{D} \rightarrow \mathbb{R}^+$ that can be equivalently defined either using the indicator function, either the Heaviside step function,

$$g(\mathbf{x}) = \mathbf{1}_{\{\text{rdg}(\cdot, t + \tilde{\delta}t) < \text{th}_{\text{rdg}}\}} = H(\text{th}_{\text{rdg}} - \text{rdg}(\mathbf{x}, t + \tilde{\delta}t)) \quad (6.57)$$

which vanishes if \mathbf{x} is on a significant ridge, and is equal to 1 otherwise. The ridgeness threshold th_{rdg} was introduced in condition (6.10). Evolution of the auxiliary branch is governed by the following *ridge-fitting* equation,

$$\left\{ \begin{array}{lcl} \frac{\partial \tilde{\mathbf{b}}(u, \tau)}{\partial \tau} & = & \left(\nabla D_\sigma(\tilde{\mathbf{b}}) \cdot \mathbf{n}_{\tilde{\mathbf{b}}} + \omega_{\text{reg}} \kappa_{\tilde{\mathbf{b}}} \right) \|\tilde{\mathbf{b}}'\| \mathbf{n}_{\tilde{\mathbf{b}}} \quad \forall u \in (0, 1) \\ \frac{\partial \tilde{\mathbf{b}}(0, \tau)}{\partial \tau} & = & \left(\nabla D_\sigma(\tilde{\mathbf{b}}) \cdot \mathbf{n}_{\tilde{\mathbf{b}}} \right) \|\tilde{\mathbf{b}}'\| \mathbf{n}_{\tilde{\mathbf{b}}} + g(\tilde{\mathbf{b}}) \mathbf{t}_{\tilde{\mathbf{b}}} \quad \text{if } \tilde{\mathbf{b}}(0) \text{ is a terminal endpoint} \\ \frac{\partial \tilde{\mathbf{b}}(1, \tau)}{\partial \tau} & = & \left(\nabla D_\sigma(\tilde{\mathbf{b}}) \cdot \mathbf{n}_{\tilde{\mathbf{b}}} \right) \|\tilde{\mathbf{b}}'\| \mathbf{n}_{\tilde{\mathbf{b}}} - g(\tilde{\mathbf{b}}) \mathbf{t}_{\tilde{\mathbf{b}}} \quad \text{if } \tilde{\mathbf{b}}(1) \text{ is a terminal endpoint} \\ \tilde{\mathbf{b}}(\cdot, 0) & = & \mathbf{b}_j(\cdot, t) \end{array} \right. \quad (6.58)$$

where $\mathbf{t}_{\tilde{\mathbf{b}}}$, $\mathbf{n}_{\tilde{\mathbf{b}}}$ and $\kappa_{\tilde{\mathbf{b}}}$ are the normal, tangent and curvature of $\tilde{\mathbf{b}}$ at a particular position and time. For conciseness, we dropped variables u and τ for them. In the first line of (6.58), normal, tangent and curvature are taken at (u, τ) . In the second and third line, they are taken at $(0, \tau)$ and $(1, \tau)$, respectively. The smoothed gradient of distance is also used, with dropped time $t + \tilde{\delta}t$,

$$\nabla D_\sigma(\mathbf{x}) = (\nabla G_\sigma * D)(\mathbf{x}, t + \tilde{\delta}t).$$

Weight ω_{reg} is a user-defined parameter, controlling the importance of the regularization term with respect to the data-fitting term. The derivation leading to (6.58) is detailed in Appendix C.2.

Let us detail the different lines of (6.58). The first line is the main evolution equation, that applies along the curve except at the endpoints. The deformation is oriented along normal $\mathbf{n}_{\tilde{\mathbf{b}}}$. The fist term is the data-fitting term, which drives the branch towards the nearest ridge. The gradient vector field ∇D_σ , is projected onto the normal. In relation to Eq. (6.9), the magnitude of the data-fitting term is bounded, as

$$\|\nabla D_\sigma(\mathbf{x})\| = \|(\nabla G_\sigma * D)(\mathbf{x}, t + \tilde{\delta}t)\| \leq 1.$$

At time τ_{final} , it is expected that stability condition

$$\left(\nabla D_\sigma(\tilde{\mathbf{b}}(u)) \cdot \mathbf{n}_{\tilde{\mathbf{b}}}(u) \right) \mathbf{n}_{\tilde{\mathbf{b}}}(u) = 0$$

is verified for every $u \in [0, 1]$ (variable τ_{final} is dropped). The second term in the first line, depending on curvature $\kappa_{\tilde{\mathbf{b}}}$, is the regularization term keeping the branch smooth. The second and third lines are the equations governing the motion of endpoints $\tilde{\mathbf{b}}(0, \tau)$ and $\tilde{\mathbf{b}}(1, \tau)$, respectively, if they are terminal. The evolution of endpoints contains an additional data-dependent term, $\pm g(\tilde{\mathbf{b}}) \mathbf{t}_{\tilde{\mathbf{b}}}$, which shortens the branch until the endpoint is located on a ridge.

We consider boundary condition $\kappa_{\tilde{\mathbf{b}}}(0) = \kappa_{\tilde{\mathbf{b}}}(1) = 0$, which makes the regularization term vanish at endpoints. Rigorously, normal and tangent vectors are undefined at endpoints. We take

$$\begin{aligned} \mathbf{t}_{\tilde{\mathbf{b}}}(0) &= \mathbf{t}_{\tilde{\mathbf{b}}}(\delta u) \\ \mathbf{t}_{\tilde{\mathbf{b}}}(1) &= \mathbf{t}_{\tilde{\mathbf{b}}}(1 - \delta u), \end{aligned}$$

and similarly for $\mathbf{n}_{\tilde{\mathbf{b}}}$. δu is an infinitesimal displacement along the curve.

Evolution equation (6.58) is relevant to describe the motion of one branch, with terminal endpoints. Its drawback is that it does not account for the merging or deletion of branches, nor the motion of junction points. The discretization described in Section 6.4.4 solves these issues. We have the intuition that the flow pushes $\tilde{\mathbf{b}}$ towards a local maximum of

$$E_{\text{skel}}[\tilde{\mathbf{b}}] = \int_0^1 \text{rdg}(\tilde{\mathbf{b}}(u)) \left\| \tilde{\mathbf{b}}'(u) \right\| du, \quad (6.59)$$

subjected to the pointwise inequality constraint,

$$\text{rdg}(\tilde{\mathbf{b}}(u)) \geq \text{th}_{\text{rdg}}, \quad \forall u \in [0, 1].$$

Without this constraint, the curve would grow infinitely⁷ outside of ridges. At the time of writing, we lack the proof which would lead to

$$\frac{dE_{\text{skel}}[\tilde{\mathbf{b}}(\cdot, \tau)]}{d\tau} \geq 0. \quad (6.60)$$

We leave this proof as a possible research track in Section 6.6.2.

6.4.4 Discretization

We describe the discretization process, both for the contour undergoing area-preserving regularization flow, and auxiliary branches deforming thanks to the ridge-fitting flow.

Contour, shape and distance map

Given the discrete input shape $\hat{\Omega}(0)$, the initial distance map $D(\cdot, 0)$ and discrete skeleton $\hat{S}(0)$ are computed thanks to the procedure described in Section 6.1.4. An explicit parameterization of the shape contour is built by contour linking, following boundary pixel edges in the clockwise orientation, which generates an initial sequence of contour nodes $(\mathbf{q}_i^0)_i$.

The area-preserving smoothing flow of Eq. (6.53) is discretized using centered finite differences (4.12) in space and forward explicit Euler scheme in time, with spatial step $\delta u = 1$ and time step $\delta t = 0.5$. Let curve \mathbf{c} be sampled as a sequence of nodes $(\mathbf{q}_i^t)_i$. The discrete evolution scheme for the i^{th} curve node is

$$\mathbf{q}_i^{t+1} = \mathbf{q}_i^t + \delta t \left(\kappa_i^t - \frac{2\pi}{\hat{L}^t} \right) \mathbf{n}_i^t. \quad (6.61)$$

where \hat{L} is the discrete length,

$$\hat{L}^t = \sum_i \left\| \mathbf{q}_i^t - \mathbf{q}_{i-1}^t \right\|.$$

To preserve a regular sampling, with a density of points similar to the one of the initial contour, it is necessary to resample the curve after each iteration of smoothing. Otherwise, the estimation of curvatures κ_i becomes unstable. The resampling ensures that

$$\left\| \mathbf{q}_i^t - \mathbf{q}_{i-1}^t \right\| \approx 1.$$

As explained in Section 6.4.3, the skeleton is not evolved at each iteration of the area-preserving smoothing flow. Evolution equation (6.61) is run for a number of iterations sufficiently high so that the displacement of the sampled curve $(\mathbf{q}_i^t)_i$ is at least 1 at some place. If we denote by t_0 the last time the distance map and skeleton were updated, an update of the skeleton graph is performed at time t if

$$\max_i \left\| \mathbf{q}_i^t - \mathbf{q}_i^{t_0} \right\| \geq 1.$$

In this case, the sampled curve $(\mathbf{q}_i^t)_i$, where each $\mathbf{q}_i^t \in \mathbb{R}^2$, needs to be converted back to a discrete shape $\hat{\Omega}(t)$. The contour sampled in \mathbb{R}^2 is converted to a digital curve (see Appendix A.2.2) and its interior is filled, so that the updated distance map $D(\cdot, t)$ can be computed.

⁷or, at least, until meeting a point with negative ridgeness

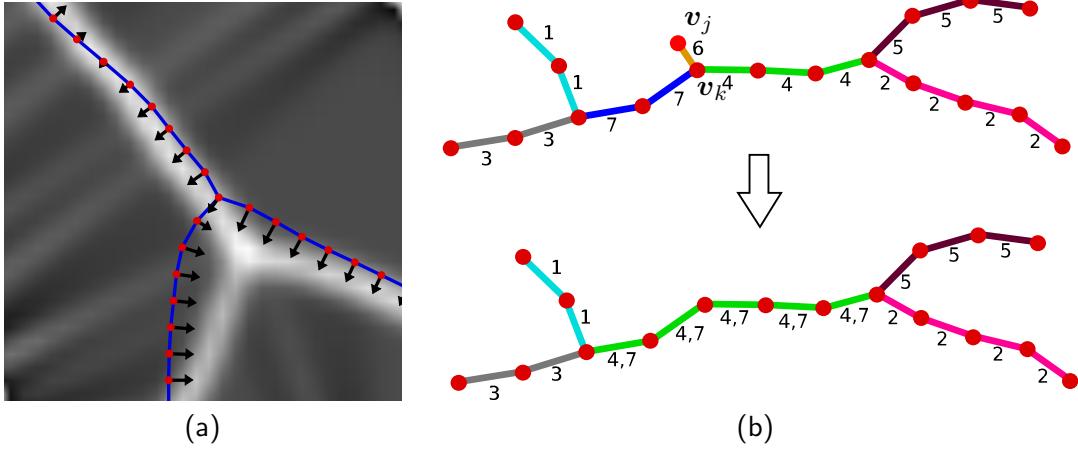


Figure 6.23: Evolution of the sampled skeleton: (a) Forces attracting vertices towards the ridges of the Euclidean distance map and (b) label propagation after deleting a branch : removing vertex v_j causes the disappearance of branch with label 6. The two adjacent branches (respectively labeled 4 and 7) are merged and labels are combined in the merged branch.

Moving skeleton vertices

Let us get back to the graph-based representation. The skeleton graph SG^t is deformed towards the ridges of the updated distance map by explicit Euler time-discretization of evolution system (6.58), at each skeleton vertex. This deformation happens between two updates of the distance map, in a nested time τ , with time step $\delta\tau$. As stated earlier, the continuous flow, which models the motion of branches independently, lacks the motion of junction points, which are involved in several branches. The discrete evolution of the sampled skeleton takes junction vertices into account.

Centered finite-differences are used for estimating normal vectors and curvatures at regular skeleton vertices. If v_j is a regular skeleton vertex with $\eta_j = \{v_a, v_b\}$, its normal is estimated as

$$\mathbf{n}_j = \frac{(\mathbf{v}_b - \mathbf{v}_a)^\perp}{\|\mathbf{v}_b - \mathbf{v}_a\|}.$$

At junction vertices, the normal is not used and is, consequently, left undefined. If v_j is a terminal vertex, with a single adjacent vertex v_k , its normal is taken as $\mathbf{n}_j = \mathbf{n}_k$.

Each vertex is moved thanks to a combination of data-fitting force, and regularization force, weighted by user-defined parameter ω_{reg} . The data-fitting force attracts the vertex towards the nearest ridge of $D(\cdot, t + \tilde{\delta}t)$. For regular skeleton vertices, the smoothed distance gradient ∇D_σ , directed towards a ridge, is locally projected onto the normal. Thus, branches already located on a ridge are not deformed, even if the distance varies along the ridge. For junction vertices, the gradient of the Euclidean distance map is taken as is, without projection. The data force for a given vertex v_j is

$$\mathbf{a}_j = \begin{cases} (\nabla D_\sigma(v_j) \cdot \mathbf{n}_j) \mathbf{n}_j & \text{if } |\eta_j| \leq 2 \\ \nabla D_\sigma(v_j) & \text{otherwise,} \end{cases} \quad (6.62)$$

where ∇D_σ is taken at time $t + \tilde{\delta}t$. Since $v_j \in \mathbb{R}^2$, $\nabla D_\sigma(v_j)$ is computed by bilinear interpolation. Note that the tangential component of the motion of terminal endpoints, which appeared in the second and third lines of Eq (6.58), is not integrated in the data-fitting force. Instead, since this tangential component only plays a role in the shortening of branches, it is rather implemented as a removal of terminal vertices. In relation with the skeleton cost defined in Eq. (6.57), a terminal vertex v_j is removed if

$$\text{rdg}(v_j) < \text{thrdg}.$$

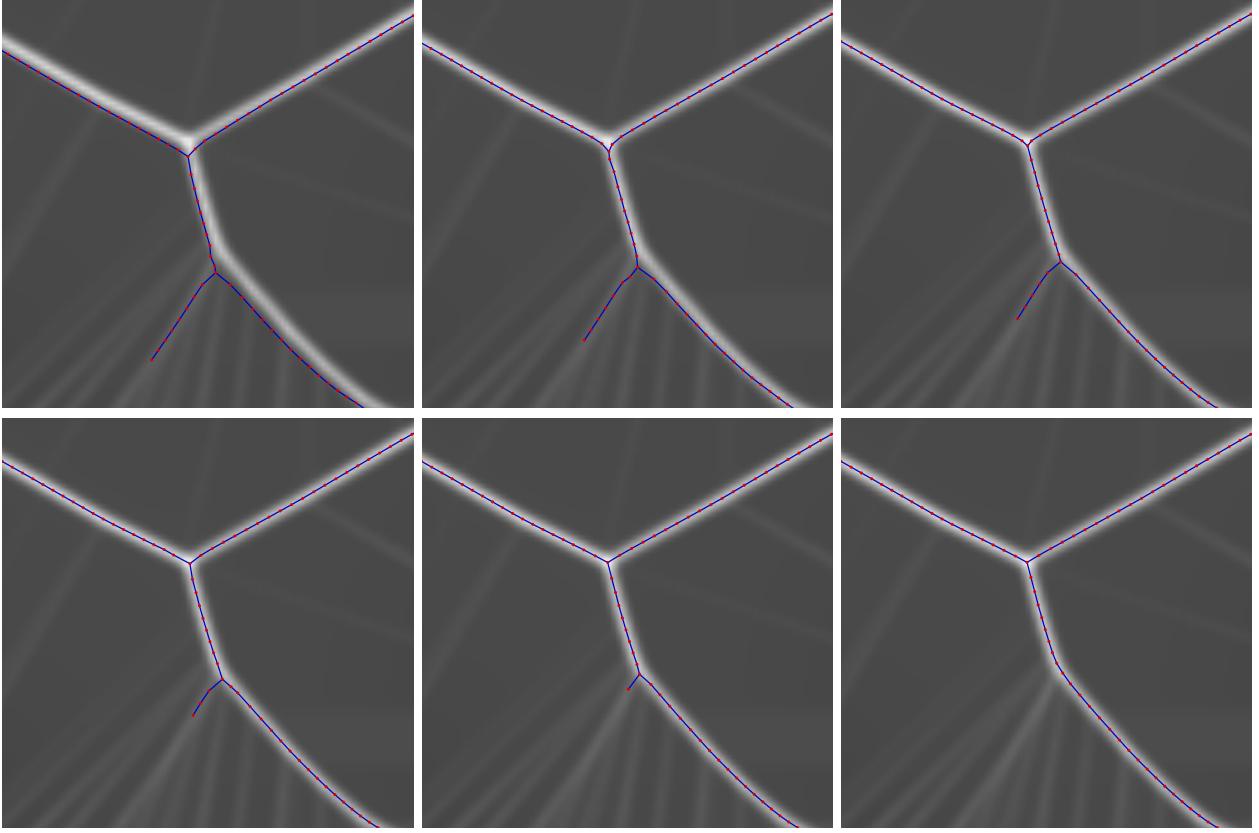


Figure 6.24: Evolution of skeleton $\text{SG}^{t+\tilde{\delta}t}$ on ridgeness map $\text{rdg}(\cdot, t + \tilde{\delta}t)$: branches are moved towards the nearest ridges, while one branch is progressively removed

The regularization force implements the curvature-dependent term in the first line of Eq (6.58), for regular skeleton vertices. We add a smoothing component to the motion of junction vertices, which tends to bring them towards the center of gravity of their neighbors, which has the effect of smoothing the junction. The regularization force for a given vertex v_j is

$$\mathbf{r}_j = \begin{cases} \kappa_j \mathbf{n}_j & \text{if } |\eta_j| = 2 \\ \left(\frac{1}{|\eta_j|} \sum_{v_k \in \eta_j} v_k \right) - v_j & \text{if } |\eta_j| > 2 \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (6.63)$$

The evolution equation at each vertex, after Euler discretization with time step $\delta\tau$, is :

$$v_j^{\tau+1} = v_j^\tau + \delta\tau (\mathbf{a}_j^\tau + \omega_{\text{reg}} \mathbf{r}_j^\tau) \quad (6.64)$$

The forces applied on vertices are illustrated in Fig. 6.23(a). Like the contour, the sampled skeleton is resampled after each iteration of evolution, in order to maintain a regular unit spacing between the vertices. If the distance between two adjacent vertices exceeds 2, a branch point is created between the two, irrespective of the degree of the two vertices. Conversely, if the distance between two adjacent vertices is less than 1, degrees are checked. If the two vertices are branch points, they are merged, the new position of the new vertex being the middle of the two old vertices. If a vertex is terminal, it is deleted. This is consistent with the *a priori* that we have about the deformations of the ridges. Indeed, as the shape contour is smoothed, the ridges of the distance map (and corresponding skeletal branches) are necessarily shortened when they are not removed.

Algorithm 5: Deformable skeleton: evolution, resampling and label propagation

Input:

SG^t : skeleton graph

$D(\cdot, t + \tilde{\delta}t)$: updated distance map

Output:

$\text{SG}^{t+\tilde{\delta}t}$: updated skeleton graph

begin

$\text{SG}^{t+\tilde{\delta}t} := \text{SG}^t$

while $\text{SG}^{t+\tilde{\delta}t}$ is not stable **do**

Discrete deformation

foreach vertex $v_j \in \text{SG}_V^{t+\tilde{\delta}t}$ **do**

 | Calculate a_j and r_j and according to Eqs (6.62) and (6.63)

end

foreach vertex $v_j \in \text{SG}_V^{t+\tilde{\delta}t}$ **do**

 | $v_j := v_j + \delta\tau(a_j + \omega_{\text{reg}}r_j)$

end

Branch shortening

foreach terminal vertex $v_j \in \text{SG}_V^{t+\tilde{\delta}t}$ **do**

 | $v_k :=$ the only neighbor of v_j

 | **if** $\text{rdg}(v_j, t + \tilde{\delta}t) < \text{th}_{\text{rdg}}$ or $\|v_j - v_k\| < 1$ **then**

 | | **if** $|\eta_k| = 3$ **then**

 | | | Update $\phi^{t+\tilde{\delta}t}$ by label propagation (see Fig. 6.23(b))

 | | **end**

 | Delete v_j

end

end

Resampling

foreach non-terminal vertex $v_j \in \text{SG}_V^{t+\tilde{\delta}t}$ **do**

foreach vertex $v_k \in \eta_j$ **do**

 | **if** $\|v_j - v_k\| > 2$ **then**

 | | Add a vertex between v_j and v_k

 | **else if** $\|v_j - v_k\| < 1$ **then**

 | | Merge v_j and v_k

 | **end**

end

end

end

return $\text{SG}^{t+\tilde{\delta}t}$

end

Label propagation

When a terminal vertex v_j is deleted, either by resampling or by removing non-ridge points, the degree of its adjacent vertex v_k is checked. If it is higher than 2, it means v_k is a junction vertex, and that branch (v_j, v_k) disappears. If the degree of v_k is equal to 2 after removal of v_j , the two remaining branches starting from v_k actually form a single branch. The edges of the same branch must have the same labels, thus, propagation of labels is performed, as shown in Figure 6.23(b). The labels of the two old branches are added to the edges of the merged branch.

The modification of labeling $\phi^{t+\tilde{\delta}t}$ can be formulated as follows. We drop superscript $t + \tilde{\delta}t$ for notational simplicity, and denote by v_a and v_b the two other neighbors of junction vertex v_k , before the removal of v_j . In the two paths, starting from v_k , to the nearest junction or terminal vertices, labeling ϕ is updated so that all edges in these two paths are equally labeled. For every edge e in the two paths,

$$\phi(e) := \phi(\{v_k, v_a\}) \cup \phi(\{v_k, v_b\}).$$

The nested evolution of the sampled skeleton, including resampling, terminal vertex deletion, and label propagation is detailed in Algorithm 5.

6.4.5 Weighting the initial skeleton graph

Algorithm 6: Generation of the weighted skeleton graph

Input:

$Se = (SG^{t_0}, SG^{t_2}, \dots, SG^{t_{n_{skel}-1}})$: sequence of skeleton graphs

Output:

(SG^{t_0}, wb) : weighted initial skeleton graph

begin

```

 $\mathcal{Q} := \emptyset$ 
for  $i := n_{skel} - 1$  to 1 with step  $-1$  do
    foreach edge  $e \in SG_E^{t_i}$  do
        if  $\phi^{t_i}(e) \cap \mathcal{Q} = \emptyset$  then
            foreach edge  $f \in SG_E^{t_0}$  do
                if  $\phi^{t_0}(f) \cup \phi^{t_i}(e) \neq \emptyset$  then
                     $\phi^{t_0}(f) := \phi^{t_i}(e)$ 
                     $wb(f) := \frac{t_i}{t_{n_{skel}-1}}$ 
                end
            end
             $\mathcal{Q} := \mathcal{Q} \cup \phi^{t_i}(e)$ 
        end
    end
end

```

The successive smoothing operations of contour $(q_i)_i$ and updates of the skeleton graph generate a sequence of n_{skel} skeleton graphs,

$$Se = (SG^{t_0}, SG^{t_1}, \dots, SG^{t_{n_{skel}-1}}), \quad (6.65)$$

where $t_0 = 0$. Each element SG^{t_i} in this sequence was generated by an instance of Algorithm 5. The weighted skeleton graph is SG^0 , endowed with a weight wb on edges,

$$wb : SG_E^0 \longrightarrow [0, 1].$$

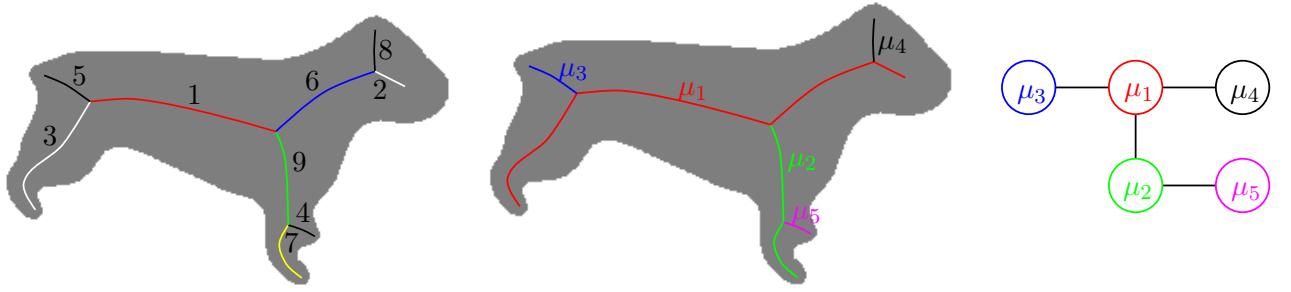


Figure 6.25: From the skeleton graph to the hierarchical skeleton: (left) skeleton with branches, (center) skeleton with BSs (right) hierarchical skeleton

Edges with the same weight and labeling will be gathered in one BS in the hierarchical skeleton. Weights and BS labels are integrated into the initial skeleton graph SG^{t_0} by backpropagating labels ϕ through time. The sequence of skeleton graphs is scanned in reverse-time order. Each set of edges with the same labeling will form a BS. More precisely, each time a new labeling is met at a given time t_i , it means that the BS with this labeling persisted until time t_i . Thus, all edges in SG^0 , belonging to this BS, receive this labeling and are assigned weight

$$wb := \frac{t_i}{t_{n_{\text{ske}1}-1}},$$

which is the relative time of disappearance. As an example, consider Fig. 6.25 (left and center). Early in the backpropagation, an edge labeled $\{1, 2, 3, 6\}$ will be met in the skeleton graph SG^{t_i} . Thus, all edges in SG^0 , either labeled $\{1\}$, $\{2\}$, $\{3\}$ or $\{6\}$ are gathered into the first BS created, μ_1 , with label $\{1, 2, 3, 6\}$. Labels in subset $\{1, 2, 3, 6\}$ will not be considered again in the next iterations. Labelings met in sequence \mathcal{S}_e will create the following BSs:

$$\begin{aligned}\phi = \{1, 2, 3, 6\} &\rightarrow \mu_1 \\ \phi = \{7, 9\} &\rightarrow \mu_2 \\ \phi = \{5\} &\rightarrow \mu_3 \\ \phi = \{8\} &\rightarrow \mu_4 \\ \phi = \{4\} &\rightarrow \mu_5.\end{aligned}$$

Note that the first edges met in the backpropagation, which form μ_1 in the present example, will be assigned weight 1. The weighting procedure is detailed in Algorithm 6. Processed labels are stored in set \mathcal{Q} , which ensures that labels of BSs that were already met are not propagated again.

Finally, algorithm 7 describes the overall process to obtain the weighted skeleton from the initial discrete shape. Fig. 6.26 illustrates the weighted skeleton, with weight wb shown for each BS. Discretization artifacts prevent us from proving formally that weighting is invariant to affine transformations of input shape $\hat{\Omega}$. At least, we observe the stability of BS weights under angle-preserving affine transformations (in the present cases, rotation and scaling).

6.5 Matching hierarchical skeletons

6.5.1 The hierarchical skeleton

The hierarchical skeleton SH is easily built from the weighted skeleton (SG^0, wb) . As illustrated in Fig. 6.25(center and right), it contains BSs and connections between them in a higher representation level than the skeleton graph SG^0 . The hierarchical skeleton is an undirected Attributed Relational Graph (ARG) (SH_V, SH_E, wb) , where

- SH_V is the set of V vertices. A vertex $\mu_m, 1 \leq m \leq V$ corresponds to a BS, which is valued with the

Algorithm 7: Overall process: generation of weighted skeleton graph from input discrete shape

Input: $\hat{\Omega}(0) \subset \hat{\mathcal{D}}$: input discrete shape**Output:** (SG^0, wb) : weighted skeleton graph**begin** Generate discrete skeleton $\hat{S}(0)$ from $\hat{\Omega}(0)$ (Section 6.1.4) Generate explicit contour $(\mathbf{q}_i^0)_i$ from $\hat{\Omega}(0)$ Generate sampled skeleton SG^0 from $\hat{S}(0)$ $t := 0$ $t_{last} := 0$ $Se := (SG^0)$ **while** $IQ((\mathbf{q}_i^t)_i) < 1 - \epsilon$ (Eq. (6.54)) **do** **while** $\max_i \|\mathbf{q}_i^t - \mathbf{q}_i^{t_{last}}\| < 1$ **do** *# Smooth the shape contour* **foreach** curve point \mathbf{q}_i **do** $\mathbf{q}_i^{t+1} := \mathbf{q}_i^t + \delta t \left(\kappa_i^t - \frac{2\pi}{\hat{L}^t} \right) \mathbf{n}_i^t$ **end** Resample discrete curve $(\mathbf{q}_i^t)_i$ $t := t + 1$ **end** Generate discrete shape $\hat{\Omega}(t)$ from $(\mathbf{q}_i^t)_i$ Update $D(\cdot, t)$ from $\hat{\Omega}(t)$ Evolve SG^t from $SG^{t_{last}}$ and $D(\cdot, t)$ (Algorithm 5) Add SG^t into Se $t_{last} := t$ **end** Generate weighted skeleton graph (SG^0, wb) from sequence Se (Algorithm 6)**end**

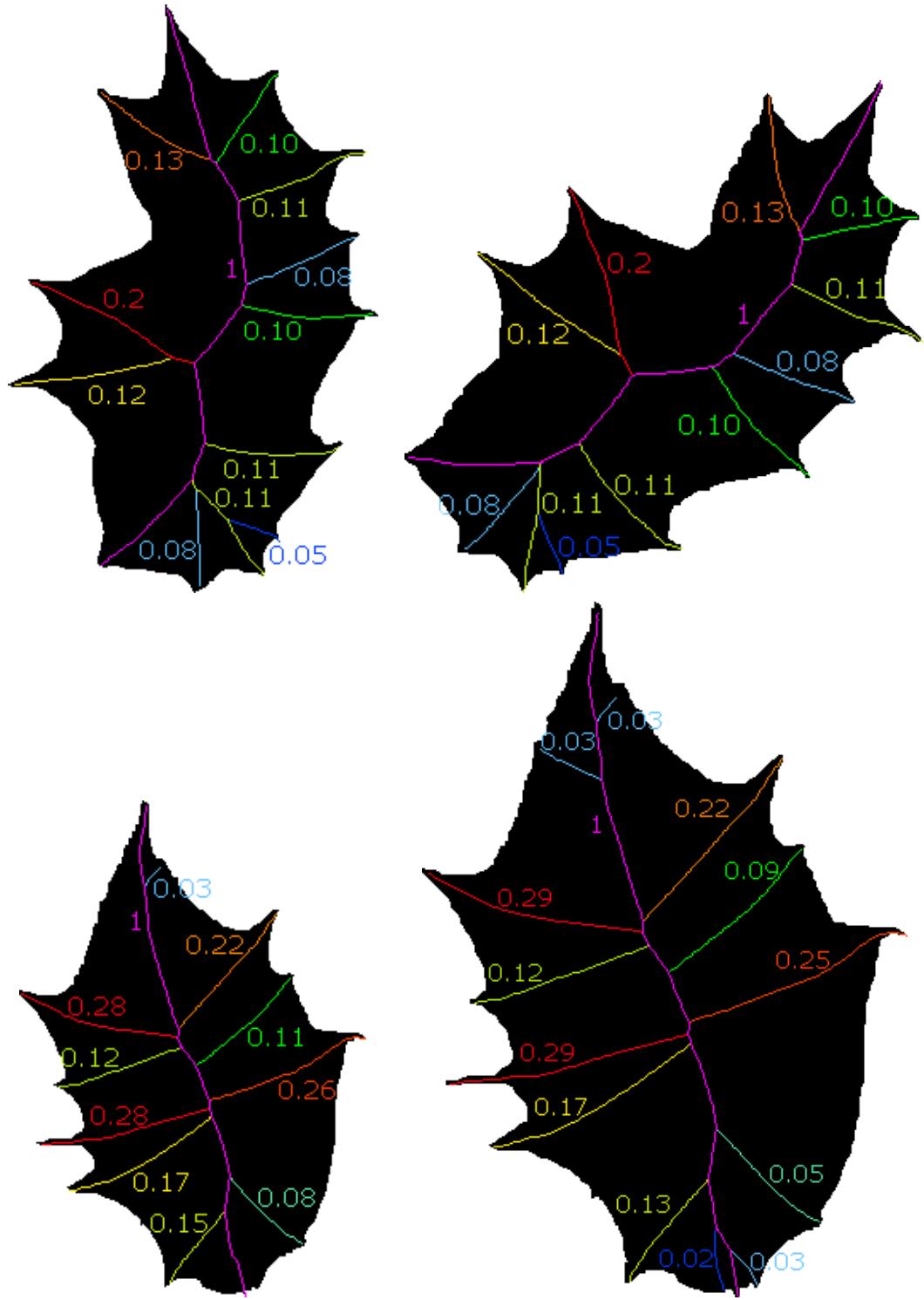


Figure 6.26: Stability of the weighted skeleton under angle-preserving affine transformation of the initial discrete shape: (top) rotation and (bottom) scaling

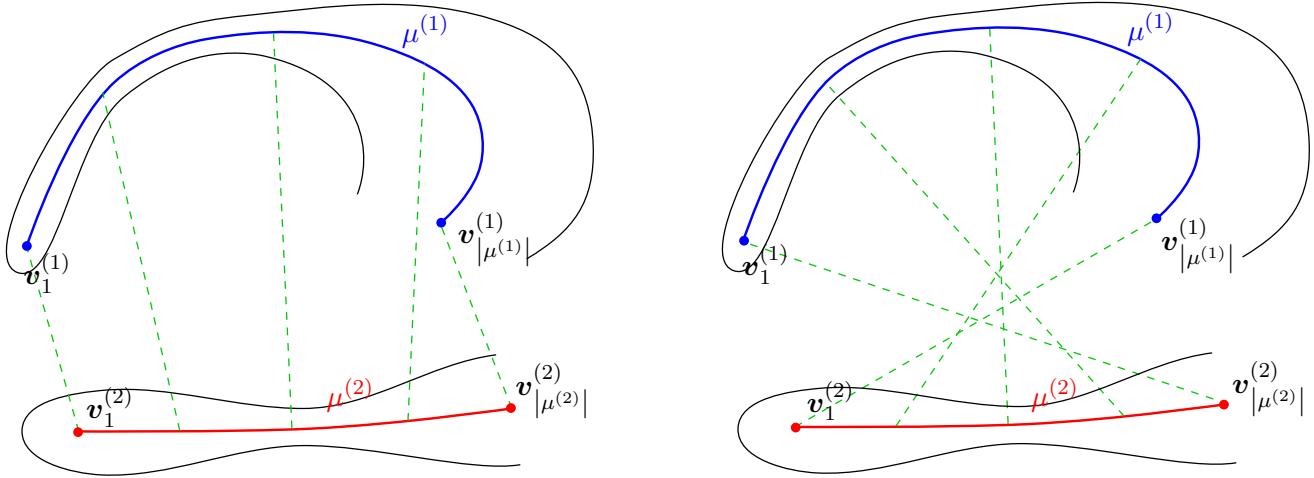


Figure 6.27: In the geometric dissimilarity, the sum of distance deviations is calculated for the two travel directions

sequence of skeleton points ($v_{m,1}, v_{m,2}, \dots, v_{m,|\mu_m|}$). From now on, the term *vertex* will refer to an element of SH_V , whereas *skeleton point* will refer to a vertex in the lower-level representation SG.

- SH_E is the set of undirected edges (two-sets of vertices), representing connections between BSs. An edge $\{\mu_a, \mu_b\}$ exists if BSs μ_a and μ_b have at least one common junction skeleton point in SG^0 .
- wb is the BS weighting function,

$$wb : \text{SH}_V \longrightarrow [0, 1],$$

which was formerly a function of edges in the lower-level representation SG. Since all edges between skeleton points in a given BS have the same weight, wb can be naturally reconsidered as a function of vertices.

Note that $|\mu_m|$ is the size of BS μ_m , i.e. the number of skeleton points it contains, which should be distinguished from the Euclidean length of μ_m ,

$$L(\mu_m) = \sum_{i=1}^{|\mu_m|-1} \|v_{m,i+1} - v_{m,i}\|.$$

The matching of two ARGs consists in establishing a vertex-to-vertex correspondence, in order to minimize a matching cost. The proposed matching methods borrows aspects from the Path-Similarity Skeleton-based method [Bai and Latecki, 2008], as well as the Nested Assignment Structure [Kim et al., 2010]. It extends several elements of these methods, incorporating the novel BS importance weight wb in the matching procedure. Note that the matching itself is not the final purpose. It is rather the computation of a dissimilarity measure between shapes.

6.5.2 Matching costs between vertices

Let $\text{SH}^{(1)}$ and $\text{SH}^{(2)}$ be the two hierarchical skeletons to be matched. It is assumed that one of them is the skeleton of the query shape, whereas the other one belongs to the representatives of the tested shape category. Before performing the matching in the strict sense of the term, we need to calculate a matching cost between each vertex of $\text{SH}^{(1)}$ and each vertex of $\text{SH}^{(2)}$. This cost should take into account

- the attributes/intrinsic properties of vertices: geometry of BSs, and importance wb of BSs,
- the context/local structure of vertices, i.e. the adjacency relations contained in the edges.

Intrinsic dissimilarity between two vertices

Let $\mu^{(1)}$ be a BS of $\text{SH}^{(1)}$ and $\mu^{(2)}$ a BS of $\text{SH}^{(2)}$. Recall that $D(\mathbf{x}, 0)$ is the value of Euclidean distance map at point $\mathbf{x} \in \mathcal{D}$. The time variable will be dropped, as we work on the initial shapes. We introduce the normalization factor, which can be indifferently the square-rooted area,

$$\aleph = \sqrt{|\hat{\Omega}|},$$

or the average distance,

$$\aleph = \frac{1}{|\hat{\Omega}|} \sum_{\mathbf{x} \in \hat{\Omega}} D(\mathbf{x}).$$

Normalizing lengths and distances by \aleph ensures invariance to scaling. The *geometric dissimilarity* between two BSs $\mu^{(1)}$ and $\mu^{(2)}$ is inspired from the path distance introduced in [Bai and Latecki, 2008] and used subsequently in [Xu et al., 2009, Shen et al., 2013]. According to [Bai and Latecki, 2008], it achieves good results in the presence of articulation, stretching and deformation of the contour.

The geometric dissimilarity is quantified by summing the poinwise squared differences between normalized distances, along $\mu^{(1)}$ and $\mu^{(2)}$. Comparing distances instead of positions allows invariance to angle-preserving affine transformations, and a certain robustness to articulated deformations. We perform a linear match, in the sense that the two BSs are traversed at constant speed. The two BSs will generally have different sizes. Hence, the biggest BS is traversed with a unit step, whereas the shortest one is traversed with step < 1 . The sum of distance deviations is calculated for the two travel directions, as depicted in Fig. 6.27. The travel direction leading to the minimal deviation is favored. Assuming that $|\mu^{(1)}| \geq |\mu^{(2)}|$, the geometric dissimilarity is

$$\begin{aligned} \Xi_r(\mu^{(1)}, \mu^{(2)}) &= \frac{1}{|\mu^{(1)}|} \min \left\{ \sum_{i=1}^{|\mu^{(1)}|} \left(\frac{D^{(1)}(\mathbf{v}_i^{(1)})}{\aleph^{(1)}} - \frac{D^{(2)}(\mathbf{v}_j^{(2)})}{\aleph^{(2)}} \right)^2, \sum_{i=1}^{|\mu^{(1)}|} \left(\frac{D^{(1)}(\mathbf{v}_i^{(1)})}{\aleph^{(1)}} - \frac{D^{(2)}(\mathbf{v}_k^{(2)})}{\aleph^{(2)}} \right)^2 \right\} \\ &\text{with } j = (i-1) \frac{|\mu^{(2)}|-1}{|\mu^{(1)}|-1} + 1, \\ &\text{and } k = (|\mu^{(1)}|-i-1) \frac{|\mu^{(2)}|-1}{|\mu^{(1)}|-1} + 1. \end{aligned} \tag{6.66}$$

Since j is real, indexing by j should be interpreted as interpolation between skeleton points $\mathbf{v}_{\lfloor j \rfloor}$ and $\mathbf{v}_{\lceil j \rceil}$, as

$$\mathbf{v}_j^{(2)} = (\lceil j \rceil - j) \mathbf{v}_{\lfloor j \rfloor}^{(2)} + (j - \lfloor j \rfloor) \mathbf{v}_{\lceil j \rceil}^{(2)},$$

and similarly for $\mathbf{v}_k^{(2)}$. There are in fact two linear interpolations: one for calculating \mathbf{v}_j , and one for estimating $D^{(1)}$ and $D^{(2)}$ at real coordinates.

We denote by $\Xi_l(\mu^{(1)}, \mu^{(2)})$ the absolute difference between normalized lengths between two BSs,

$$\Xi_l(\mu^{(1)}, \mu^{(2)}) = \left| \frac{L(\mu^{(1)})}{\aleph^{(1)}} - \frac{L(\mu^{(2)})}{\aleph^{(2)}} \right|. \tag{6.67}$$

The intrinsic dissimilarity, or *branch dissimilarity*, is a weighted average of geometric dissimilarity and length difference,

$$\text{bd}(\mu^{(1)}, \mu^{(2)}) = \alpha \Xi_r(\mu^{(1)}, \mu^{(2)}) (1 - \alpha) \Xi_l(\mu^{(1)}, \mu^{(2)}), \tag{6.68}$$

where weight $\alpha \in [0, 1]$ is discussed in Section 6.5.5.

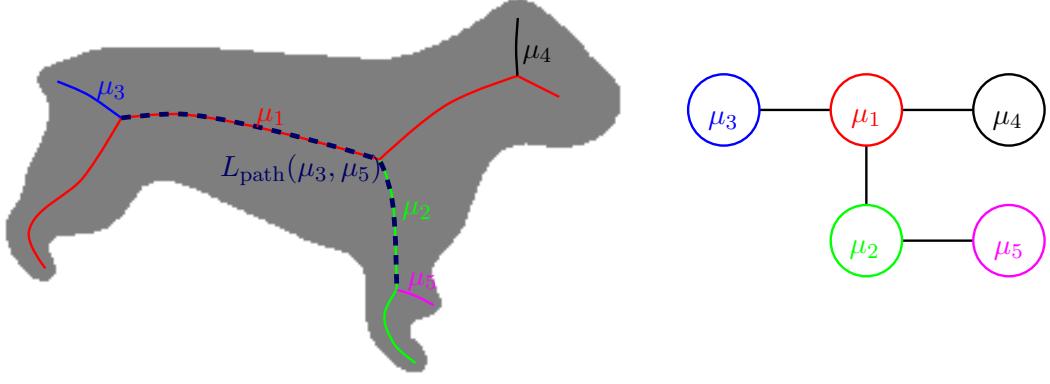


Figure 6.28: Shortest path of skeleton points between BSs μ_3 and μ_5 (dark dashed blue curve).

Context dissimilarity between two vertices

To integrate the context into the matching cost between two BSs $\mu_a^{(1)}$ and $\mu_b^{(2)}$, we associate $\mu_a^{(1)}$ (respectively, $\mu_b^{(2)}$) to each of the other BSs of $\text{SH}^{(1)}$ (respectively, $\text{SH}^{(2)}$). In this purpose, we introduce notation $\mu_a \succ \mu_i$, to represent the association of two given BSs, indexed by a and i , in the same hierarchical skeleton.

The context of a BS μ_a is the "relative position" of μ_a with respect to the other BSs of SH. This relative position is encoded using the lengths of the shortest paths between $\mu_a^{(1)}$ and the other BSs. In our context, a shortest path between two BSs, illustrated in Fig. 6.28, is a sequence of skeleton points linking the two nearest junction points of the respective BSs. Since we are interested in the length of this path, we insist on the fact that it should not be understood as an ordered sequence of edges in SH_E , but rather as a path in the former skeleton graph SG^0 . This path travels along branches of SG^0 , or, equivalently, portions of BSs. Since we only deal with simply-connected shapes, SG^0 and SH do not contain any cycle. The shortest path is therefore unique.

Let $L_{\text{path}}(\mu_a, \mu_i)$ be the length of the shortest path between two BSs μ_a and μ_i . Trivially, $L_{\text{path}}(\mu_a, \mu_i) = 0$ if $a = i$, or if BSs μ_a and μ_i are adjacent in SH. An association between two BSs is attributed with the length of the shortest path between them. Given two associations $\mu_a^{(1)} \succ \mu_i^{(1)}$ and $\mu_b^{(2)} \succ \mu_j^{(2)}$ in hierarchical skeletons $\text{SH}^{(1)}$ and $\text{SH}^{(2)}$, respectively, the absolute difference between the lengths of corresponding shortest paths,

$$\left| L_{\text{path}}(\mu_a^{(1)}, \mu_i^{(1)}) - L_{\text{path}}(\mu_b^{(2)}, \mu_j^{(2)}) \right|,$$

is the context dissimilarity of these two associations. It is used in the matching cost explained below.

Calculating the matching cost using EMD

Context dissimilarity is computed on all associations involving all vertices. Thus, for any pair of vertices $(\mu_a^{(1)}, \mu_b^{(2)}) \in \text{SH}_V^{(1)} \times \text{SH}_V^{(2)}$, there are $V^{(1)}V^{(2)}$ context dissimilarity values computed. We wish to obtain only one value representing the matching cost between the two vertices, taking both geometric and context dissimilarities into account. Since we are dealing with the matching cost of a given pair $(\mu_a^{(1)}, \mu_b^{(2)})$, indices a and b are fixed in this subsection.

This matching cost can be viewed as the cost of the solution to a transportation problem. This solution is calculated using the Earth Mover's Distance (EMD) [Rubner et al., 2000, Kim et al., 2010]. Let us consider two discrete non-normalized distributions⁸, which do not necessarily sum to 1. The first one is viewed as a set

⁸When handling actual distributions, that sum to 1, the EMD is the discrete counterpart of the Wasserstein distance addressed in literature on optimal transport, e.g. [Cuturi, 2013, Ferradans et al., 2014]

of heaps of earth, each having a certain volume in some space, and the second one as a set of holes of greater or lesser volume in the same space. The EMD allows to compute the smallest amount of work necessary to fill the holes with all the heaps of earth, taking into account the volume of displaced earth, as well as distance to be traveled from the source heap to the destination. This distance is referred to as *ground distance*. In our case, the EMD should take the context of BSs into account. Hence, the ground distance is subsequently built from dissimilarities between associations. Each earth heap corresponds to an association $\mu_a^{(1)} \succ \mu_i^{(1)}$, whereas each hole corresponds to an association $\mu_b^{(2)} \succ \mu_j^{(2)}$. The volume of earth heaps (respectively, holes) is represented by weights $\text{wb}(\mu_i^{(1)})$ (respectively, $\text{wb}(\mu_j^{(2)})$).

The sequence made up of all associations involving $\mu_a^{(1)}$, endowed with corresponding weights,

$$\{(\mu_a^{(1)} \succ \mu_1^{(1)}, \text{wb}(\mu_1^{(1)})), \dots, (\mu_a^{(1)} \succ \mu_{V^{(1)}}^{(1)}, \text{wb}(\mu_{V^{(1)}}^{(1)}))\},$$

is the *signature* of vertex $\mu_a^{(1)}$. Similarly,

$$\{(\mu_b^{(2)} \succ \mu_1^{(2)}, \text{wb}(\mu_1^{(2)})), \dots, (\mu_b^{(2)} \succ \mu_{V^{(2)}}^{(2)}, \text{wb}(\mu_{V^{(2)}}^{(2)}))\}$$

is the signature of $\mu_b^{(2)}$. These signatures contain the weight of vertices, corresponding to the volume of earth to be transported. The dissimilarity between associations $\mu_a^{(1)} \succ \mu_i^{(1)}$ and $\mu_b^{(2)} \succ \mu_j^{(2)}$ is defined as follows:

$$\begin{aligned} \psi(\mu_a^{(1)} \succ \mu_i^{(1)}, \mu_b^{(2)} \succ \mu_j^{(2)}) &= \beta \left(\text{bd}(\mu_a^{(1)}, \mu_b^{(2)}) + \text{bd}(\mu_i^{(1)}, \mu_j^{(2)}) \right) \\ &\quad + (1 - \beta) \left| L_{\text{path}}(\mu_a^{(1)}, \mu_i^{(1)}) - L_{\text{path}}(\mu_b^{(2)}, \mu_j^{(2)}) \right| \end{aligned} \quad (6.69)$$

where bd is the intrinsic dissimilarity between two vertices, defined in Eq. (6.68), and $L_{\text{path}}(\cdot, \cdot)$ the length of the shortest path between two vertices. This dissimilarity represents the ground distance between association $\mu_a^{(1)} \succ \mu_i^{(1)}$ and the hole corresponding to the association $\mu_b^{(2)} \succ \mu_j^{(2)}$. Since indices a and b are fixed, from now on, we shorten $\psi(\mu_a^{(1)} \succ \mu_i^{(1)}, \mu_b^{(2)} \succ \mu_j^{(2)})$ to $\psi_{i,j}$.

To determine the amount of work necessary for transportation, we should calculate a flow $\mathbf{f} = (f_{i,j})_{i=1 \dots V^{(1)}, j=1 \dots V^{(2)}}$, where $(f_{i,j})$ is the amount of earth transported from association $\mu_a^{(1)} \succ \mu_i^{(1)}$ to association $\mu_b^{(2)} \succ \mu_j^{(2)}$. Since we handle non-normalized distributions, the amount of earth can be different from the capacity of holes, and thus the optimal transportation will be, most of the time, partial. The optimal flow is the solution of the following linear problem:

$$\mathbf{f}^* = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \psi_{i,j} f_{i,j},$$

subject to the linear constraints,

$$\begin{aligned} f_{i,j} &\geq 0 && \forall i \in \{1, \dots, V^{(1)}\}, \forall j \in \{1, \dots, V^{(2)}\} \\ \sum_{j=1}^{V^{(2)}} f_{i,j} &\leq \text{wb}(\mu_i^{(1)}) && \forall i \in \{1, \dots, V^{(1)}\} \\ \sum_{i=1}^{V^{(1)}} f_{i,j} &\leq \text{wb}(\mu_j^{(2)}) && \forall j \in \{1, \dots, V^{(2)}\} \\ \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} f_{i,j} &= \min \left(\sum_{i=1}^{V^{(1)}} \text{wb}(\mu_i^{(1)}), \sum_{j=1}^{V^{(2)}} \text{wb}(\mu_j^{(2)}) \right). \end{aligned} \quad (6.70)$$

Naturally, the flow is positive (constraint 1). The summed flow coming from a given heap, indexed by i , should not exceed its available quantity (constraint 2). Similarly, the summed flow arriving at a given hole, indexed by j , should not exceed its capacity (constraint 3). The total flow is the minimum between the total quantity in heaps and the total capacity of holes (constraint 4). Note that the sums of weights, in the right-hand side of constraint 4, are independent of the considered vertex pair $(\mu_a^{(1)}, \mu_b^{(2)})$. Since we have a linear objective function of real variables, and linear constraints, the optimal flow can be found using the Simplex algorithm [Spielman and Teng, 2004]. Once the optimal flow is determined, the *raw matching cost* between the two vertices is

$$\text{emd}_{a,b} = \frac{\sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \psi_{i,j} f_{i,j}^*}{\sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} f_{i,j}^*}, \quad (6.71)$$

which, according to constraint 4, can be rewritten as

$$\text{emd}_{a,b} = \frac{\sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \psi_{i,j} f_{i,j}^*}{\min \left(\sum_{i=1}^{V^{(1)}} \text{wb}(\mu_i^{(1)}), \sum_{j=1}^{V^{(2)}} \text{wb}(\mu_j^{(2)}) \right)}.$$

The denominator being the same for all pairs (a, b) , this normalization could be ignored for subsequent steps. We however keep the original approach that was developed in Aurélie's PhD, and leave discussions about this point for the perspective in Section 6.6.2.

Intuitively, a given BS should be matched to one with a similar importance. Moreover, BSs with high importance should be matched in priority. In other words, the general appearance of the shape should be favored over the details. Hence, we introduce a penalty function, which simultaneously increases the cost when the importances of vertices are different and decreases it when the two vertices have high importances. The *penalized matching cost* is obtained by multiplying the raw matching cost by this penalty:

$$C_{a,b} = \text{emd}_{a,b} \left(\exp \left(2|\text{wb}(\mu_a^{(1)}) - \text{wb}(\mu_b^{(2)})| \right) + (2 - \text{wb}(\mu_a^{(1)}) - \text{wb}(\mu_b^{(2)})) \right) \quad (6.72)$$

6.5.3 Matching the two hierarchical skeletons using the Hungarian algorithm

In this part, we consider that the penalized matching cost $C_{a,b}$ of Eq. (6.72) has been computed for every pair $(\mu_a^{(1)}, \mu_b^{(2)})$, with $a = 1 \dots V^{(1)}$ and $b = 1 \dots V^{(2)}$. The final matching, *i.e.* assigning each vertex of $\text{SH}^{(1)}$ to only one vertex of $\text{SH}^{(2)}$, is formulated as a linear assignment problem in a bipartite graph. The considered bipartite graph is made up of the vertices of $\text{SH}_V^{(1)}$ on one side, and the vertices of $\text{SH}_V^{(2)}$ on the other side. The optimal matching \mathbf{M}^* , or assignment matrix, is the solution to the following $\{0, 1\}$ linear programming problem,

$$\mathbf{M}^* = \underset{\mathbf{M}}{\operatorname{argmin}} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} m_{i,j} C_{i,j},$$

subject to

$$\begin{aligned} m_{i,j} &\in \{0, 1\} \quad \forall i \in \{1, \dots, V^{(1)}\}, j \in \{1, \dots, V^{(2)}\} \\ \sum_{\substack{j=1 \\ V^{(2)}}} m_{i,j} &= 1 \quad \forall i \in \{1, \dots, V^{(1)}\} \\ \sum_{\substack{i=1 \\ V^{(1)}}} m_{i,j} &= 1 \quad \forall j \in \{1, \dots, V^{(2)}\} \end{aligned}$$

One may note that a necessary condition for these constraints to be satisfied is $V^{(1)} = V^{(2)}$. In most situations in our shape matching task, this will not be the case. Thus, it is necessary to adapt the data.

Consider that penalized matching costs are gathered in matrix \mathbf{C} , each row (respectively, column) being assigned to a vertex of $SH^{(1)}$ (respectively, $SH^{(2)}$). Let $V_{\max} = \max(V^{(1)}, V^{(2)})$ be the maximum number of vertices between $SH^{(1)}$ and $SH^{(2)}$. If $V^{(1)} < V^{(2)}$ (respectively $V^{(2)} < V^{(1)}$) we add as many rows (respectively, columns) as necessary to get a square matrix $\bar{\mathbf{C}}$. These rows or columns are filled with a *ghost* value, representing the cost of leaving the vertex unassigned.

Given the square cost matrix, the linear assignment problem can be solved by the Hungarian algorithm, also known as Kuhn-Munkres algorithm [Kuhn, 1955, Munkres, 1957] and its numerous variants [Tomizawa, 1971, Jonker and Volgenant, 1987][Lawler, 1976, Chapter 5]. While the original Kuhn-Munkres algorithm operates in the worst case in $O(V_{\max}^4)$, the improvement in [Jonker and Volgenant, 1987] is in $O(V_{\max}^3)$.

We set the ghost value to the mean of the entries of \mathbf{C} , as in [Bai and Latecki, 2008], so that leaving a vertex unassigned has an average cost. For example, if $V^{(2)} = V^{(1)} - 2$, the square matrix of order V_{\max} is thus

$$\bar{\mathbf{C}} = \begin{bmatrix} C_{1,1} & \dots & C_{1,V^{(2)}} & \text{avg}(\mathbf{C}) & \text{avg}(\mathbf{C}) \\ C_{2,1} & \dots & C_{2,V^{(2)}} & \text{avg}(\mathbf{C}) & \text{avg}(\mathbf{C}) \\ \dots & \dots & \dots & \text{avg}(\mathbf{C}) & \text{avg}(\mathbf{C}) \\ C_{V^{(1)},1} & \dots & C_{V^{(1)},V^{(2)}} & \text{avg}(\mathbf{C}) & \text{avg}(\mathbf{C}) \end{bmatrix}.$$

Algorithm 8: Overall matching procedure

Input:
 $SH^{(1)}, SH^{(2)}$: hierarchical skeletons
Output:
 $\bar{\mathbf{C}}$: square matrix of penalized costs
 \mathbf{M} : assignment matrix
begin
 # Construction of \mathbf{C} using the EMD
 for $i := 1 \dots V^{(1)}$ **do**
 for $j := 1 \dots V^{(2)}$ **do**
 | Calculate $C_{i,j}$ using EMD and penalty function (Eq. (6.72))
 end
 end
 # Matching of vertices using Hungarian algorithm on \mathbf{C}
 $\bar{\mathbf{C}} := \text{squareMatrix}(\mathbf{C})$
 $\mathbf{M} := \text{hungarianAlgorithm}(\bar{\mathbf{C}})$
end

For the details of the assignment algorithm, we refer the reader to [Jonker and Volgenant, 1987]. The whole process to generate the matching is summarized in Algorithm 8. Once the assignment \mathbf{M} is determined, the

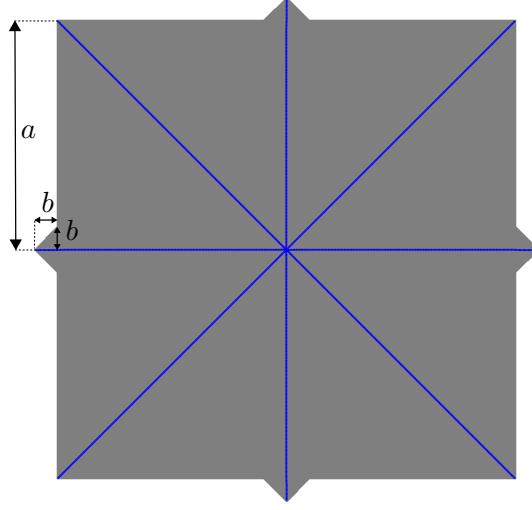


Figure 6.29: An example of shape where the "length" of skeleton is greater than the length of the shape contour

final dissimilarity between the two hierarchical skeletons is calculated from \mathbf{C} and \mathbf{M} , as well as the weight values,

$$\Upsilon(\text{SH}^{(1)}, \text{SH}^{(2)}) = \frac{\sum_{i=1}^{V_{\max}} \sum_{j=1}^{V_{\max}} \bar{\mathbf{C}}_{i,j} m_{i,j} \max(\text{wb}(\mu_i^{(1)}), \text{wb}(\mu_j^{(2)}))}{\sum_{i=1}^{V_{\max}} \sum_{j=1}^{V_{\max}} m_{i,j} \max(\text{wb}(\mu_i^{(1)}), \text{wb}(\mu_j^{(2)}))}, \quad (6.73)$$

Since indices i and j stop at V_{\max} , $\mu_i^{(1)}$ or $\mu_j^{(2)}$ can be a ghost value, whether $V^{(1)} < V^{(2)}$ or the opposite. The weight assigned to a ghost vertex is $\text{wb}(\mu) = 0$. This final dissimilarity is the one used to compare the query shape to other shapes in the database, as mentioned in Section 6.5.5. Note that, by construction, it is symmetric,

$$\Upsilon(\text{SH}^{(1)}, \text{SH}^{(2)}) = \Upsilon(\text{SH}^{(2)}, \text{SH}^{(1)}),$$

since the square cost matrix involved in the calculation of $\Upsilon(\text{SH}^{(1)}, \text{SH}^{(2)})$ is just the transpose of the one of $\Upsilon(\text{SH}^{(2)}, \text{SH}^{(1)})$.

6.5.4 Complexity analysis

We analyze the complexity of the overall process, which gathers the generation of the weighted skeleton graph, summarized in Algorithm 7, and the matching of the hierarchical skeleton in Algorithm 8. We cannot provide an asymptotic order as a function of a single variable representing the "size" of data. Indeed, the complexity involves several data sizes,

- $|\hat{\Omega}|$, the number of pixels in the initial discrete shape $\hat{\Omega}$,
- $|\partial\hat{\Omega}|$, the number of pixels in the discrete inner boundary of $\hat{\Omega}$,
- $|\hat{S}|$, the number of pixels in the discrete skeleton,
- $|\text{SG}_V|$, the number of nodes in the skeleton graph,
- V , the number of BSs in the hierarchical skeleton.

At least, the orders of magnitudes of these different sizes can be sorted,

$$V \ll |\text{SG}_V| \approx |\hat{S}| < |\hat{\Omega}|$$

In addition, we have

$$|\partial\hat{\Omega}| < |\hat{\Omega}|.$$

At the time of writing, we are unaware of any work establishing a relationship between $|\hat{S}|$ and $|\partial\hat{\Omega}|$. In practice, if the shape contour is smooth, it is expected that

$$|\hat{S}| < |\partial\hat{\Omega}|.$$

However, counterexamples can be easily found, as illustrated in Fig. 6.29. The shape has length $8a+8(\sqrt{2}-1)b$, whereas the skeleton has "length"⁹ $4(\sqrt{2}+1)a+4b$. Note that this counterexample is based on a continuous shape Ω and skeleton S , which remains true on a discretized version, as soon as b is large enough, and that the Euclidean "length" is replaced by a number of pixels.

Leaving this counterexample, it can also be noticed that the number of pixels in $\hat{\Omega}$ is bounded above by the squared number of pixels in the inner boundary,

$$|\hat{\Omega}| = O(|\partial\hat{\Omega}|^2),$$

which comes from a discrete interpretation of the isoperimetric quotient IQ, defined in Eq. (6.54).

Complexity of the generation of the weighted skeleton graph

The computation of the initial discrete skeleton, the first step of Algorithm 7, is

$$\underbrace{O(|\hat{\Omega}|)}_{\text{Distance transform}} + \underbrace{O(|\hat{\Omega}| \log |\hat{\Omega}|)}_{\text{Iterative thinning}} = O(|\hat{\Omega}| \log |\hat{\Omega}|)$$

The successive smoothings of shape contour and updates of deformable skeleton generates a sequence of n_{skel} skeleton graphs, referred to in Eq. (6.65). Equivalently, n_{skel} is the number of times that the condition in the inner **while** loop of Algorithm 7 is false. Let us also assume that t_{final} is the number of iterations of the outer **while** loop. The complexity of generating the sequence of skeleton graphs is

$$\underbrace{t_{\text{final}} O(|\partial\hat{\Omega}|)}_{\text{Smoothing}} + n_{\text{skel}} \left(\underbrace{O(|\hat{\Omega}|)}_{\text{Distance transform}} + \underbrace{O(|\hat{S}|)}_{\text{Update skeleton}} \right),$$

which, since $t_{\text{final}} > n_{\text{skel}}$, is bounded above by

$$\begin{aligned} t_{\text{final}} (O(|\partial\hat{\Omega}|) + O(|\hat{\Omega}|) + O(|\hat{S}|)) &= t_{\text{final}} O(\max(|\partial\hat{\Omega}|, |\hat{\Omega}|, |\hat{S}|)) \\ &= t_{\text{final}} O(|\hat{\Omega}|). \end{aligned}$$

To compute the weighted skeleton graph, the algorithm needs to run over every skeleton point of every of the n_{skel} skeletons, and is thus in $n_{\text{skel}} O(|\hat{S}|)$. The entire generation of the weighted skeleton graph is thus bounded above by $t_{\text{final}} O(|\hat{\Omega}|)$.

⁹In case of a continuous skeleton, the "length" $|S|$ is an improper denomination for the 1-dimensional Hausdorff measure $\mathcal{H}^1(S)$. We refer the reader to [Royden and Fitzpatrick, 2010, Chapter 20] for a presentation of Hausdorff measures on metric spaces

Complexity of the matching algorithm

Computing the intrinsic dissimilarity bd , for a given pair $(\mu_a^{(1)}, \mu_b^{(2)})$, is in $O(\max(|\mu_a^{(1)}|, |\mu_b^{(2)}|))$. Computing it for every vertex pair is thus bounded above by

$$\begin{aligned} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} O(\max(|\mu_a^{(1)}|, |\mu_b^{(2)}|)) &= O\left(V^{(1)}V^{(2)} \max\left(\max_{i=1 \dots V^{(1)}} |\mu_i^{(1)}|, \max_{j=1 \dots V^{(2)}} |\mu_j^{(2)}|\right)\right) \\ &= O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right) \end{aligned}$$

Similarly, computing a shortest path between two vertices in an association is bounded above by $O(|\hat{S}|)$. Since it is computed for all associations in $\text{SH}^{(1)}$ on the one hand, and all associations in $\text{SH}^{(2)}$ on the other hand, the computation of all context dissimilarities is in

$$\sum_{a=1}^{V^{(1)}-1} \sum_{i=a+1}^{V^{(1)}} O(|\hat{S}^{(1)}|) + \sum_{b=1}^{V^{(2)}-1} \sum_{j=b+1}^{V^{(2)}} O(|\hat{S}^{(2)}|) = O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right).$$

While the Simplex is a theoretically exponential algorithm, it was shown in [Spielman and Teng, 2004] that it is "in practice" in linear time, $O(\max(V^{(1)}, V^{(2)})) = O(V_{\max})$. Hence, the generation of the penalized cost matrix \mathbf{C} and the matching itself, *i.e.* solving the linear vertex assignment, is in

$$\underbrace{O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right)}_{\text{Intrinsic dis.}} + \underbrace{O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right)}_{\text{Context dis.}} + V^{(1)}V^{(2)} \underbrace{O(V_{\max})}_{\text{One EMD}} + \underbrace{O(V_{\max}^3)}_{\text{Matching}} \\ = O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right) + O(V_{\max}^3)$$

As $\max(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|) > V_{\max}$, the matching complexity is bounded above by

$$O\left(V_{\max}^2 \max\left(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|\right)\right).$$

6.5.5 Experiments

Let \mathcal{B} be the shape database and Ω the *query shape*. To determine the η first shapes in \mathcal{B} closest to Ω , we assume that the hierarchical skeletons of shapes belonging to \mathcal{B} have been previously computed. The η shapes most similar to Ω are sorted by increasing dissimilarity, in the sense of the final dissimilarity defined in Eq. (6.73).

Methods of the literature used for comparison

Path similarity method [Bai and Latecki, 2008] The authors proposed a skeleton matching algorithm based on the comparison of the shortest paths between terminal points, which encode the prominent contour areas (*cf.* Figure 6.30). In other words, the terminal branches can be seen as the visual parts of the original shape.

To match terminal points of two discrete skeletons $\hat{S}^{(1)}$ and $\hat{S}^{(2)}$, they begin by ordering them in clockwise orientation. Shortest paths between all terminal points are computed in $\hat{S}^{(1)}$ and $\hat{S}^{(2)}$. Then, for each pair of terminal points $(t^{(1)}, t^{(2)}) \in \hat{S}^{(1)} \times \hat{S}^{(2)}$, they compute the *path distance* between every path emanating from $t^{(1)}$ and every path emanating from $t^{(2)}$. The path distance is a function of the pointwise sum of differences of normalized distances, as well as the difference between lengths. For each matrix from $(t^{(1)}, t^{(2)})$, they apply the *Optimal Subsequence Bijection (OSB)* algorithm of [Latecki et al., 2007] in order to obtain a dissimilarity value between $t^{(1)}$ and $t^{(2)}$. The OSB method is intended to match the paths from $t^{(1)}$ and $t^{(2)}$, respecting the clockwise ordering of terminal points, while allowing to skip paths if their matching is not relevant.

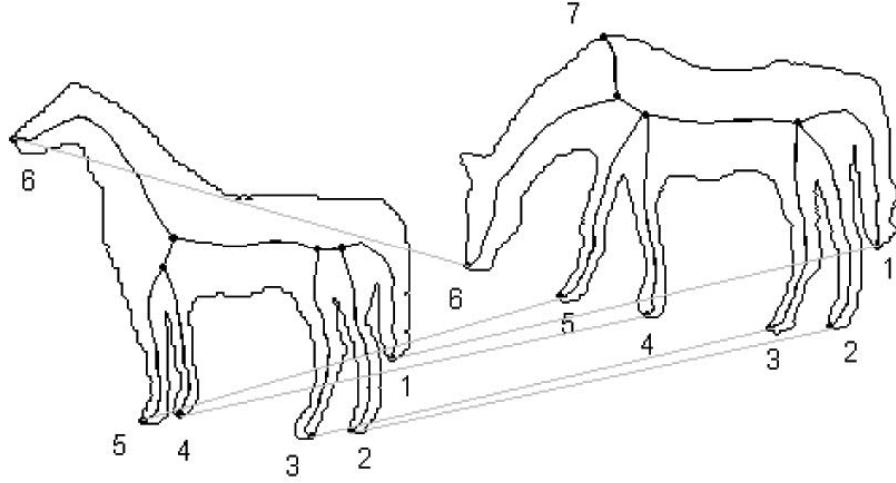


Figure 6.30: Matching of terminal points according to [Bai and Latecki, 2008].

At the end of this step, the authors obtain a dissimilarity value for each pair of terminal points, which they group together in a cost matrix. Then, they apply the Hungarian algorithm on this cost matrix to assign each terminal point in $\hat{S}^{(1)}$ to a terminal point in $\hat{S}^{(2)}$.

Regarding the complexity of the algorithm, let T be the number of terminal skeleton points in a discrete skeleton \hat{S} . With a view to compare the complexity of [Bai and Latecki, 2008] to ours, note that a terminal skeleton point belongs to only one BS, and that one BS can link one or two terminal points, thus $T \geq V$. There are $O(T^2)$ paths between terminal points of a skeleton. Hence, the complexity of computing the path distance between every path in $\hat{S}^{(1)}$ and every path in $\hat{S}^{(2)}$ is bounded above by

$$O\left(T^{(1)2}T^{(2)2} \max(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|)\right) = O\left(T_{\max}^4 \max(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|)\right),$$

where $T_{\max} = \max(T^{(1)}, T^{(2)})$. The complexity of the improved Hungarian algorithm is $O(T_{\max}^3)$. Hence, the total complexity of [Bai and Latecki, 2008] is bounded by

$$O\left(T_{\max}^4 \max(|\hat{S}^{(1)}|, |\hat{S}^{(2)}|)\right),$$

which is, while remaining polynomial, two degrees higher than ours.

Shape context [Belongie et al., 2002] The shape context method is based on the relative spatial distribution (Euclidean distance and orientation) of contour points. It uses a sampling of the contour, $(q_i)_i$, as our discrete evolution scheme described Section 6.4.4. For each sampled point q_i , the authors construct a histogram of the positions of other sampled points, relatively to q_i . It is a 2D log-polar histogram, as it represents the joint distribution of orientations and logarithms of distances. The histogram is then flattened to a feature vector, referred to as the shape context of q_i .

The set of shape contexts for each of them is then used to match the shapes, as illustrated in Figure 6.31. A matching cost is computed between every shape context in $\Omega^{(1)}$ and $\Omega^{(2)}$, in the sense of the χ^2 distance. Let $SC_i^{(1)}$ and $SC_j^{(2)}$ be the shape contexts of points $q_i^{(1)}$ and $q_j^{(2)}$, respectively. Their matching cost is

$$\sum_{b=1}^B \frac{\left(SC_i^{(1)}(b) - SC_j^{(2)}(b)\right)^2}{SC_i^{(1)}(b) + SC_j^{(2)}(b)},$$

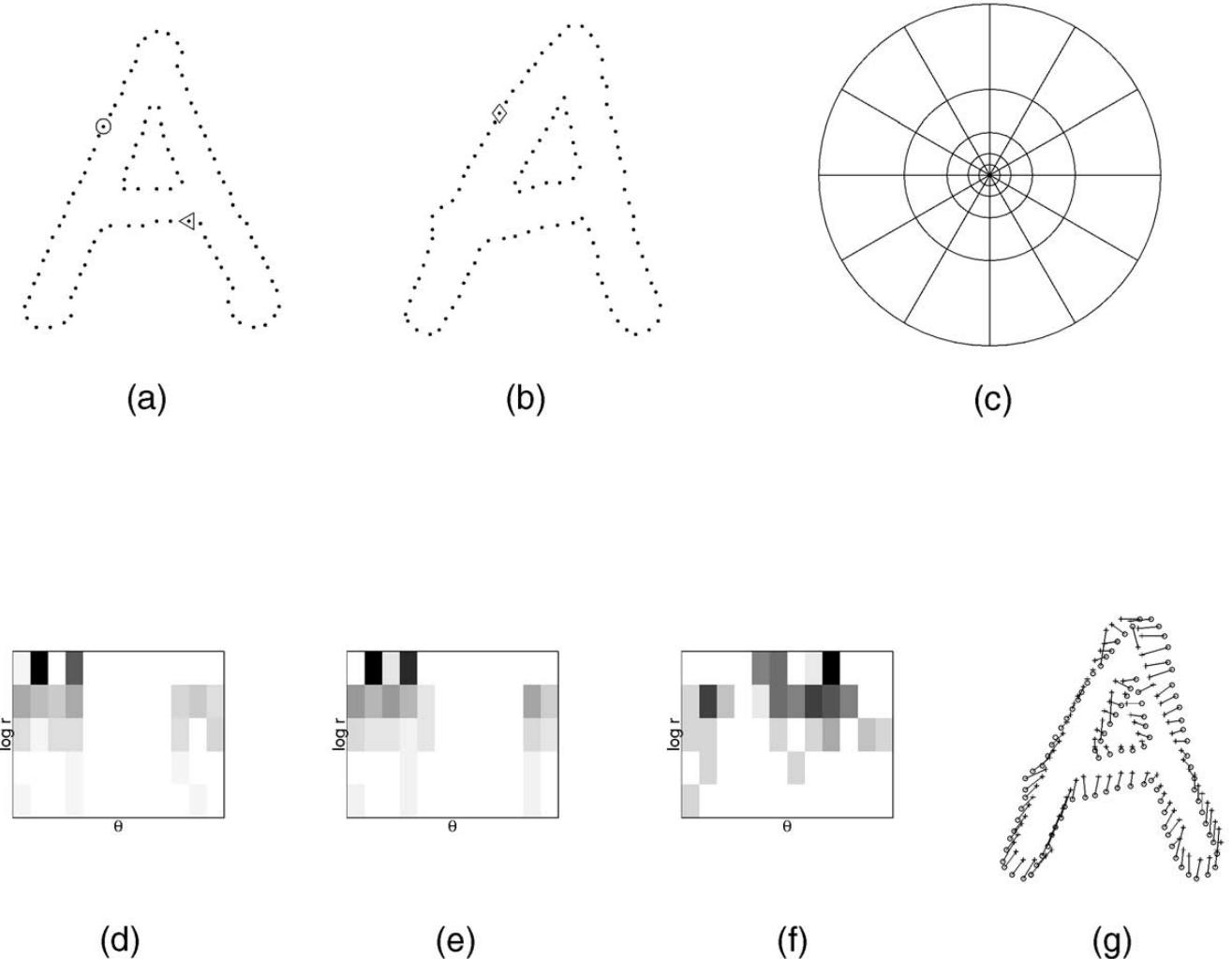


Figure 6.31: Shape context ([Belongie et al., 2002]). (a) and (b) Sampling of the contours of two shapes. (c) Bins of log-polar histograms. (d), (e) and (f) Shape contexts of points marked with symbols \circ , \diamond and \triangleleft on Figures (a) and (b). (g) Pointwise matching between shapes (a) and (b).

where B is the total number of bins in the flattened histogram. As in the previous method, the assignment between contour points is determined by applying the Hungarian algorithm on the obtained cost matrix. The variant of [Jonker and Volgenant, 1987] is used. Denoting by N_{\max} the size of the longest sampled contour, between $\Omega^{(1)}$ and $\Omega^{(2)}$, the assignment is performed in $O(N_{\max}^3)$ operations. This method was chosen since it is also used by [Bai and Latecki, 2008] to evaluate their work. Moreover, this allows us to position our work against a method which is not based on the skeleton.

Experimental results

We carried out our main tests on the Kimia216 dataset [Sebastian et al., 2004], containing 18 classes, each with 12 shapes. An overview of this dataset is shown in Figure 6.32. Each shape was considered as a query, the goal being to find the eleven other shapes of the same category, in a leave-one-out fashion. This is the protocol used in [Bai and Latecki, 2008]. We conduct our experiments in the same setting as them. The three compared dissimilarities - ours, the path similarity method, and the shape context - are symmetric. For each tested method, we thus compute $216 \times 215/2$ dissimilarities.

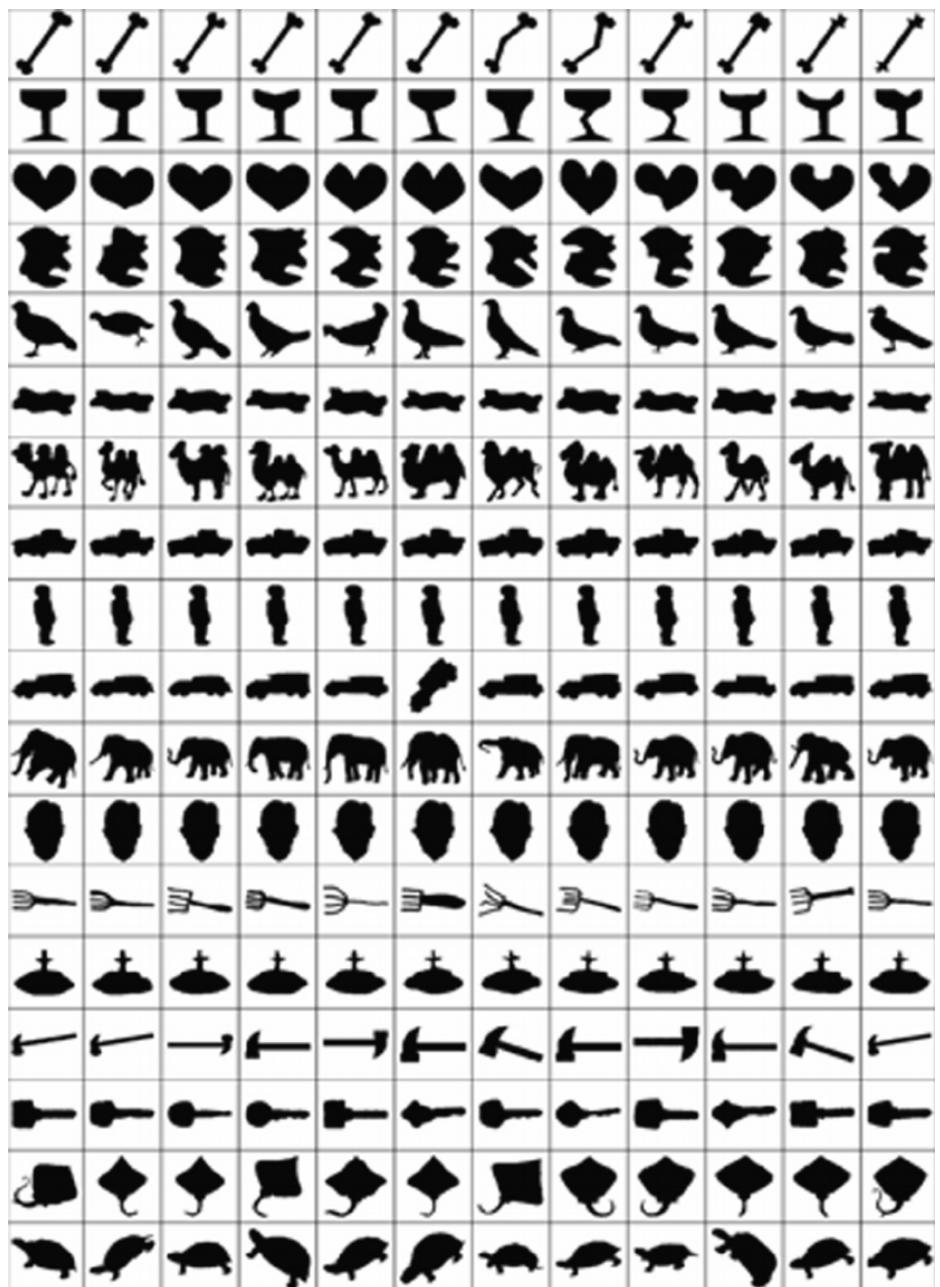


Figure 6.32: Kimia216 dataset [Sebastian et al., 2004].

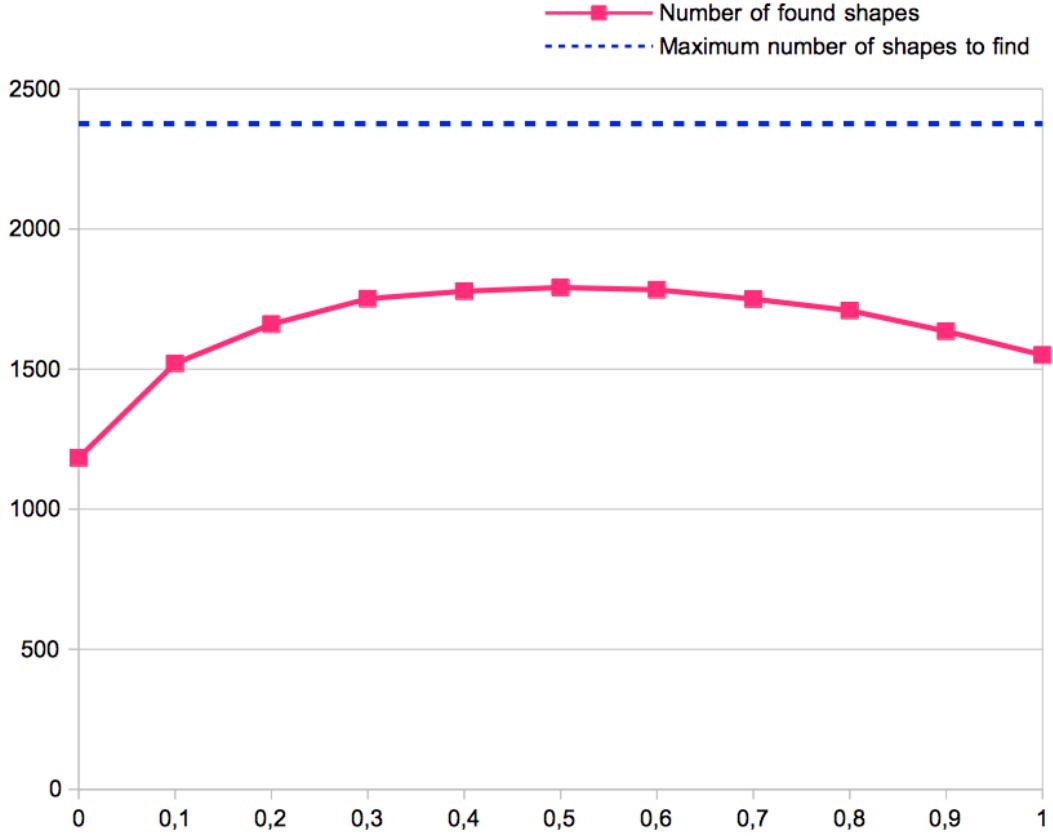


Figure 6.33: Number of shapes found versus weight α .

Parameter tuning The proposed matching method requires parameters α and β . We carried out a study to show their influence on the rate of good classification. Each class having 12 examples, we consider for each shape the 11 closest ones. Among these 11 matches, we count the number of shapes of the same class. The maximum number of shapes that can be well classified is $216 \times 11 = 2376$. Note that this counts each good match twice, which is not an

Parameter α : This parameter weights the geometric term against the length in the intrinsic dissimilarity, in Eq. (6.68). To do this, we disabled the context dissimilarity term by setting $\beta = 1$ in Eq. (6.69). The total number of well-retrieved shapes versus α is shown in Fig. 6.33. We observe that the best results were obtained with $\alpha = 0.5$. We therefore fixed this weight to 0.5 afterwards. Hence, in the intrinsic dissimilarity distance, the geometric term and the length term have equal contributions.

Parameter β : We then determined in which extent the context should be taken into account. α has been fixed previously. Fig. 6.34 presents the results obtained regarding the number of shapes found as a function of β . The best results were obtained for β equal to 0.75, and do not deteriorate significantly when this weight increases. This means that the geometry of the shape is dominant with respect to the context. So, we chose this value in the remaining experiments.

Comparison with other methods and discussions Table 6.2 reports the amount of good classifications on the Kimia216 dataset. For each shape, we rank the 11 closest shapes by increasing dissimilarity. The number of good classifications, *i.e.* the number of times a shape of the same class as the query is found, is counted at each rank. Intuitively, the amount of good classifications decreases, non-strictly, with respect to the rank. Our method provides results comparable to the shape context method. Our results are slightly

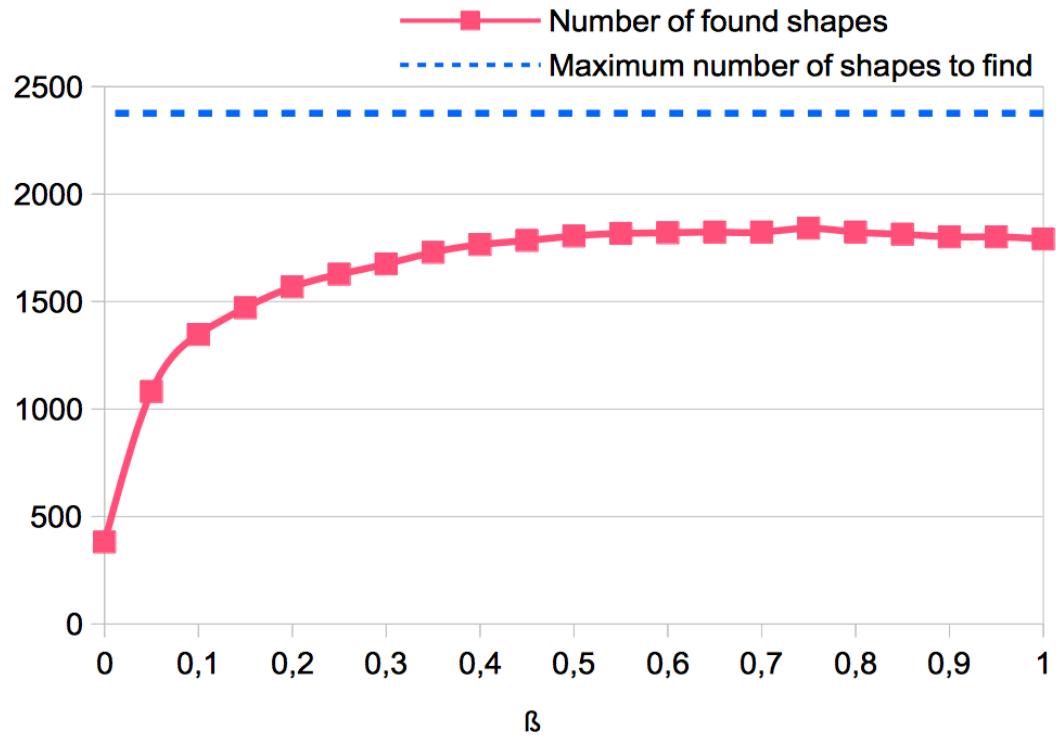


Figure 6.34: Number of shapes found versus weight β .

	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th	9 th	10 th	11 th
[Bai and Latecki, 2008]	216	216	215	216	213	210	210	207	205	191	177
[Belongie et al., 2002]	214	209	205	197	191	178	161	144	131	101	78
Proposed method	208	205	194	186	178	168	164	156	153	122	107

Table 6.2: Number of shapes associated with a shape of the same class in the Kimia216 dataset in each rank

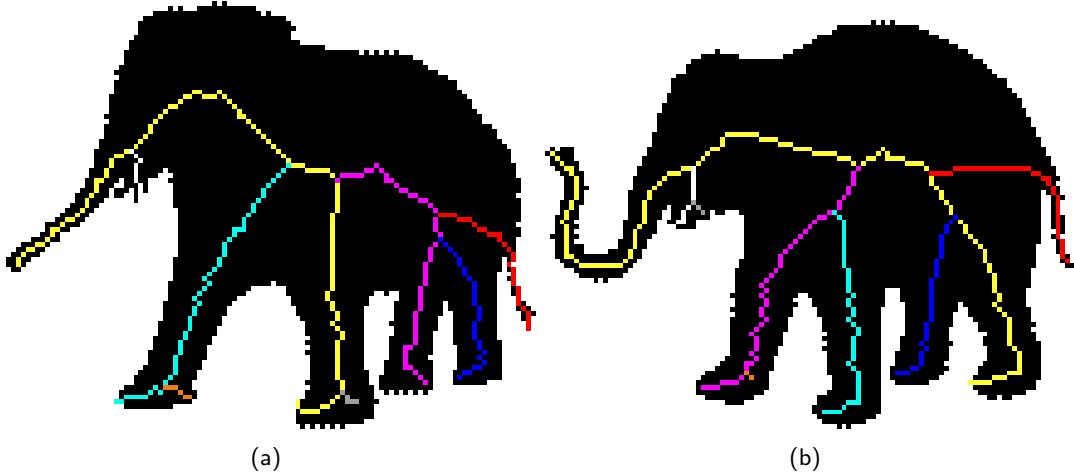


Figure 6.35: BSs obtained on two shapes of the same category, but with different arrangements of shape parts

worse in the first ranks but better in the lower ranks.

While our method has a lower complexity than [Bai and Latecki, 2008], we did not reach their performances on this database. Our interpretation is that our method is more sensitive to the rearrangement of the internal branches, as shown in Figure 6.35. If the internal geometry of the shape is different, the BSs constructed in our method will also be different. Conversely, in [Bai and Latecki, 2008], all paths between all terminal skeleton points are compared. Hence, the inter-point matching cost is robust to re-ordering of shape parts.

Classification of tree leaves

We used a dataset of tree leaf images ¹⁰, itself generated from the bases created by the organizers of the "Plant Identification" challenge, as part of the ImageClef conference [Goëau et al., 2011, Goëau et al., 2012].

The average size of images in the leaf dataset is 650×800 , whereas it was 100×100 in the Kimia216 dataset. The tuning of parameters performed on the Kimia216 dataset being independent from the size of images, the same values were used. The dataset contains 170 images gathered in 20 species. Since some species are represented by only two leaves, we sought to determine the matching only at the first rank. In other words, a leaf is considered as well-recognized if the closest match is of the same species. Note that the major difficulty in matching tree leaves is the high intra-class variability and the low inter-class variability that can arise for some species, as depicted in Figs 6.37.

Fig. 6.36 shows the number of well-matched images per species. We obtain a 72% recognition rate at first rank. In a previous collaboration [CLEF11, CVIU13], we had used leaf-specific descriptors and obtain a recognition rate of 42% at first rank. These results are, however, to be balanced with the fact that the dataset used in [CVIU13] contains more species (86 classes), and more images per class. Overall, the current experiment shows that our matching method can be used in a real context.

6.6 Conclusion

6.6.1 Summary of contributions

In the first contribution, we studied the AOF and ridgeness measures on Euclidean distance maps, on local theoretical shape configurations. These measures depend on the local geometry of the shape borders.

¹⁰This work was part of the ReVeRIES project (Reconnaissance de Vegetaux Recreative, Interactive et Educative sur Smartphone), in the 2015-2019 period, supported by the French ANR with the reference ANR- 15-CE38-004-01.

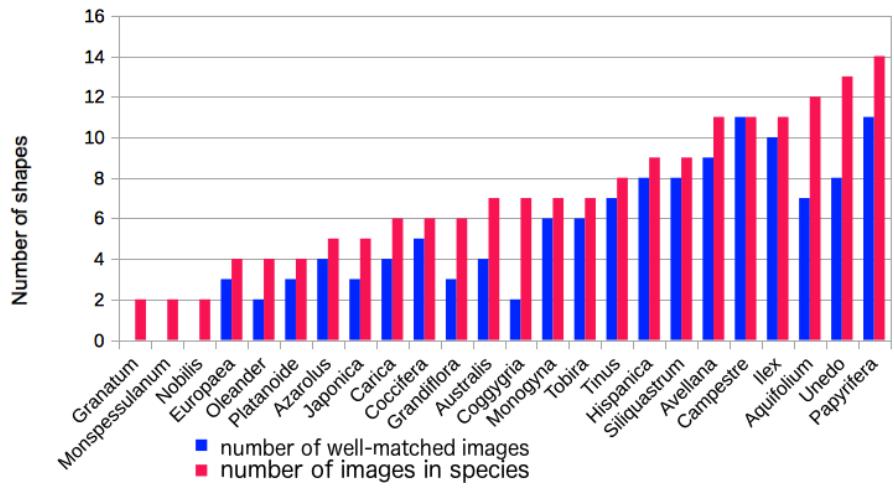


Figure 6.36: Number of shapes found in the first rank for each leaf species.

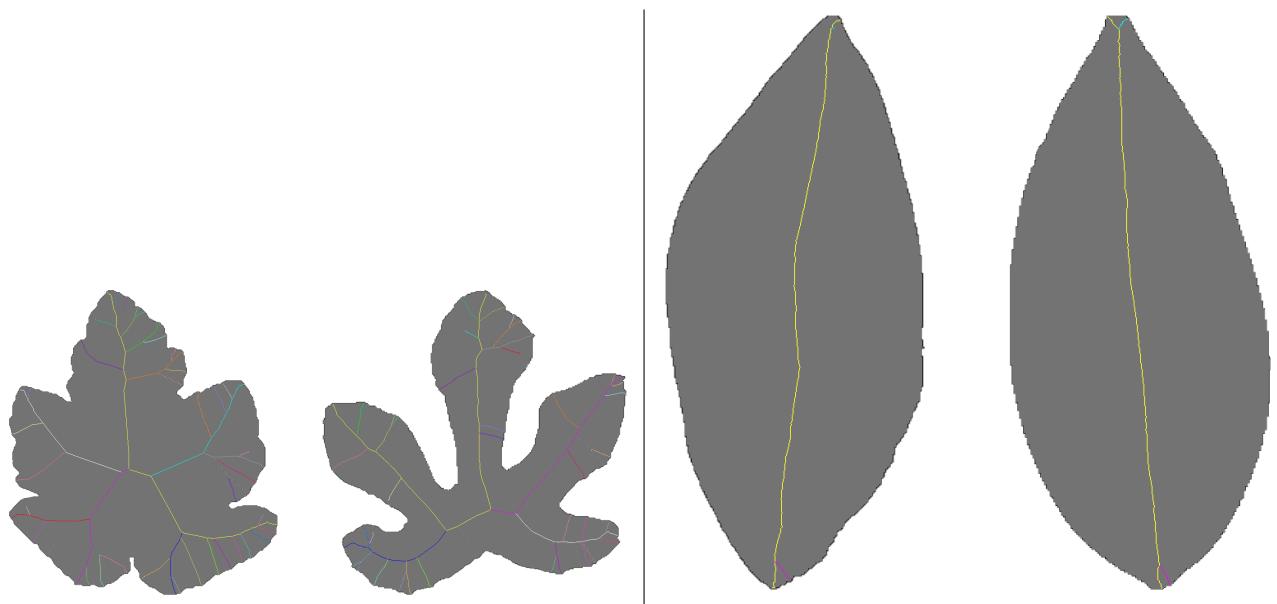


Figure 6.37: Matching of (left) two tree leaves of the same species (*carica*) with high dissimilarity and (right) two tree leaves from two different species (*grandiflora* and *aquifolium*) with very similar shapes, and low dissimilarity

Closed-form exact expressions could be obtained for regular skeleton points and their neighboring points, as well as for skeleton endpoints. As regards peak points, ligatures and junction points, exact expressions using elliptic integrals and simpler closed-form approximations based on the Taylor expansion of the distance function were derived. We established a strong relationship between the spatial parameter (r or σ) and the corresponding ideal threshold, based on an analysis of regular skeleton points and their neighboring points. This was validated by experiments on a shape dataset with known ground-truth skeletons.

In the second contribution, we developed a dissimilarity measure for binary planar shapes. In a skeleton, branches have different importances. We proposed to quantify the importance of branch sequences and favor the matching of branch sequences having a similar importance, by attributing more weight to those which encode the general appearance of the shape and less to those derived from details. The importance was quantified according to the persistence of branches as the shape is iteratively smoothed. A hierarchical skeleton was constructed from these weighted branch sequences and adjacency relations between them. The developed matching algorithm relies on the intrinsic characteristics of the branches and on the context made up of relative branch positions. Matching is formulated as a linear assignment problem, solved using a variant of the Hungarian algorithm. The final dissimilarity measure between two shapes is determined from the matching cost. Comparison against state-of-the-art algorithms on the Kimia216 dataset showed similar recognition rates, but with an algorithm of lower complexity. The method was also tested on tree leaves, a difficult context in view of the great intra-class variability and the low inter-class variability.

6.6.2 Possible research tracks

Study of 3D curve skeletons An extension of this work, that naturally comes to mind, is to tackle 3D shapes. As a possible continuation to the first contribution, AOF and ridgeness measures could be studied for theoretical configurations of 3D shapes. The extension to volumetric shapes is not trivial, as they admit two types of skeletons. *3D surface skeletons* are 2D manifolds, i.e. $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ mappings, which contain the loci of maximally-inscribed balls, whereas *3D curve skeletons* are composed of $\mathbb{R} \rightarrow \mathbb{R}^3$ curves which are locally centered in the shape. The problem is that 3D curve skeletons admit many definitions, as pointed in [Jalba et al., 2016]. A formal relationship between the surface skeleton and the curve skeleton is still not unanimously accepted. As a consequence, determining what an interesting theoretical 3D shape configuration is may not be trivial, whether one considers the surface skeleton, or the curve skeleton. Back to the 2D case, further investigation could be conducted on the corrected AOF. In that case, approximate analytical solutions to the transport equation (6) in [Torsello and Hancock, 2006] would be necessary.

Convergence of the continuous deformable skeleton Evolution equation (6.58), which fits skeleton branches to an updated distance map, was designed in a somewhat *ad hoc* manner. It was constructed from a modification of the minimization of the geodesic active contour energy, as shown in Appendix C.2. From a rigorous optimization perspective, it is unsatisfactory, as Eq. (6.58) does not directly arise from the calculus of variations of a specific energy. At the time of writing this report, the intuition is that (6.58) is a maximizer of (6.59). We currently lack the derivation leading to relation (6.60), which would make our proof. This would be a nice result, bringing extra justification to the deformable skeleton framework.

Investigation on BS matching cost In Section 6.5.2, for the calculation of the dissimilarity between two hierarchical skeletons $\text{SH}^{(1)}$ and $\text{SH}^{(2)}$, we calculated a raw matching cost between each vertex $\mu_a^{(1)}$ of $\text{SH}^{(1)}$ and each vertex $\mu_b^{(2)}$ of $\text{SH}^{(2)}$. The raw matching cost was given in Eq. (6.71), based on the optimal transportation $(f_{i,j}^*)_{i,j}$, solution of linear programming problem (6.70). Let $\overline{\text{wb}}_{\min}$ be the minimum sum of branch importances,

$$\overline{\text{wb}}_{\min} = \min \left(\sum_{i=1}^{\text{V}^{(1)}} \text{wb}(\mu_i^{(1)}), \sum_{j=1}^{\text{V}^{(2)}} \text{wb}(\mu_j^{(2)}) \right).$$

Using constraint 4 of (6.70) and Eqs (6.69) and (6.71), it can be easily shown that the raw matching cost can be expanded as

$$\begin{aligned}
\text{emd}_{a,b} &= \frac{\sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \psi_{i,j} f_{i,j}^*}{\sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} f_{i,j}^*} = \frac{1}{\overline{w}\overline{b}_{\min}} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \psi_{i,j} f_{i,j}^* \\
&= \frac{1}{\overline{w}\overline{b}_{\min}} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \left(\beta \left(\text{bd}(\mu_a^{(1)}, \mu_b^{(2)}) + \text{bd}(\mu_i^{(1)}, \mu_j^{(2)}) \right) \right. \\
&\quad \left. + (1 - \beta) \left| L_{\text{path}}(\mu_a^{(1)}, \mu_i^{(1)}) - L_{\text{path}}(\mu_b^{(2)}, \mu_j^{(2)}) \right| \right) f_{i,j}^* \\
&= \beta \text{bd}(\mu_a^{(1)}, \mu_b^{(2)}) + \frac{1}{\overline{w}\overline{b}_{\min}} \sum_{i=1}^{V^{(1)}} \sum_{j=1}^{V^{(2)}} \left(\beta \text{bd}(\mu_i^{(1)}, \mu_j^{(2)}) \right. \\
&\quad \left. + (1 - \beta) \left| L_{\text{path}}(\mu_a^{(1)}, \mu_i^{(1)}) - L_{\text{path}}(\mu_b^{(2)}, \mu_j^{(2)}) \right| \right) f_{i,j}^*
\end{aligned}$$

This expansion highlights the fact that the EMD cost, resulting from a transportation problem, actually contains a component independent from this transportation problem. The issue that arises is that weight β is involved both in the EMD-independent and EMD-dependent terms. It challenges the relevancy of the current calculation of the BS matching cost, and its responsibility in the results shown in Table 6.2. Further experiments could be done with independent weights on the branch dissimilarity $\text{bd}(\mu_a^{(1)}, \mu_b^{(2)})$ and the EMD-dependent term.

Extension to non simply-connected shapes In the second contribution, the area-preserving smoothing flow is well defined for simply-connected shapes only. This prevents the matching of non simply-connected shapes. Future work on the matching aspect could investigate extension to shapes containing holes. Unlike shapes considered in this paper, holed shapes lead to skeletons with cycles. Such extension would require to determine an appropriate stopping criterion in the iterative smoothing process, as the use of a threshold on the isoperimetric quotient (6.54) is not natural if the shape contour is not a single simple closed curve. We could consider smoothing the distance map instead of curve, as in the scale-space approach of [Sapiro and Tannenbaum, 1995]. Smoothing the distance map does not necessarily preserve topology, as holes can be filled up. Whether the topology should be maintained or not while smoothing, for determining the weights of BSs, is an open question.

Chapter 7

What next?

7.1 Looking back over my shoulder

In the previous chapters, I described a large part of the work I was involved in, for the past 12 years, targetting applications in image processing and computer vision:

- **Single-object path-based segmentation** (chapter 4). We addressed user-guided segmentation, where we are given a sequence of ordered seed points on the target object contour. In this extent, the segmentation problem consists in linking successive seeds with paths, so that the resulting curve is simple, and fits the actual object contour. The contributions described can be considered as successive refinements. We started from a rather naive unconstrained approach (Section 4.6.2), where successive seeds were connected with the shortest path, regardless of paths connecting other seed pairs. Based on the shortcomings of this method, we proceeded to an extension where several candidate disjoint paths per seed pair were generated (Section 4.7). Limitations of disjoint paths were brought to light. We thus switched to a further refinement consisting in generating the candidate paths from the saddle points of the geodesic distance map (Section 4.8).
- **Multiple object segmentation in videos** (chapter 5). This chapter described two contributions, in different directions. The first one, covered by sections 5.3 and 5.4), dealt with supervised video object segmentation. Labeling of individual objects is provided in the initial frame, the problem of detecting new objects being ignored. Pixelwise tracking to the next frame was performed by matching color distributions between objects and local surroundings of objects, which was tagged as "short-sight" background model. This part of my research arised from my integration into the Imagine team of the LIRIS, with colleagues working on video detection and tracking.

The second contribution, described in Section 5.5, was linked to the supervision of PhD student Imtiaz Ali. It addressed unsupervised video object segmentation using a pixelwise background model. Unlike in the first contribution, the emphasis is put on the detection. Object-wise labeling, as well as inter-frame correspondence of objects, are ignored, pixels being marked as "object" or "background". The pixelwise background model represents spatiotemporal texture, targetting applications where objects are to be detected in backgrounds with repetitive local patterns.

- **Skeleton-based shape matching** (chapter 6). The supervision of the PhD of Aurélie Leborgne drove me towards binary 2D shape classification, *i.e.* assigning a planar *query* object to a given class. In this purpose, my colleagues' appetite for discrete geometry led us to choose the skeleton as the basic shape descriptor, owing to its appealing properties for shape representation. The first main contribution, described in Section 6.2 was a fork from the PhD main track, independent of the final shape recognition goal. It consisted in studying ridge detectors on Euclidean distance maps, on local theoretical shape configurations. Depending on the local configuration, we could derive exact closed-form or integral-form expressions on one hand, or approximations and bounds on the other hand. Detecting candidate skeleton

points requires some thresholding on the ridge measures. As a side effect, the mathematical derivations performed in this first contribution brought justifying elements for choosing threshold parameters.

The second contribution was more directly related to the PhD. A multiscale representation of the shape was constructed, by iterative smoothing operations. We introduced the deformable skeleton, which updates to evolving shape and associated distance map, instead of calculating the skeleton from scratch for each scale. The persistence of branches across successive smoothings was used to quantify the importance of branches, and exploit this importance measure in the matching process. Using the so-called hierarchical skeleton, we developed an inter-shape dissimilarity measure based on intrinsic and extrinsic properties of branches, matched using a Hungarian-like algorithm. This contribution was reported in Sections 6.4 and 6.5. In comparison to Aurélie's original manuscript, these sections bring additional mathematical models describing the deformable skeleton.

For the two main computer vision purposes addressed in this manuscript, *i.e.* image/video object segmentation and matching, collaborators and I always tried to develop methods based on solid mathematical foundations. In this context, I was lucky to collaborate with researchers having various skills in computer science and applied mathematics. It pushed me to dabble several fundamental research fields at the interface of these two domains, such as

- optimization,
- partial differential equations,
- probability, statistics, information theory,
- differential geometry, discrete geometry,
- graph theory.

In parallel to my research perspectives, regardless of the applications that could be targetted, I wish to go on learning material in these domains.

7.2 Long-term perspective

Possible short-term and mid-term extensions related to each research part were given in Sections 4.11.2, 5.6.2 and 6.6.2, respectively. Most of them describe extensions that were envisioned and discussed with collaborators at the time, typically when final revisions of corresponding papers were under review. I wrote them in this manuscript, willingly ignoring nowadays trends in image processing/computer vision, as I still find these extensions interesting, from a very personal point of view. However, how do they fit in the current context? For example, establishing a solid probabilistic basis of the short-sight video object segmentation, as stated in Section 5.6.2, or proving that the evolution equation of a skeleton branch is a maximizer of some reward functional, as written in Section 6.6.2, are typical examples of research perspectives that fit my taste, but that would have a very restricted audience! Should this work bring enough material for publication, it would certainly be a difficult sell for top-rank "general-purpose" image processing/computer vision conferences¹ and journals². Venues with a stronger mathematical orientation³, and less demanding as regards benchmarking, are now more relevant options.

The main scientific challenges of computer vision (segmentation, detection, motion estimation, object recognition, *etc*), and the ways we assess the methods tackling these challenges, did not change over the years. Obviously, available datasets become more numerous, harder and bigger. The point is that,

¹such as CVPR, ICCV, ECCV, BMVC or ICIP

²such as International Journal of Computer Vision, IEEE Trans. Pattern Analysis and Machine Intelligence or IEEE Trans. Image Processing

³like SSVM, Journal of Mathematical Imaging and Vision, SIAM Journal on Imaging Sciences

as raised in Chapter 3, the past decade saw the massive arrival of machine learning in image processing and computer vision. Deep learning introduced the concept of end-to-end learning, where every step of the recognition process can be learned from training data. ConvNets, LSTMs, and more general concepts (generative models, adversarial learning, transfer learning, variational inference, etc) dramatically transformed the landscape of theoretical tools employed to solve imaging tasks, from semantic segmentation [Shelhamer et al., 2017] to object detection [He et al., 2020]. To push further, every application targeted in this report can now be addressed successfully with deep neural networks, most of it coming from their ability to automate the difficult feature engineering phase by learning hierarchical feature extractors from data.

My core business is low-level image processing, especially segmentation by means of energy minimization. As, I believe, many of my counterparts, an important part of my recent research time was pushed towards the testing and understanding of deep neural networks. The supervisions of the PhDs of Fabien Baradel and Luong Phat Nguyen, mentioned in Section 1.4.2 and Chapter 3, have motivated this scientific broadening. Diving into the wide world of deep networks, it is easy to fall into the trap of using building blocks of networks (convolutional layers, pooling layers, loss functions, etc) as black boxes, and focus the efforts on the design of the network architecture. To some extent, there is a challenge in avoiding this trap, going on proposing new models and means of understanding data, while staying on par with the performances of modern deep learning approaches. As stated in [O'Mahony et al., 2019], "*There are clear trade-offs between traditional computer vision and deep learning-based approaches. Classic computer vision algorithms are well-established, transparent, and optimized for performance, while deep learning offers greater accuracy and versatility at the cost of large amounts of computing resources.*" . As a personal questioning, how do my fundamental research tastes and skills stand in relation to the deep learning trend?

Writing the final chapter of this manuscript is the occasion of thinking about the various criteria coming into play to select long-term research tracks:

- domains of predilection, which were listed at the end of Section 7.1.
- research domains of team colleagues
- need to stay focused on a few areas of expertise
- keeping a publication potential
- maintaining connection to modern theoretical tools and trends (this aspect is linked to the previous one)

Hereafter, I give perspectives that I believe to be reasonable trade-offs between these criteria, and are linked to the work reported in Chapters 4 and 5. As I wish to refocus on segmentation, I do not develop long-term perspectives in shape matching.

Interactive segmentation The overwhelming majority of segmentation methods that are nowadays published in major image/vision-oriented conferences and journals are based on neural networks with convolutional layers. Famous examples are the Fully Convolutional Network [Long et al., 2015, Shelhamer et al., 2017], the U-Net architecture [Ronneberger et al., 2015], DeepLab [Chen et al., 2018] or SegNet [Badrinarayanan et al., 2017]. Once trained, all these networks are devoted to fully-automated segmentation. However, it seems that interactive segmentation has still its ticket for major computer vision venues [Li et al., 2018, Oh et al., 2019, Sofiiuk et al., 2020].

Midterm perspectives in Section 4.11.2 highlighted the relevance of extending the current work, which admits interaction at initialization only, to a fully-interactive method, which would allow on-the-fly addition, deletion or moving of seed points, and update the segmentation accordingly. It targets especially medical image segmentation for clinical purpose, or object delineation in image editing software. Moreover, recent publications suggest that path-based segmentation methods could touch a larger computer vision audience. Path-based

methods can be used as tools to help manual ground-truth segmentation, as in [Chen et al., 2020a], facilitating the construction of annotated datasets. The work in [Ling et al., 2019] is a remarkable example of interactive segmentation method combining paths and ConvNets, based on Graph Convolutional Networks [Kipf and Welling, 2017].

Video object segmentation Pixelwise video object segmentation (VOS) was successfully addressed with deep networks [Zhao et al., 2019, Chen et al., 2020b]. Using the notations introduced in Section 5.1, let us recall that the purpose is to determine a labeling, of each frame, $(\phi_t)_{t=1 \dots T}$, given an image sequence $(\mathbf{f}_t)_{t=1 \dots T}$. Integrating these notations into a supervised learning setting, we are given a set of N training image sequences and ground-truth labelings $\{(\mathbf{f}_t^j)_{t=1 \dots T^j}, (\phi_t^j)_{t=1 \dots T^j}\}_{j=1 \dots N}$. A network \mathcal{M} , with trainable parameters θ , is trained in order to minimize the error between the predicted labeling $\hat{\phi}$ and the ground-truth labeling ϕ ,

$$E[\theta] = \sum_{j=1}^N \sum_{t=1}^{T^j} \mathcal{L}(\hat{\phi}_t^j, \phi_t^j) = \sum_{j=1}^N \sum_{t=1}^{T^j} \mathcal{L}(\mathcal{M}(\mathbf{f}_t^j; \theta), \phi_t^j),$$

where \mathcal{L} is a loss summed over all pixels, typically a negative log-likelihood loss, or multiclass cross-entropy, assuming that \mathcal{M} generates soft-max outputs for each pixel. Assuming that the network operates in a sequential manner, predicting the labeling in the next frame, given current labeling, current image and next image, we would have the following error function,

$$E[\theta] = \sum_{j=1}^N \sum_{t=1}^{T^j-1} \mathcal{L}(\mathcal{M}(\mathbf{f}_t^j, \mathbf{f}_{t+1}^j, \phi_t^j; \theta), \phi_{t+1}^j),$$

So far, it appears that deep learning and prior knowledge have not been well integrated together, as pointed out in [Marcus, 2018]. Possible extensions would consist in integrating *VOS-friendly* terms in the loss function, to balance the influence of the supervised term, depending on the ground-truth annotation ϕ , and the effect of ground-truth-independent terms

$$E[\theta] = \sum_{j=1}^N \sum_{t=1}^{T^j-1} \mathcal{L}(\mathcal{M}(\mathbf{f}_t^j, \mathbf{f}_{t+1}^j, \phi_t^j; \theta), \phi_{t+1}^j) + E_{\text{data}}[\mathcal{M}(\mathbf{f}_t^j, \mathbf{f}_{t+1}^j, \phi_t^j; \theta)] + E_{\text{reg}}[\mathcal{M}(\mathbf{f}_t^j, \mathbf{f}_{t+1}^j, \phi_t^j; \theta)].$$

Here, energies E_{data} and E_{reg} are ground-truth-independent terms, that could be the ones used in Eq. (5.17), or any data and regularization terms relevant for joint segmentation and tracking. Ground-truth-independent terms embody prior knowledge on motion, shape and appearance of tracked objects. The key intuition is that combining ground-truth dependent and independent terms could improve generalization abilities of the network, making it less dependent on the training data.

Appendix A

Minimal paths

A.1 Mathematical derivations of overlap and exteriority terms

A.1.1 Overlap term

Let \mathbf{c} be a regular curve parameterized over $[0, L]$. Let ϕ be a C^1 function defined over $[0, L]^2$ representing the distance between two positions on the curve:

$$\phi(u, v) = \|\mathbf{c}(u) - \mathbf{c}(v)\|^p$$

where p is an arbitrary positive real exponent. The length of the zero level set of ϕ ,

$$|\mathcal{Z}_c| = \int_0^L \int_0^L \delta(\phi(u, v)) \|\nabla \phi(u, v)\| \, du \, dv, \quad (\text{A.1})$$

quantifies the self-overlap of \mathbf{c} .

Proposition:

If \mathbf{c} is simple, i.e. without self-intersection and self-tangency, then $|\mathcal{Z}_c| = L\sqrt{2}$.

Proof:

As a preliminary calculation, let us express the gradient of ϕ (partial derivatives are written using the indexed notation):

$$\begin{aligned} \nabla \phi(u, v) &= [\phi_u(u, v) \ \phi_v(u, v)]^T \\ &= p \|\mathbf{c}(u) - \mathbf{c}(v)\|^{p-2} \begin{bmatrix} \mathbf{c}'(u) \cdot (\mathbf{c}(u) - \mathbf{c}(v)) \\ -\mathbf{c}'(v) \cdot (\mathbf{c}(u) - \mathbf{c}(v)) \end{bmatrix} \end{aligned}$$

If \mathbf{c} is regular and simple, varying with respect to u in range $[0, L]$, $\phi(u, v)$ is nowhere zero except when $u = v$. Hence, for a fixed v , we have:

$$\delta(\phi(u, v)) = \frac{\delta(u - v)}{|\phi_u(v, v)|} \quad (\text{A.2})$$

Integrating (A.2) into (A.1) and applying the definition of measure δ :

$$\begin{aligned} |\mathcal{Z}_c| &= \int_0^L \int_0^L \delta(\phi(u, v)) \|\nabla \phi(u, v)\| \, du \, dv \\ &= \int_0^L \int_0^L \frac{\delta(u - v)}{|\phi_u(v, v)|} \|\nabla \phi(u, v)\| \, du \, dv \\ &= \int_0^L \frac{\|\nabla \phi(v, v)\|}{|\phi_u(v, v)|} \, dv \end{aligned}$$

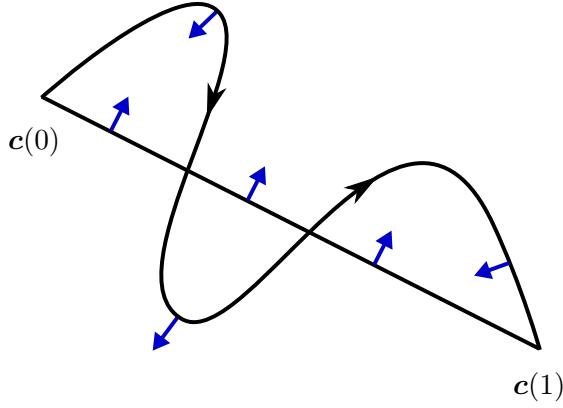


Figure A.1: The exteriarity of an open curve is measured as the signed area of the multiple connected region that it forms with the line segment joining its two endpoints.

Trivially, $\phi(v, v) = 0$. However expanding the gradient gives:

$$\begin{aligned} |\mathcal{Z}_c| &= \int_0^L \left\{ p \|\mathbf{c}(v) - \mathbf{c}(v)\|^{p-2} \frac{\sqrt{2(\mathbf{c}'(v) \cdot (\mathbf{c}(v) - \mathbf{c}(v)))^2}}{|\mathbf{c}'(v) \cdot (\mathbf{c}(v) - \mathbf{c}(v))|} \right\} dv \\ &= \int_0^L \sqrt{2} dv \\ &= L\sqrt{2} \end{aligned}$$

A.1.2 Exteriarity term

Let c be a piecewise-smooth regular open curve parameterized over $[0, 1]$.

Proposition:

The signed area of the curve formed by the concatenation of c and the line segment from $c(1)$ returning to $c(0)$ (see Fig. A.1), which we use as the *exteriarity* measure in Section 4.9.2, may be expressed as:

$$\mathcal{X}[c] = \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \mathbf{c}^\perp(1) \cdot \mathbf{c}(0)$$

Proof:

Let $S : [0, 1] \rightarrow \mathcal{D}$ be the parametrization of the line segment joining $c(1)$ and $c(0)$,

$$S(u) = (1-u)c(1) + uc(0).$$

The exteriarity is then obtained by using the signed area (4.5) on a piecewise basis:

$$\begin{aligned} \mathcal{X}[c] &= \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \int_0^1 S^\perp(u) \cdot S'(u) du \\ &= \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \int_0^1 ((1-u)\mathbf{c}^\perp(1) + u\mathbf{c}^\perp(0)) \cdot (\mathbf{c}(0) - \mathbf{c}(1)) du \\ &= \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \int_0^1 (1-u)\mathbf{c}^\perp(1) \cdot \mathbf{c}(0) + u\mathbf{c}^\perp(1) \cdot \mathbf{c}(0) du \\ &= \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \int_0^1 \mathbf{c}^\perp(1) \cdot \mathbf{c}(0) du \\ &= \frac{1}{2} \int_0^1 \mathbf{c}^\perp \cdot \mathbf{c}' du + \frac{1}{2} \mathbf{c}^\perp(1) \cdot \mathbf{c}(0) \end{aligned}$$

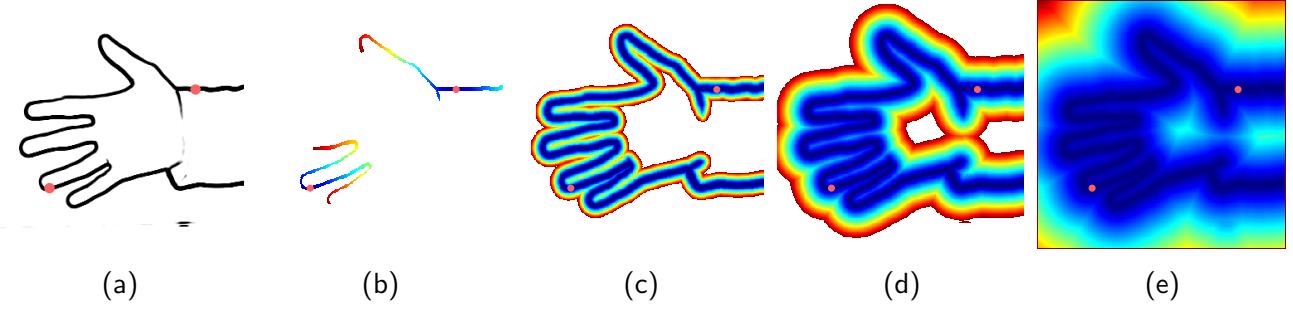


Figure A.2: (a) Potential and two source points. (b-e) Several steps of the computation of the minimal action map.

A.2 Implementation details

A.2.1 Fast Marching propagation and extraction of saddle points

As described in Section 4.8.1, admissible paths joining two source points \mathbf{a} and \mathbf{b} are constructed by extracting the saddle points of the combined action map $U_{\mathbf{a}, \mathbf{b}}$, located on the medial set $\mathcal{M}_{\mathbf{a}, \mathbf{b}}$. The minimal action map $U_{\mathbf{a}, \mathbf{b}}$ can be estimated, at each point of the discrete image domain $\hat{\mathcal{D}} = \mathcal{D} \cap \mathbb{Z}^2$, by solving the discrete derivative free Eikonal equation [Tsitsiklis, 1995, Peyré et al., 2010]

$$\begin{cases} \forall \mathbf{x} \in \hat{\mathcal{D}} \setminus \{\mathbf{a}, \mathbf{b}\}, \quad U_{\mathbf{a}, \mathbf{b}}(\mathbf{x}) = \mathcal{F}_{\mathbf{x}}(U_{\mathbf{a}, \mathbf{b}}), \\ U_{\mathbf{a}, \mathbf{b}}(\mathbf{a}) = U_{\mathbf{a}, \mathbf{b}}(\mathbf{b}) = 0, \end{cases} \quad (\text{A.3})$$

where

$$\mathcal{F}_{\mathbf{x}}(U_{\mathbf{a}, \mathbf{b}}) = \min_{\mathbf{y} \in \partial N(\mathbf{x})} U_{\mathbf{a}, \mathbf{b}}(\mathbf{y}) + W_P(\mathbf{y}, \mathbf{x}) \|\mathbf{x} - \mathbf{y}\|, \quad (\text{A.4})$$

and $\partial N(\mathbf{x}) \subset \mathcal{D}$ defines the boundary of a topological ball, usually the boundary of the convex hull of the 4-connected or 8-connected grid neighbors of \mathbf{x} . According to (A.4), the minimal action at \mathbf{x} is approximated by the minimal combination of the Euclidean distance between \mathbf{x} and a point $\mathbf{y} \in \partial N(\mathbf{x})$, weighted by potential P along segment $\overline{\mathbf{x}\mathbf{y}}$, and the minimal action at \mathbf{y} obtained by affine interpolation of $U_{\mathbf{a}, \mathbf{b}}$ from its two nearest grid points in $\partial N(\mathbf{x})$. Let \mathbf{x}_i and \mathbf{x}_j be these two points. Then, update operator (A.4) becomes

$$\begin{aligned} \mathcal{F}_{\mathbf{x}}(U_{\mathbf{a}, \mathbf{b}}) &= \min_{\overline{\mathbf{x}_i \mathbf{x}_j} \in \partial N(\mathbf{x})} \min_{t \in [0, 1]} \alpha(t), \\ \alpha(t) &= (1-t)U_{\mathbf{a}, \mathbf{b}}(\mathbf{x}_i) + tU_{\mathbf{a}, \mathbf{b}}(\mathbf{x}_j) + W_P(\mathbf{y}, \mathbf{x}) \|\mathbf{x} - \mathbf{y}\|, \end{aligned} \quad (\text{A.5})$$

where $\mathbf{y} = (1-t)\mathbf{x}_i + t\mathbf{x}_j$. Several interpolation approaches have been explored to compute the weight W_P (see [Appia and Yezzi, 2010] for recent comparisons). In our experiments we have simply used $W_P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x})$. Regardless of the choice of the neighborhood and the weight, the solution of (A.3) can be approximated by the Fast Marching algorithm, which propagates two discrete fronts simultaneously, one starting from \mathbf{a} and the other from \mathbf{b} , until they meet to form a discrete version of the medial set $\mathcal{M}_{\mathbf{a}, \mathbf{b}}$, as illustrated in Fig. A.2. Algorithm 9 presents a non-optimized version of the Fast Marching. Fronts are represented by a set Q initialized with the source points.

At each step (while loop), a point of the fronts having a minimal action is removed from Q , and the minimal action of its neighbors is updated in consequence by solving (A.5) (see [Appia and Yezzi, 2010, Peyré et al., 2010] for details). In particular, a neighbor which has not yet been explored by the fronts becomes a new point of the fronts and is thus added to Q . This propagation process is iterated until Q is empty, which guarantees the computation of $U_{\mathbf{a}, \mathbf{b}}$ at each point of $\hat{\mathcal{D}}$. Other instructions are related to the computation of the medial set, explained in the following.

Algorithm 9: Minimal action map and Voronoi map.

```

func VoronoiMap( $a, b, P, \text{PATH2LABEL}$ )
input
 $a, b \in \hat{\mathcal{D}}$  : source points
 $P : \hat{\mathcal{D}} \rightarrow \mathbb{R}^{+*}$  : discrete potential
 $\text{PATH2LABEL}$  : a method solving (A.8)
variables
 $U : \hat{\mathcal{D}} \rightarrow \mathbb{R}^+$ : minimal action map
 $\nabla U : \hat{\mathcal{D}} \rightarrow \mathbb{R}^2$ : gradient of  $U$ 
 $\text{vor} : \hat{\mathcal{D}} \rightarrow \{0, 1, 2\}$ : Voronoi map
 $Q$ : front set encoded as a priority queue

 $U[x] := +\infty$ ,  $\text{vor}[x] := 0$ ,  $\nabla U[x] := 0$ ,  $\forall x \in \hat{\mathcal{D}}$ 
 $U[a] := U[b] := 0$ ,  $\text{vor}[a] := 1$ ,  $\text{vor}[b] := 2$ 
 $Q := \{a, b\}$ 
while  $Q \neq \emptyset$  do
     $x := \text{argmin} \{U[x] \mid x \in Q\}$ 
     $Q := Q \setminus \{x\}$ 
     $\text{vor}[x] := \text{PATH2LABEL}(x, \nabla U, \text{vor})$ 
    for  $y \in N(x)$  do
        if  $U[y] = +\infty$  then
             $Q := Q \cup \{y\}$ 
        end if
         $u_{\text{new}} := \mathcal{F}_y(U)$ 
        if  $u_{\text{new}} < U[y]$  then
             $U[y] := u_{\text{new}}$ 
            update  $\nabla U[y]$ 
        end if
    end for
end while
return  $(U, \nabla U, \text{vor})$ 

```

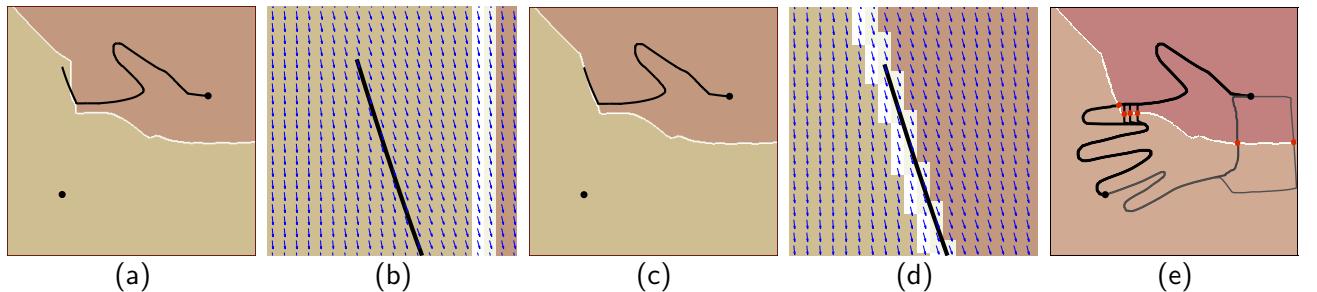


Figure A.3: (a) Voronoi map obtained by updating Voronoi labels locally [Benmansour and Cohen, 2009, Peyré et al., 2010]: points are not always well labelize, particularly in regions of \mathcal{D} where the gradient of the minimal action map and the medial set are aligned. (b) A zoom of (a) with $-\nabla U_{a,b}$. (c) Voronoi map constructed using back-propogation to determine labels (by Algorithm 9): minimal paths are ensured to be located in only one Voronoi region. (e) Saddles (red dots) and associated minimal paths between the two source points.

Saddle points on the medial set $\mathcal{M}_{a,b}$ are difficult to localize during the propagation, mainly due to discretiza-

tion. It is easier to compute a discrete medial set which is then traversed in order to approximate the saddle points. The medial set separates \mathcal{D} into two regions, the first one composed of the points at least as close to a as to b ,

$$\text{reg}_a = \{x \in \mathcal{D} \mid U_a(x) \leq U_b(x)\}, \quad (\text{A.6})$$

and the other one, reg_b , defined similarly. The underlying partition of \mathcal{D} , called *Voronoi partition*, satisfies $\mathcal{D} = \text{reg}_a \cup \text{reg}_b$ and $\mathcal{M}_{a,b} = \text{reg}_a \cap \text{reg}_b$. Together with this partition, we also define the *Voronoi map*

$$\text{vor}(x) = \{l(y) \mid x \in \text{reg}_y, y \in \{a, b\}\}, \quad (\text{A.7})$$

which provides for each point $x \in \mathcal{D}$ the labels of its nearest source points, according to a label function $l : \{a, b\} \rightarrow \{1, 2\}$. Obviously, the medial set satisfies $\mathcal{M}_{a,b} = \{x \in \mathcal{D} \mid \text{card}(\text{vor}(x)) > 1\}$. In practice, the Voronoi map is only estimated at points of $\hat{\mathcal{D}}$, when (A.3) is solved. This is usually realized locally, in the neighborhood $N(x)$ of each point $x \in \hat{\mathcal{D}}$, by assigning to x the label of the grid point closest to the optimal neighbor $y \in \partial N(x)$. Using the Fast Marching algorithm, labels are thus propagated simultaneously to minimal actions, starting with $\text{vor}(a) = 1$ and $\text{vor}(b) = 2$ (see [Benmansour and Cohen, 2009, Peyré et al., 2010]). By construction, considering only one optimal point $y \in \partial N(x)$ implies that each point of $\hat{\mathcal{D}}$, being or not on the medial set $\mathcal{M}_{a,b}$, is always connected to exactly one source point by a unique minimal path. As illustrated in Fig. A.3(a), the resulting Voronoi map is not always consistent with the definition of a Voronoi partition, as some minimal paths may connect two points having different labels. This leads to errors in the localization of the medial set¹. To overcome this drawback, which is mainly due to discretization, neighborhoods used to determine labels must be extended. This can be done by explicitly solving back-propagation (4.27) at each step of the Fast Marching algorithm (function PATH2LABEL in while loop of Algorithm 9). Let $x \in Q$ be the front point having a minimal action and which is going to be labelized. Starting from x , the back-propagation is stopped when a source point, or a point surrounded by points of $\hat{\mathcal{D}}$ having the same Voronoi label, is reached. The proposed *discrete Voronoi map* is finally defined by $\tilde{\text{vor}} : \hat{\mathcal{D}} \rightarrow \{1, 2\}$,

$$\tilde{\text{vor}}(x) = \tilde{\text{vor}}(\tilde{\gamma}(t_l)), \quad \forall x \in \hat{\mathcal{D}}, \quad (\text{A.8})$$

such that $\gamma \subset \mathcal{D}$ is the minimal path solving

$$\begin{cases} \forall t > 0, \quad \frac{d}{dt} \gamma(t) = -\frac{\nabla U_{a,b}(\gamma(t))}{\|\nabla U_{a,b}(\gamma(t))\|}, \\ \gamma(0) = x, \end{cases} \quad (\text{A.9})$$

$\tilde{\gamma}(t_l) \in \hat{\mathcal{D}}$ is the nearest grid neighbor of $\gamma(t_l) \in \mathcal{D}$, and $t_l \geq 0$ satisfies

$$\tilde{\gamma}(t_l) \in \{a, b\}, \text{ or } \tilde{\text{vor}}(\tilde{\gamma}(t_l)) = \tilde{\text{vor}}(y), \quad \forall y \in N_8(\tilde{\gamma}(t_l)),$$

where $N_8(\tilde{\gamma}(t_l)) \subset \hat{\mathcal{D}}$ represents the 8-connected neighbors of $\tilde{\gamma}(t_l)$. Such a neighborhood ensures $\tilde{\gamma}(t_l)$ to be located in the interior of a Voronoi region being propagated. Remark that the gradient of the combined action map, involved in (A.9), can be estimated during the propagation from the updated minimal action (for loop of Algorithm 9). The proposed Voronoi labelling approach allows to preserve the computational advantages of the Fast Marching algorithm, while guarantying discrete Voronoi maps, and associated medial sets, consistent with the extraction of minimal paths, as observed in Fig. A.3(c).

Based on the discrete Voronoi map, an approximation of the medial curve can now be defined as a sequence of connected pixel corners, as illustrated in Fig. A.4. Since the two Voronoi regions are 4-connected sets, the discrete medial curve is 4-connected and simple. Hence, it can be extracted by a basic edge-linking procedure. Starting from any pixel corner on the boundary, pixel edges incident to pixels having different Voronoi labels are followed. Two situations may arise, whether the medial curve is open or closed. In the first

¹Note that computing $U_{a,b}$ in one propagation also introduces errors on the minimal actions, on and near the medial set, comparing to the map $\min(U_a, U_b)$ obtained with two separate propagations. But this should not affect so much the localization of the Voronoi regions.

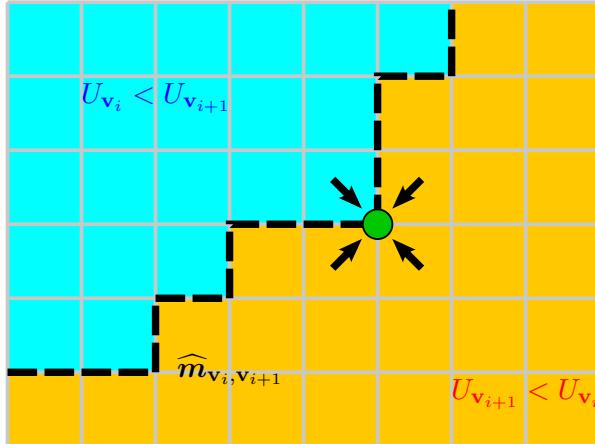


Figure A.4: Digital 4-connected medial curve on pixel edges. The combined action map on the medial point (pixel corner) is estimated using bilinear interpolation over the 4 neighboring pixels.

situation, the two Voronoi regions are simply connected, i.e. without hole, and the medial curve reaches the boundaries of the image domain (see Fig. A.3(e)). The edge-linking procedure should be run twice, taking the two possible directions starting from the initial pixel corner. In the second situation, one of the Voronoi region is entirely surrounded by the other, making the medial curve closed. Quite simply, the edge-linking procedure is stopped when the initial pixel corner is met again. Note that the orientation of the curve is not critical for the extraction of saddle points, thus the edge-linking step may be performed arbitrarily clockwise or counter-clockwise.

Let us shorten $U_{a,b}$ to U , and let \widehat{m} be the discrete medial curve approximating $m_{a,b}$. We chose to estimate U at each pixel corner on \widehat{m} by bilinear interpolation. Trivially, the combined action at a pixel corner is taken as the average of the combined actions at the four neighboring pixels. To account for possible noise, values of the combined action are smoothed along \widehat{m} by Laplacian regularization. In order to extract robust saddle points in spite of the effect of discretization, local minima are considered up to second order. A point \widehat{m}_j is marked as a saddle point of U if

$$U(\widehat{m}_{j-2}) > U(\widehat{m}_{j-1}) > U(\widehat{m}_j) \quad \text{and} \quad U(\widehat{m}_j) < U(\widehat{m}_{j+1}) < U(\widehat{m}_{j+2}).$$

Actually, not all saddle points are kept as starting points for path construction. In practice, we limit the number of admissible paths per set so that it does not exceed a user-defined threshold K_{\max} . If the number of detected local minima is greater than the threshold, they are sorted by increasing combined action and only the K_{\max} lowest saddle points are taken as initial points for admissible paths.

A.2.2 Using digital curves

Discrete computation of region integrals involved in the region term in Eq. (4.35), as well as the amount of self-tangency and twisting in Eqs. (4.30) and (4.34) is performed over digital 4-connected curves. Gradient descent over the minimal action map generates admissible paths which are sampled continuous curves, i.e. sequences of points in \mathbb{R}^2 separated by an arbitrary step. Approximating region integrals and detecting intersections is more convenient on curves with integer coordinates and constant spacing between points, thus each admissible path is discretized as an open digital 4-connected curve (see Fig. A.5). The resulting digital assembled contour $\hat{\Gamma}$ is a sequence of \hat{n} points in \mathbb{Z}^2 ,

$$\hat{\Gamma} = \langle \hat{\Gamma}_1, \hat{\Gamma}_2, \dots, \hat{\Gamma}_{\hat{n}} \rangle,$$

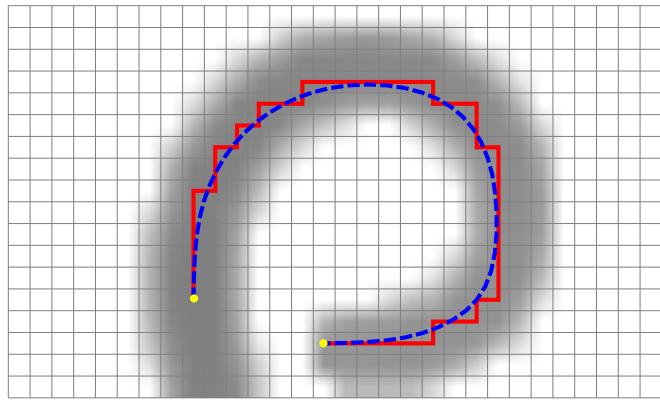


Figure A.5: Path with real coordinates with subpixel precision (dashed blue line) and its corresponding digital 4-connected path (plain red line) over the potential grid.

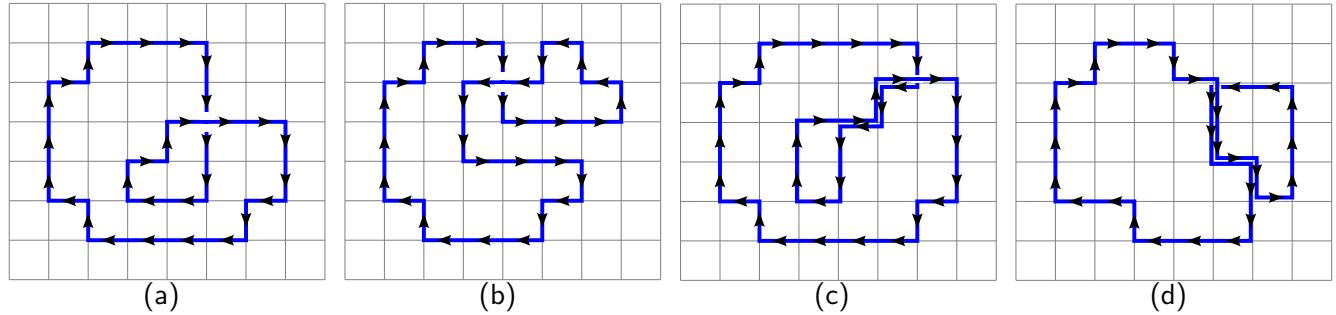


Figure A.6: Various types of simples loops on digital 4-connected curves: (a) Positive crossing with no overlapping section (b) Negative crossing with no overlapping section (c) Positive crossing with overlapping section (d) Negative crossing with overlapping section.

such that $\|\hat{\Gamma}_{i+1} - \hat{\Gamma}_i\| = 1$, $\forall 1 \leq i \leq \hat{n} - 1$, i.e. successive points are distinct and differ by only one coordinate. Moreover, the sequence is closed, hence $\|\hat{\Gamma}_1 - \hat{\Gamma}_{\hat{n}}\| = 1$. We apply the discrete Green's theorem [Brlek et al., 2005, Tang, 1982] on $\hat{\Gamma}$ to efficiently approximate the region integrals.

We now focus on the implementation of self-tangency and twisting in Eqs. (4.30) and (4.34). One should be aware that detecting intersections and corresponding loops may be done on continuous sampled curves, by finding intersections between line segments [Bentley and Ottmann, 1979, Chazelle and Edelsbrunner, 1992]. A part of the literature deals with this non-trivial task for self-intersection prevention and topology changes in active contours, as in [Nakhmani and Tannenbaum, 2012]. However, self-intersections and self-tangencies are more conveniently studied on digital 4-connected curves. When such a curve crosses or overlaps itself, it necessarily contains several occurrences of the same point, which is straightforward to detect.

Instead of a naive implementation of the self-tangency measure derived from Eq. (4.30), which would yield a $O(\hat{n}^2)$ complexity, it is computed in a single pass by storing visited pixels in the discretized image domain $\hat{\mathcal{D}}$ and detecting edges between adjacent pixels that were already taken. Simultaneously, we check that the curve does not contain points with multiplicity greater than 2. If such case happens, the curve is most likely an irrelevant candidate combination and its energy is set to $+\infty$. In this way, we make sure that the upcoming twisting term can be safely computed. This simple method is summarized in Algorithm 10.

Algorithm 10: Self-tangency measure

```

func selfTangency( $\hat{\Gamma}$ ) :  $\mathbb{R}$ 
input
 $\hat{\Gamma}$  : digital 4-connected curve, sequence of  $\hat{n}$  points in  $\mathbb{Z}^2$ 
variables
 $L \in \mathbb{R}$ : length of self-tangent parts
 $M : \hat{\mathcal{D}} \mapsto \mathbb{N}$ : array of visited positions
 $D : \hat{\mathcal{D}} \mapsto \{\text{TRUE}, \text{FALSE}\}$ : array indicating double points

 $L := 0$ 
for all  $x \in \hat{\mathcal{D}}$  do
     $M[x] := 0$ 
     $D[x] := \text{FALSE}$ 
end for
for  $i := 1$  to  $\hat{n}$  do
    if  $M[\hat{\Gamma}_i] \neq 0$  then
        if  $D[\hat{\Gamma}_i] = \text{FALSE}$  then
             $j := M[\hat{\Gamma}_i]$ 
             $D[\hat{\Gamma}_i] := \text{TRUE}$ 
            if  $\hat{\Gamma}_{j+1} = \hat{\Gamma}_{i+1}$  or  $\hat{\Gamma}_{j-1} = \hat{\Gamma}_{i+1}$  then
                 $L := L + 1$ 
            end if
        else
            return  $+\infty$ 
        end if
    end if
     $M[\hat{\Gamma}_i] := i$ 
end for
return  $\frac{L}{\hat{n}}$ 

```

As regards the twisting measure, its computation basically implies to detect positive and negative crossings. Ideally, a self-crossing involves a single point - put another way, two curve positions - as depicted in Fig. A.6(a) and A.6(b). For a positive (resp. negative) crossing, the curve arrives from the right (resp. left) and leaves to the left (resp. right) at the same point. However, due to the effect of discretization and the fact that the curves rarely crosses itself orthogonally, intersections with overlapping sections are more common, as depicted in Fig. A.6(c) and A.6(d). Since only areas of inverted loops should be considered to compute the twisting measure, regardless of the overlap created by these inverted loops, we focus solely on the events along the curve that helps to detect inverted single and double loops, as previously shown in Fig. 4.13. Let us consider that curve points are marked as long as the curve is traveled. Two basic events can be detected:

- **entrance:** the curve comes from an unmarked area and “enters” on a marked curve point
- **leaving:** the curve “leaves” a marked point and goes into an unmarked area.

The entrance is kept as the event meaning that a loop is created. It turns out that detecting *left-sided events*, i.e. the curve leaves to the left or enters from the left, is sufficient to detect simple and double inverted loops. At this stage, we safely assume that the curve contains only points of multiplicity ≤ 2 . This property was previously checked during the computation of the self-tangency term in Algorithm 10. Suppose the curve enters a previously marked curve point from the left and denote the crossing by (i, j) such that $\hat{\Gamma}_i$ is the “intersected” point (the position where the curve was visited first) and $\hat{\Gamma}_j$ is the “intersecting” point (the current position). The curve is traveled again from index i until j is met again (in such case, the loop is simple) or a previous left-sided leaving is met (in such case, the loop is double). Whether an event is left-sided or not depends on the sign of the dot product between the approximated normal at $dcurve_i$ and tangent at $\hat{\Gamma}_j$,

$$(\hat{\Gamma}_j - \hat{\Gamma}_{j-1}) \cdot (\hat{\Gamma}_i - \hat{\Gamma}_{i-1})^\perp,$$

which should be positive for an entrance and negative for a leaving. The extraction of sets of single loops SL and double loops DL is detailed in Algorithm 11. The twisting measure is then computed by applying the discrete Green’s theorem to Eq. (4.34).

A.2.3 Computation of the best combination of paths

The optimization procedure described in Section 4.9.2 is formalized in Algorithm 12. Variables $e_{\text{min-iter}}$ and e_{min} are respectively the current local minimum - the best combination found at the current iteration - and the current global minimum, i.e. the best configuration found since the beginning of the procedure. A special case arises when $e_{\text{min-iter}}$ is infinite at the end of the iteration, which happens when all tested curves have a negative area or an infinite simplicity term. In this case, the combination with the smallest area is selected as the starting sequence for the next iteration. This allows to have the slowest dilation of the contour and hence reduces the risk of skipping combinations that could be relevant. With a view to conciseness, the energy of the current combination of selected admissible paths $E[\gamma_{1,p_1} \sqcup \gamma_{2,p_2} \sqcup \dots \gamma_{n,p_n}]$ is shortened to $E(p_1, \dots, p_n)$. Similarly, the inner area of this combination is shortened to $A(p_1, \dots, p_n)$.

Algorithm 11: Extraction of single and double loops

```

func singleAndDoubleLoops( $\hat{\Gamma}$ )
input
 $\hat{\Gamma}$ : digital 4-connected path, sequence of  $\hat{n}$  points in  $\mathbb{Z}^2$ 
output
SL: set of pairs of indices  $\subset \mathbb{N}^2$ 
DL: set of double pairs of indices  $\subset (\mathbb{N}^2)^2$ 
variables
onCurve  $\in \{\text{TRUE}, \text{FALSE}\}$ 
loopFound  $\in \{\text{TRUE}, \text{FALSE}\}$ 
 $M : \hat{\mathcal{D}} \mapsto \mathbb{N}$ : array of visited positions
Le  $\subset \mathbb{N}^2$ : set of indices indicating left-sided events

onCurve := FALSE
SL :=  $\emptyset$ , DL :=  $\emptyset$ , Le :=  $\emptyset$ 
for each  $x \in \hat{\mathcal{D}}$  do
     $M[x] := 0$ 
end for
for  $j := 1$  to  $\hat{n}$  do
    if onCurve = FALSE then
        if  $M[\hat{\Gamma}_j] = 0$  then
             $M[\hat{\Gamma}_j] := j$ 
        else
            onCurve := TRUE ;  $i := M[\hat{\Gamma}_j]$ 
            if  $(\hat{\Gamma}_j - \hat{\Gamma}_{j-1}) \cdot (\hat{\Gamma}_i - \hat{\Gamma}_{i-1})^\perp > 0$  then
                Le := Le  $\cup \{(i, j)\}$ 
                 $k := i + 1$  ; loopFound := FALSE
                while loopFound = FALSE do
                    if  $k = j$  then
                        loopFound := TRUE
                        SL := SL  $\cup \{(i, j)\}$ 
                    else
                        if  $\exists l \text{ s.t. } (k, l) \in \text{Le}$  then
                            loopFound := TRUE
                            DL := DL  $\cup \{(i, j), (k, l)\}$ 
                        end if
                    end if
                     $k := k + 1$ 
                end while
            end if
        end if
    else
        if  $M[\hat{\Gamma}_j] = 0$  then
             $M[\hat{\Gamma}_j] := j$  ; onCurve := FALSE ;  $i := M[\hat{\Gamma}_{j-1}]$ 
            if  $(\hat{\Gamma}_j - \hat{\Gamma}_{j-1}) \cdot (\hat{\Gamma}_i - \hat{\Gamma}_{i-1})^\perp < 0$  then
                Le := Le  $\cup \{(i, j)\}$ 
            end if
        end if
    end if
end for

```

Algorithm 12: Greedy search to minimize E over the set of combinations of admissible paths

```
( $p_1, \dots, p_n$ ) := ( $1, \dots, 1$ )
 $e_{\min} := E(p_1, \dots, p_n)$ 
 $\mathcal{S}_{\min} := (p_1, \dots, p_n)$ 
while  $\exists i, 1 \leq i \leq n$ , s.t.  $p_i < K_i$  do
     $e_{\min\text{-iter}} := +\infty$ 
    for  $i := 1..n$  do
        if  $p_i < K_i$  then
             $e := E(p_1, \dots, p_i + 1, \dots, p_n)$ 
            if  $e < e_{\min\text{-iter}}$  then
                 $e_{\min\text{-iter}} := e$ 
                 $\mathcal{S}_{\min\text{-iter}} := (p_1, \dots, p_i + 1, \dots, p_n)$ 
            end if
            if  $e < e_{\min}$  then
                 $e_{\min} := e$ 
                 $\mathcal{S}_{\min} := (p_1, \dots, p_i + 1, \dots, p_n)$ 
            end if
        end if
    end if
    end for
    if  $\mathcal{S}_{\min\text{-iter}} := +\infty$  then
         $a_{\min\text{-iter}} := +\infty$ 
        for  $i := 1..n$  do
            if  $p_i < K_i$  then
                 $a := A(p_1, \dots, p_i + 1, \dots, p_n)$ 
                if  $a < a_{\min\text{-iter}}$  then
                     $a_{\min\text{-iter}} := a$ 
                     $\mathcal{S}_{\text{area-min}\text{-iter}} := (p_1, \dots, p_i + 1, \dots, p_n)$ 
                end if
            end if
        end if
    end for
    ( $p_1, \dots, p_n$ ) :=  $\mathcal{S}_{\text{area-min}\text{-iter}}$ 
else
    ( $p_1, \dots, p_n$ ) :=  $\mathcal{S}_{\min\text{-iter}}$ 
end if
end while
```

Appendix B

Video object segmentation

B.1 Derivations involving divergence and entropy

B.1.1 Expectation

Let $\mathbf{f} : \mathcal{D} \rightarrow \mathcal{Q} \subset \mathbb{R}^d$ be our vector-valued image, and $\Omega \in \mathcal{D}$. Let p_Ω be the Dirac mixture representing the empirical density of \mathbf{f} on Ω ,

$$p_\Omega(\mathbf{q}) = \frac{1}{|\Omega|} \int_{\Omega} \delta(\mathbf{q} - \mathbf{f}(\mathbf{x})) \, d\mathbf{x}$$

where δ is the d -dimensional Dirac distribution, such that

$$\int_{\mathcal{Q}} \delta(\mathbf{q}) \, d\mathbf{q} = 1,$$

and, for any $\mathbf{q}_0 \in \mathcal{Q}$ and $\mathbf{g} : \mathcal{Q} \rightarrow \mathbb{R}^p$,

$$\int_{\mathcal{Q}} \delta(\mathbf{q} - \mathbf{q}_0) \mathbf{g}(\mathbf{q}) \, d\mathbf{q} = \mathbf{g}(\mathbf{q}_0).$$

Proposition 30. *For any function $\mathbf{g} : \mathcal{Q} \rightarrow \mathbb{R}^p$, the expectation $\mathbb{E}_{p_\Omega}[\mathbf{g}]$, with respect to density p_Ω , can be written*

$$\mathbb{E}_p[\mathbf{g}] = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{g}(\mathbf{f}(\mathbf{x})) \, d\mathbf{x} \tag{B.1}$$

Proof.

$$\begin{aligned} \mathbb{E}_{p_\Omega}[\mathbf{g}] &= \int_{\mathcal{Q}} p(\mathbf{q}) \mathbf{g}(\mathbf{q}) \, d\mathbf{q} \\ &= \int_{\mathcal{Q}} \left(\frac{1}{|\Omega|} \int_{\Omega} \delta(\mathbf{q} - \mathbf{f}(\mathbf{x})) \, d\mathbf{x} \right) \mathbf{g}(\mathbf{q}) \, d\mathbf{q} \\ &= \frac{1}{|\Omega|} \int_{\Omega} \left(\int_{\mathcal{Q}} \delta(\mathbf{q} - \mathbf{f}(\mathbf{x})) \mathbf{g}(\mathbf{q}) \, d\mathbf{q} \right) \, d\mathbf{x} \\ &= \frac{1}{|\Omega|} \int_{\Omega} \mathbf{g}(\mathbf{f}(\mathbf{x})) \, d\mathbf{x} \end{aligned}$$

□

B.1.2 Data term of [Chiverton et al., 2008]

We prove an equivalence on the data term $E_{\text{data}}^{\text{Chiverton}}$ reviewed in Section 5.2.3. The initial formulation of this data term is

$$\begin{aligned} E_{\text{data}}^{\text{Chiverton}}[\tilde{p}_t^1, (\tilde{p}_{t+1}^i)_{i=0 \dots 1}] &= - \int_{\Omega_{t+1}^0} \log \left(\frac{\tilde{p}_{t+1}^0(\mathbf{f}_{t+1}(\mathbf{x}))}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x} \\ &\quad + \int_{\Omega_{t+1}^1} \log \left(\frac{\tilde{p}_{t+1}^1(\mathbf{f}_{t+1}(\mathbf{x}))}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x}. \end{aligned} \quad (\text{B.2})$$

Let us set (B.1) into the VOS framework. For any region $i = 0 \dots n$, time $t = 1 \dots T$, and function $g : \mathcal{Q} \rightarrow \mathbb{R}$, we have

$$\frac{1}{|\Omega_t^i|} \int_{\Omega_t^i} g(\mathbf{f}_t(\mathbf{x})) d\mathbf{x} = \mathbb{E}_{p_t^i}[g]. \quad (\text{B.3})$$

Using (B.3), the first term of the right-hand side in Eq. (B.2) can be written as a function of expectation with respect to density p_{t+1}^0 ,

$$- \int_{\Omega_{t+1}^0} \log \left(\frac{\tilde{p}_{t+1}^0(\mathbf{f}_{t+1}(\mathbf{x}))}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x} = - |\Omega_{t+1}^0| \mathbb{E}_{p_{t+1}^0} \left[\log \left(\frac{\tilde{p}_{t+1}^0}{\tilde{p}_t^1} \right) \right]$$

which, using Eq. (5.10), is the smooth KL divergence from \tilde{p}_t^1 to p_{t+1}^0 , multiplied by $-|\Omega_{t+1}^0|$,

$$- \int_{\Omega_{t+1}^0} \log \left(\frac{\tilde{p}_{t+1}^0(\mathbf{f}_{t+1}(\mathbf{x}))}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x} = - |\Omega_{t+1}^0| \widetilde{D}_{\text{KL}}(p_{t+1}^0 \| \tilde{p}_t^1).$$

Similarly, the second term of the right-hand side in Eq. (B.2) can be written with respect to the smooth KL divergence from \tilde{p}_t^1 to p_{t+1}^1 ,

$$\begin{aligned} \int_{\Omega_{t+1}^1} \log \left(\frac{\tilde{p}_{t+1}^1(\mathbf{f}_{t+1}(\mathbf{x}))}{\tilde{p}_t^1(\mathbf{f}_{t+1}(\mathbf{x}))} \right) d\mathbf{x} &= |\Omega_{t+1}^1| \mathbb{E}_{p_{t+1}^1} \left[\log \left(\frac{\tilde{p}_{t+1}^1}{\tilde{p}_t^1} \right) \right] \\ &= |\Omega_{t+1}^1| \widetilde{D}_{\text{KL}}(p_{t+1}^1 \| \tilde{p}_t^1). \end{aligned}$$

Finally,

$$E_{\text{data}}^{\text{Chiverton}}[\tilde{p}_t^1, (p_{t+1}^i)_{i=0 \dots 1}] = - |\Omega_{t+1}^0| \widetilde{D}_{\text{KL}}(p_{t+1}^0 \| \tilde{p}_t^1) + |\Omega_{t+1}^1| \widetilde{D}_{\text{KL}}(p_{t+1}^1 \| \tilde{p}_t^1).$$

B.1.3 Data term of the "full-background" model

We prove an equivalence on the data term E_{data} introduced in Section 5.4.1. If we write it as a functional of kernel-density estimates at time t , $(\tilde{p}_t^i)_{i=0 \dots n}$, and Dirac mixtures at time $t+1$, $(p_{t+1}^i)_{i=0 \dots n}$, we have

$$E_{\text{data}}[(\tilde{p}_t^i)_{i=0 \dots n}, (p_{t+1}^i)_{i=0 \dots n}] = - \sum_{i=0}^n \int_{\Omega_{t+1}^i} \log \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x}. \quad (\text{B.4})$$

Using (B.3), each integral over Ω_{t+1}^i in Eq. (B.4) can be written as a function of expectation with respect to density p_{t+1}^i

$$- \int_{\Omega_{t+1}^i} \tilde{p}_t^i(\mathbf{f}_{t+1}(\mathbf{x})) d\mathbf{x} = |\Omega_{t+1}^i| \mathbb{E}_{p_{t+1}^i}[-\log \tilde{p}_t^i],$$

which is the cross entropy of density \tilde{p}_t^i relative to density p_{t+1}^i , multiplied by the area of Ω_{t+1}^i . Hence,

$$E_{\text{data}}[(\tilde{p}_t^i)_{i=0 \dots n}, (p_{t+1}^i)_{i=0 \dots n}] = \sum_{i=0}^n |\Omega_{t+1}^i| \text{CE}(p_{t+1}^i, \tilde{p}_t^i)$$

Appendix C

Mathematical derivations on Euclidean distance-based skeletons

C.1 Derivations related to AOF and ridgeness

C.1.1 Relation between AOF and ridgeness

From Eq. (6.6), we have

$$\text{rdg}(\mathbf{x}, \sigma) = - \int_{\mathbb{R}^2} D(\mathbf{y}) \Delta G_\sigma(\mathbf{x} - \mathbf{y}) d\mathbf{y},$$

which can be written in polar coordinates as

$$\begin{aligned} \text{rdg}(\mathbf{x}, \sigma) &= - \int_0^\infty \rho \Delta G_\sigma(\rho \cos \theta, \rho \sin \theta) \int_0^{2\pi} D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta d\rho \\ &= \frac{1}{\pi \sigma^4} \int_0^\infty \rho \left(1 - \frac{\rho^2}{2\sigma^2}\right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \int_0^{2\pi} D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta d\rho \end{aligned}$$

Considering the following indefinite integral

$$\int \rho \left(1 - \frac{\rho^2}{2\sigma^2}\right) \exp\left(-\frac{\rho^2}{2\sigma^2}\right) d\rho = \frac{\rho^2}{2} \exp\left(-\frac{\rho^2}{2\sigma^2}\right),$$

we integrate by parts w.r.t ρ ,

$$\begin{aligned} \text{rdg}(\mathbf{x}, \sigma) &= \frac{1}{\pi \sigma^4} \left\{ \left[\frac{\rho^2}{2} \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \int_0^{2\pi} D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta \right]_{\rho=0}^{\rho=+\infty} \right. \\ &\quad \left. - \int_0^\infty \frac{\rho^2}{2} \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \int_0^{2\pi} \nabla D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta d\rho \right\} \\ &= -\frac{1}{\pi \sigma^4} \int_0^\infty \frac{\rho^2}{2} \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \int_0^{2\pi} \nabla D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta d\rho \end{aligned}$$

Using Eq. (6.4), it comes

$$\text{rdg}(\mathbf{x}, \sigma) = -\frac{1}{\sigma^4} \int_0^\infty \rho^2 \exp\left(-\frac{\rho^2}{2\sigma^2}\right) \text{aof}(\mathbf{x}, \rho) d\rho$$

which proves Proposition 1.

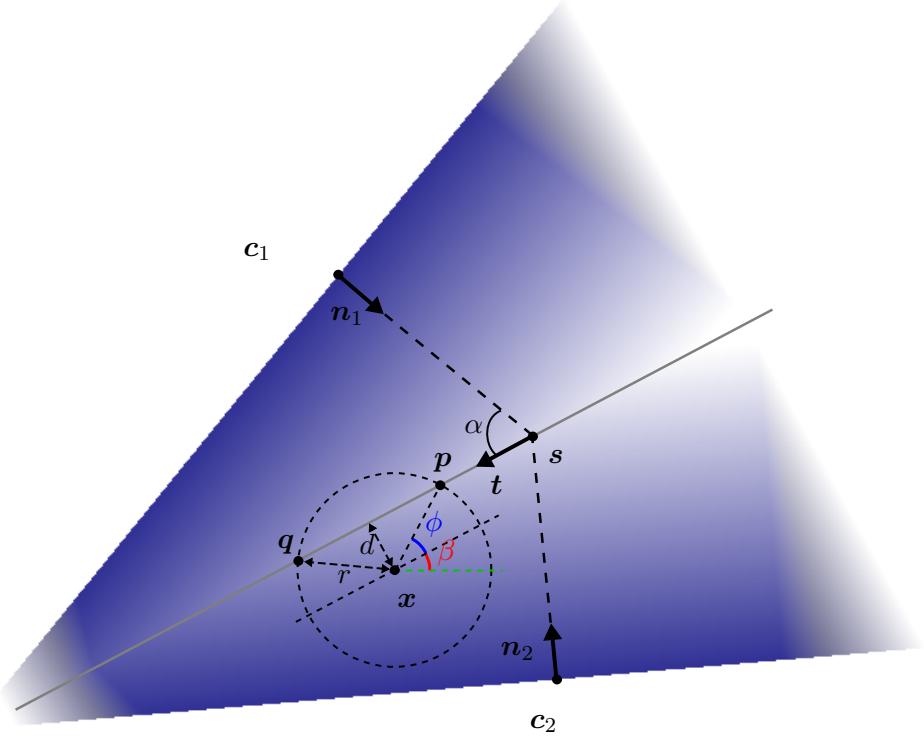


Figure C.1: Regular skeleton point at x

C.1.2 Regular skeleton point - AOF

Let β be the angle formed by t and the horizontal axis, such that

$$\mathbf{t} = \frac{(\mathbf{n}_1 - \mathbf{n}_2)^\perp}{\|\mathbf{n}_1 - \mathbf{n}_2\|} = \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix}$$

Let $d = (x - s) \cdot \mathbf{t}^\perp$ be the distance between x and the nearest point on the skeleton branch. We get rid of the trivial case $d > r$. In this case, $\text{aof}_{\text{regular}}(x) = 0$. Otherwise, the circle of radius r and center x intersects the skeleton branch at points p and q , as depicted in Fig. C.1. We denote by ϕ the angle formed by the line passing through x and p , and the skeleton branch, such that

$$\begin{bmatrix} r \cos \phi \\ r \sin \phi \end{bmatrix} = \begin{bmatrix} \sqrt{r^2 - d^2} \\ d \end{bmatrix}$$

Considering Eq. (6.4), the circle is split into two arcs: the one from angle $\beta + \phi$ to $\beta + \pi - \phi$, on which $\nabla D = \mathbf{n}_1$, and the other one from angle $\beta + \pi - \phi$ to $\beta + \phi + 2\pi$ on which $\nabla D = \mathbf{n}_2$. Hence, it comes

$$\begin{aligned}
\text{aof}_{\text{regular}}(\mathbf{x}) &= \frac{1}{2\pi} \int_0^{2\pi} \nabla D \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \\
&= \frac{1}{2\pi} \left(\int_{\beta+\phi}^{\beta+\pi-\phi} \mathbf{n}_1 \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta + \int_{\beta+\pi-\phi}^{\beta+\phi+2\pi} \mathbf{n}_2 \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right) \\
&= \frac{1}{2\pi} \left(\mathbf{n}_1 \cdot \int_{\beta+\phi}^{\beta+\pi-\phi} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta + \mathbf{n}_2 \cdot \int_{\beta+\pi-\phi}^{\beta+\phi+2\pi} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right) \\
&= \frac{1}{2\pi} \left(\mathbf{n}_1 \cdot \begin{bmatrix} -\sin(\beta-\phi) - \sin(\beta+\phi) \\ \cos(\beta-\phi) + \cos(\beta+\phi) \end{bmatrix} + \mathbf{n}_2 \cdot \begin{bmatrix} \sin(\beta-\phi) + \sin(\beta+\phi) \\ -\cos(\beta-\phi) - \cos(\beta+\phi) \end{bmatrix} \right) \\
&= \frac{1}{2\pi} (\mathbf{n}_1 - \mathbf{n}_2) \cdot \begin{bmatrix} -2 \sin \beta \cos \phi \\ 2 \cos \beta \cos \phi \end{bmatrix} \\
&= \frac{1}{\pi} \cos \phi (\mathbf{n}_1 - \mathbf{n}_2) \cdot \begin{bmatrix} -\sin \beta \\ \cos \beta \end{bmatrix} \\
&= \frac{1}{\pi} \cos \phi (\mathbf{n}_1 - \mathbf{n}_2) \cdot -\frac{\mathbf{n}_1 - \mathbf{n}_2}{\|\mathbf{n}_1 - \mathbf{n}_2\|} \\
&= -\frac{1}{\pi} \frac{\sqrt{r^2 - d^2}}{r} \|\mathbf{n}_1 - \mathbf{n}_2\|
\end{aligned}$$

Using the definition of object angle α in Section 6.2.2, we finally obtain

$$\text{aof}_{\text{regular}}(\mathbf{x}) = -\frac{2 \sin \alpha}{\pi r} \sqrt{r^2 - d^2}$$

which proves Proposition 2.

C.1.3 Regular skeleton point - Ridgeness

The Dirac delta distribution δ has the following properties:

$$\delta(ax + b) = \frac{1}{|a|} \delta \left(y - \frac{b}{a} \right) \quad (\text{C.1})$$

$$\int_{-\infty}^{\infty} \delta(x - c) g(x) dx = g(c) \quad (\text{C.2})$$

To calculate the ridgeness at \mathbf{x} in the vicinity of a regular skeleton point s , consider Eqs. (6.7) and (6.13) simultaneously. Setting $\mathbf{n} = [n_x, n_y]^T = \mathbf{n}_1 - \mathbf{n}_2$ for convenience, we can write

$$\begin{aligned}
\text{rdg}_{\text{regular}}(\mathbf{x}) &= \int_{\mathbb{R}^2} -\Delta D(\mathbf{x} - \mathbf{y}) G_\sigma(\mathbf{y}) d\mathbf{y} \\
&= \int_{\mathbb{R}^2} \frac{\|\mathbf{n}\|^2}{2\pi\sigma^2} \delta((\mathbf{x} - \mathbf{y} - s) \cdot \mathbf{n}) \exp \left(-\frac{\|\mathbf{y}\|^2}{2\sigma^2} \right) d\mathbf{y}
\end{aligned}$$

Splitting over x and y -coordinates and applying rule (C.1) with $a = -n_x$ and $b = (\mathbf{x} - s) \cdot \mathbf{n} - yn_y$, we get

$$\text{rdg}_{\text{regular}}(\mathbf{x}) = \frac{\|\mathbf{n}\|^2}{2\pi\sigma^2 |n_x|} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta \left(x - \frac{(\mathbf{x} - s) \cdot \mathbf{n} - yn_y}{n_x} \right) \exp \left(-\frac{x^2 + y^2}{2\sigma^2} \right) dx dy,$$

which is integrated w.r.t x using rule (C.2),

$$\begin{aligned}\text{rdg}_{\text{regular}}(\mathbf{x}) &= \frac{\|\mathbf{n}\|^2}{2\pi\sigma^2|n_x|} \int_{-\infty}^{\infty} \exp\left(-\frac{((\mathbf{x}-\mathbf{s}) \cdot \mathbf{n} - yn_y)^2}{2\sigma^2 n_x^2} - \frac{y^2}{2\sigma^2}\right) dy \\ &= \frac{\|\mathbf{n}\|^2}{2\pi\sigma^2|n_x|} \exp\left(-\frac{((\mathbf{x}-\mathbf{s}) \cdot \mathbf{n})^2}{2\sigma^2 n_x^2}\right) \int_{-\infty}^{\infty} \exp\left(-\frac{\|\mathbf{n}\|^2}{2\sigma^2 n_x^2} y^2 + \frac{n_y(\mathbf{x}-\mathbf{s}) \cdot \mathbf{n}}{\sigma^2 n_x^2} y\right) dy\end{aligned}$$

Considering the following indefinite integral (with $a > 0$)

$$\int_{-\infty}^{\infty} \exp(-ax^2 + bx) dx = \sqrt{\frac{\pi}{a}} \exp\left(\frac{b^2}{4a}\right)$$

with $a = \frac{\|\mathbf{n}\|^2}{2\sigma^2 n_x^2}$ and $b = \frac{n_y(\mathbf{x}-\mathbf{s}) \cdot \mathbf{n}}{\sigma^2 n_x^2}$, we get

$$\begin{aligned}\text{rdg}_{\text{regular}}(\mathbf{x}) &= \frac{\|\mathbf{n}\|^2}{2\pi\sigma^2|n_x|} \exp\left(-\frac{((\mathbf{x}-\mathbf{s}) \cdot \mathbf{n})^2}{2\sigma^2 n_x^2}\right) \frac{\sigma|n_x| \sqrt{2\pi}}{\|\mathbf{n}\|} \exp\left(\frac{n_y^2((\mathbf{x}-\mathbf{s}) \cdot \mathbf{n})^2}{2\sigma^2 n_x^2 \|\mathbf{n}\|^2}\right) \\ &= \frac{\|\mathbf{n}\|}{\sigma\sqrt{2\pi}} \exp\left(-\frac{((\mathbf{x}-\mathbf{s}) \cdot \mathbf{n})^2}{2\sigma^2 \|\mathbf{n}\|^2}\right)\end{aligned}$$

Using the definition of object angle α in Section 6.2.2, it eventually comes

$$\text{rdg}_{\text{regular}}(\mathbf{x}) = \frac{\sqrt{2\pi} \sin \alpha}{\pi\sigma} \exp\left(-\frac{((\mathbf{x}-\mathbf{s}) \cdot \mathbf{t}^\perp)^2}{2\sigma^2}\right)$$

which proves Proposition 3.

C.1.4 Complementary notes on elliptic integrals

In this section, we provide additional properties of elliptic integrals introduced in Section 6.2.3, allowing to derive the AOF and ridgeness at peaks, endpoints, ligatures and junctions. The following form will be met:

$$\int_0^\psi \frac{\sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta$$

It can be expressed using incomplete elliptic integrals:

$$\begin{aligned}\int_0^\psi \frac{\sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta &= \int_0^\psi \frac{1}{k^2} \frac{k^2 \sin^2 \theta + 1 - 1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta \\ &= \frac{1}{k^2} \int_0^\psi \left(\frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} - \frac{1 - k^2 \sin^2 \theta}{\sqrt{1 - k^2 \sin^2 \theta}} \right) d\theta \\ &= \frac{1}{k^2} \left(\int_0^\psi \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta + \int_0^\psi \sqrt{1 - k^2 \sin^2 \theta} d\theta \right) \\ &= \frac{1}{k^2} (F(\psi, k) - E(\psi, k))\end{aligned}\tag{C.3}$$

Incomplete elliptic integrals with purely imaginary modulus can be transformed into elliptic integrals with real modulus as follows [Carlson, 2010, p. 491],

$$\begin{aligned}F(\psi, ik) &= \frac{1}{\bar{k}} \left(K\left(\frac{k}{\bar{k}}\right) - F\left(\frac{\pi}{2} - \psi, \frac{k}{\bar{k}}\right) \right) \\ E(\psi, ik) &= \bar{k} \left(E\left(\frac{k}{\bar{k}}\right) - E\left(\frac{\pi}{2} - \psi, \frac{k}{\bar{k}}\right) \right)\end{aligned}\tag{C.4}$$

with $i^2 = -1$ and $\check{k} = \sqrt{1 + k^2}$. The corresponding special cases for complete elliptic integrals are

$$K(ik) = \frac{1}{\check{k}} K\left(\frac{k}{\check{k}}\right) \quad E(ik) = \check{k} E\left(\frac{k}{\check{k}}\right) \quad (\text{C.5})$$

The complete elliptic integrals of the first and second kind have the following power series expansions [Carlson, 2010, p. 490]:

$$\begin{aligned} K(k) &= \frac{\pi}{2} \sum_{n=0}^{\infty} \left[\frac{(2n)!}{2^{2n}(n!)^2} \right]^2 k^{2n} = \frac{\pi}{2} + \frac{\pi}{8} k^2 + \frac{9\pi}{128} k^4 + O(k^6) \\ E(k) &= \frac{\pi}{2} \sum_{n=0}^{\infty} \left[\frac{(2n)!}{2^{2n}(n!)^2} \right]^2 \frac{k^{2n}}{1-2n} = \frac{\pi}{2} - \frac{\pi}{8} k^2 - \frac{3\pi}{128} k^4 + O(k^6) \end{aligned} \quad (\text{C.6})$$

The incomplete elliptic integrals of the first and second kind have the following expansions around $k = 0$ [Wolfram Research, a, Wolfram Research, b]:

$$\begin{aligned} F(\psi, k) &= \sum_{n=0}^{\infty} \frac{1}{2^{2n} n!} \left(\frac{1}{2}\right)_n \left(\psi \binom{2n}{n} + \sum_{j=1}^n \frac{(-1)^j}{j} \binom{2n}{n-j} \sin(2j\psi) \right) k^{2n} \\ &= \psi + \frac{2\psi - \sin(2\psi)}{8} k^2 + \frac{3}{256} (12\psi - 8\sin(2\psi) + \sin(4\psi)) k^4 + O(k^6) \\ E(\psi, k) &= \sum_{n=0}^{\infty} \frac{1}{2^{2n} n!} \left(-\frac{1}{2}\right)_n \left(\psi \binom{2n}{n} + \sum_{j=1}^n \frac{(-1)^j}{j} \binom{2n}{n-j} \sin(2j\psi) \right) k^{2n} \\ &= \psi - \frac{2\psi - \sin(2\psi)}{8} k^2 - \frac{1}{256} (12\psi - 8\sin(2\psi) + \sin(4\psi)) k^4 + O(k^6) \end{aligned} \quad (\text{C.7})$$

where $(x)_n$ is the Pochammer symbol for the rising factorial: $(x)_n = x(x+1)(x+2)\dots(x+n-1)$

C.1.5 Peak point - AOF

We switch to a polar coordinate system, centered at \mathbf{x} s.t. $\mathbf{s} = \mathbf{x} + [\mathcal{R} \cos \beta, \mathcal{R} \sin \beta]^\top$. Consider the circle of radius r centered at \mathbf{x} . Using Eq. (6.19), the gradient of the distance at any point on this circle is:

$$\begin{aligned} \nabla D\left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) &= \left\| \begin{bmatrix} \mathcal{R} \cos \beta - r \cos \theta \\ \mathcal{R} \sin \beta - r \sin \theta \end{bmatrix} \right\|^{-1} \begin{bmatrix} \mathcal{R} \cos \beta - r \cos \theta \\ \mathcal{R} \sin \beta - r \sin \theta \end{bmatrix} \\ &= \frac{1}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta)}} \begin{bmatrix} \mathcal{R} \cos \beta - r \cos \theta \\ \mathcal{R} \sin \beta - r \sin \theta \end{bmatrix} \end{aligned}$$

Plugging it into Eq (6.4), it comes

$$\begin{aligned} \text{aof}_{\text{peak}}(\mathbf{x}) &= \frac{1}{2\pi} \int_0^{2\pi} \frac{1}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta)}} \begin{bmatrix} \mathcal{R} \cos \beta - r \cos \theta \\ \mathcal{R} \sin \beta - r \sin \theta \end{bmatrix} \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \frac{-r + \mathcal{R} \cos(\theta - \beta)}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta)}} d\theta \end{aligned}$$

There is translation-invariance in range $[0, 2\pi]$, so $\cos(\theta - \beta)$ can be replaced by $\cos \theta$. Moreover, since $\cos(\pi + \theta) = \cos(\pi - \theta)$,

$$\text{aof}_{\text{peak}}(\mathbf{x}) = \frac{1}{\pi} \int_0^\pi \frac{-r + \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta$$

which proves Proposition 4. Using identity $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and a change of variable,

$$\text{aof}_{\text{peak}}(\mathbf{x}) = \frac{2}{\pi} \int_0^{\frac{\pi}{2}} \frac{-r + \mathcal{R} - 2\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta$$

The integral can be split as

$$\text{aof}_{\text{peak}}(\mathbf{x}) = \frac{2}{\pi} \left[\frac{\mathcal{R} - r}{|\mathcal{R} - r|} \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{1 + m \sin^2 \theta}} d\theta - \frac{2\mathcal{R}}{|\mathcal{R} - r|} \int_0^{\frac{\pi}{2}} \frac{\sin^2 \theta}{\sqrt{1 + m \sin^2 \theta}} d\theta \right]$$

with $m = \frac{4r\mathcal{R}}{(\mathcal{R} - r)^2}$. Using Eqs. (6.21) and (C.3), it can be expressed using the complete elliptic integrals with purely imaginary modulus:

$$\begin{aligned} \text{aof}_{\text{peak}}(\mathbf{x}) &= \frac{2}{\pi} \left[\frac{\mathcal{R} - r}{|\mathcal{R} - r|} K(ik) - \frac{2\mathcal{R}}{|\mathcal{R} - r|(ik)^2} (K(ik) - E(ik)) \right] \\ &= \frac{2}{\pi} \left[\frac{\mathcal{R} - r}{|\mathcal{R} - r|} K(ik) + \frac{|\mathcal{R} - r|}{2r} (K(ik) - E(ik)) \right] \\ &= \frac{1}{\pi r} \left(\frac{\mathcal{R}^2 - r^2}{|\mathcal{R} - r|} K(ik) - |\mathcal{R} - r| E(ik) \right) \end{aligned}$$

where $i^2 = -1$ and $k = \frac{2\sqrt{r\mathcal{R}}}{|\mathcal{R} - r|}$. We convert them to complete elliptic integrals with real modulus using rule (C.5):

$$\begin{aligned} \text{aof}_{\text{peak}}(\mathbf{x}) &= \frac{1}{\pi r} \left(\frac{\mathcal{R}^2 - r^2}{\mathcal{R} + r} K(k) - (\mathcal{R} + r) E(k) \right) \\ &= \frac{1}{\pi r} ((\mathcal{R} - r) K(k) - (\mathcal{R} + r) E(k)) \end{aligned} \quad (\text{C.8})$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$, which proves Proposition 5.

C.1.6 Peak point - AOF - Asymptotical behavior

We consider the second-order Taylor approximation of D , as defined in Eq. (6.18), in the polar coordinate system centered at \mathbf{x} . According to Eq. (6.19),

$$\begin{aligned} \nabla D(\mathbf{x}) &= \frac{\mathbf{s} - \mathbf{x}}{\|\mathbf{s} - \mathbf{x}\|} \\ \nabla^2 D(\mathbf{x}) &= -\frac{(\mathbf{s} - \mathbf{x})^\perp ((\mathbf{s} - \mathbf{x})^\perp)^\top}{\|\mathbf{s} - \mathbf{x}\|^3}, \end{aligned}$$

Since $\mathbf{s} = [\mathcal{R} \cos \beta, \mathcal{R} \sin \beta]^\top$, we have

$$\begin{aligned} \nabla D(\mathbf{x}) &= [\cos \beta, \sin \beta]^\top \\ \nabla^2 D(\mathbf{x}) &= \frac{1}{\mathcal{R}} \begin{bmatrix} -\sin^2 \beta & \cos \beta \sin \beta \\ \cos \beta \sin \beta & -\cos^2 \beta \end{bmatrix} \end{aligned}$$

So, the approximate distance on the circle of radius r centered at \mathbf{x} is

$$\begin{aligned} &\tilde{D} \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \\ &= D(\mathbf{s}) - \mathcal{R} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla D(\mathbf{x}) + \frac{1}{2} \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla^2 D(\mathbf{x}) \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= D(\mathbf{s}) - \mathcal{R} + r \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix} + \frac{r^2}{2\mathcal{R}} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} -\sin^2 \beta & \cos \beta \sin \beta \\ \cos \beta \sin \beta & -\cos^2 \beta \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \\ &= D(\mathbf{s}) - \mathcal{R} + r \cos(\theta - \beta) - \frac{r^2}{2\mathcal{R}} \sin^2(\theta - \beta) \end{aligned} \quad (\text{C.9})$$

and its gradient is

$$\begin{aligned}\nabla \tilde{D} \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) &= \nabla D(\mathbf{x}) + \nabla^2 D(\mathbf{x}) \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta \\ \sin \beta \end{bmatrix} + \frac{r}{\mathcal{R}} \begin{bmatrix} -\sin^2 \beta & \cos \beta \sin \beta \\ \cos \beta \sin \beta & -\cos^2 \beta \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}\end{aligned}$$

Plugging it into Eq. (6.26), we obtain

$$\begin{aligned}\widetilde{\text{aof}}_{\text{peak}}(\mathbf{x}) &= \frac{1}{2\pi} \int_0^{2\pi} \nabla \tilde{D} \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \cos(\theta - \beta) - \frac{r}{\mathcal{R}} \sin^2(\theta - \beta) d\theta \\ &= \frac{1}{2\pi} \int_0^{2\pi} \cos \theta - \frac{r}{\mathcal{R}} \sin^2 \theta d\theta \\ &= -\frac{r}{2\mathcal{R}}\end{aligned}$$

which proves Proposition 6. In order to prove that the AOF near a peak point is asymptotically equivalent to its second-order approximation

$$\text{aof}_{\text{peak}}(\mathbf{x}) \sim \widetilde{\text{aof}}_{\text{peak}}(\mathbf{x}) \text{ (as } \mathcal{R} \rightarrow +\infty\text{)},$$

we show that

$$\lim_{\mathcal{R} \rightarrow +\infty} -\frac{2\mathcal{R}}{r} \text{aof}_{\text{peak}}(\mathbf{x}) = 1.$$

We have

$$\begin{aligned}K \left(\frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r} \right) &= \frac{\pi}{2} \left(1 + \frac{4\mathcal{R}^3 r + 17\mathcal{R}^2 r^2 + 4\mathcal{R} r^3}{4(\mathcal{R} + r)^4} \right) + O \left(\frac{1}{\mathcal{R}^3} \right) \\ E \left(\frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r} \right) &= \frac{\pi(\mathcal{R} + 2r)(2\mathcal{R} + r)(2\mathcal{R}^2 + 2r^2 + \mathcal{R}r)}{8(\mathcal{R} + r)^4} + O \left(\frac{1}{\mathcal{R}^3} \right)\end{aligned}$$

Thanks to the power series expansions of the complete elliptic integrals in Eq. (C.6), we can derive from Eq. (C.8),

$$\text{aof}_{\text{peak}}(\mathbf{x}) = -\frac{r(2\mathcal{R}^3 + 23\mathcal{R}^2 r + 16\mathcal{R} r^2 + 4r^3)}{4(\mathcal{R} + r)^4} + O \left(\frac{1}{\mathcal{R}^2} \right).$$

Thus,

$$\lim_{\mathcal{R} \rightarrow +\infty} -\frac{2\mathcal{R}}{r} \text{aof}_{\text{peak}}(\mathbf{x}) = \lim_{\mathcal{R} \rightarrow +\infty} \frac{\mathcal{R}(2\mathcal{R}^3 + 23\mathcal{R}^2 r + 16\mathcal{R} r^2 + 4r^3)}{2(\mathcal{R} + r)^4} + O \left(\frac{1}{\mathcal{R}} \right) = 1,$$

which proves Proposition 7.

C.1.7 Peak point - Ridgeness

As for the AOF, using a polar transformation of the Laplacian of the distance defined in Eq. (6.19), we have:

$$\Delta D \left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) = -\left\| \begin{bmatrix} \mathcal{R} \cos \beta - \rho \cos \theta \\ \mathcal{R} \sin \beta - \rho \sin \theta \end{bmatrix} \right\|^{-1} = -\frac{1}{\sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos(\theta - \beta)}}$$

Plugging it into Eq. (6.27), we get

$$\begin{aligned}\text{rdg}_{\text{peak}}(\mathbf{x}) &= -\int_0^\infty \rho G_\sigma(\rho) \int_0^{2\pi} \Delta D \left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta d\rho \\ &= \int_0^\infty \rho G_\sigma(\rho) \int_0^{2\pi} \frac{1}{\sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos(\theta - \beta)}} d\theta d\rho\end{aligned}$$

Again, by translation invariance in range $[0, 2\pi]$, $\cos(\theta - \beta)$ can be replaced by $\cos \theta$. Using the evenness of \cos ,

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 2 \int_0^\infty \rho G_\sigma(\rho) \int_0^\pi \frac{1}{\sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R}\cos\theta}} d\theta d\rho$$

which proves Proposition 8. Using identity $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and a change of variable, we get

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho G_\sigma(\rho) \int_0^{\frac{\pi}{2}} \frac{1}{\sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta}} d\theta d\rho$$

which, by a derivation similar to the one in Section C.1.5, can be expressed using the complete elliptic integral of the first kind, with complex modulus:

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho G_\sigma(\rho) \frac{1}{|\mathcal{R} - \rho|} K\left(i \frac{2\sqrt{\rho\mathcal{R}}}{|\mathcal{R} - \rho|}\right) d\rho$$

Using transformation (C.5), we finally obtain:

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho G_\sigma(\rho) \frac{1}{\mathcal{R} + \rho} K\left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}\right) d\rho$$

which proves Proposition 9. Using Eqs (6.18) and (6.6) in polar coordinates, an alternative expression of the ridgeness near a peak point can be obtained:

$$\begin{aligned} \text{rdg}_{\text{peak}}(\mathbf{x}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D\left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D(\mathbf{s}) - \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R}\cos(\theta - \beta)} d\theta d\rho \end{aligned}$$

Since $\int_0^\infty \rho \Delta G_\sigma(\rho) = 0$, the constant term $D(\mathbf{s})$ can be dropped, leading to

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R}\cos(\theta - \beta)} d\theta d\rho$$

Again, by translation invariance in range $[0, 2\pi]$, $\cos(\theta - \beta)$ can be replaced by $\cos \theta$. Using the evenness of \cos ,

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 2 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^\pi \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R}\cos\theta} d\theta d\rho$$

Using identity $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and a change of variable, we get

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{\frac{\pi}{2}} \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} d\theta d\rho$$

which, by a derivation similar to the previous one, can be expressed using the complete elliptic integral of the second kind, with complex modulus:

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho \Delta G_\sigma(\rho) |\mathcal{R} - \rho| E\left(i \frac{2\sqrt{\rho\mathcal{R}}}{|\mathcal{R} - \rho|}\right) d\rho$$

Using transformation (C.5), we finally obtain:

$$\text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho \Delta G_\sigma(\rho) (\mathcal{R} + \rho) E\left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}\right) d\rho \quad (\text{C.10})$$

C.1.8 Peak point - Ridgeness - Asymptotical behavior

Using Eqs (6.30) and (C.9), the approximate ridgeness in the vicinity of a peak point is

$$\begin{aligned}\widetilde{\text{rdg}}_{\text{peak}}(\mathbf{x}) &= - \int_0^\infty \int_0^{2\pi} \rho \Delta G_\sigma(\rho) \tilde{D} \left(\mathbf{x} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D(\mathbf{s}) - \mathcal{R} + \rho \cos(\theta - \beta) - \frac{\rho^2}{2\mathcal{R}} \sin^2(\theta - \beta) d\theta d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D(\mathbf{s}) - \mathcal{R} + \rho \cos(\theta - \beta) - \frac{\rho^2}{2\mathcal{R}} \sin^2(\theta - \beta) d\theta d\rho\end{aligned}$$

By translation,

$$\begin{aligned}\widetilde{\text{rdg}}_{\text{peak}}(\mathbf{x}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D(\mathbf{s}) - \mathcal{R} + \rho \cos \theta - \frac{\rho^2}{2\mathcal{R}} \sin^2 \theta d\theta d\rho \\ &= - \int_0^\infty \Delta G_\sigma(\rho) \left(2\pi \rho (D(\mathbf{s}) - \mathcal{R}) - \rho^3 \frac{\pi}{2\mathcal{R}} \right) d\rho\end{aligned}$$

Using definite integrals $\int_0^\infty \rho \Delta G_\sigma(\rho) d\rho = 0$ and $\int_0^\infty \rho^3 \Delta G_\sigma(\rho) d\rho = \frac{2}{\pi}$, we finally obtain

$$\widetilde{\text{rdg}}_{\text{peak}}(\mathbf{x}) = \frac{1}{\mathcal{R}}$$

which proves Proposition 10. In order to prove that the ridgeness near a peak point is asymptotically equivalent to its second-order approximation

$$\text{rdg}_{\text{peak}}(\mathbf{x}) \sim \widetilde{\text{rdg}}_{\text{peak}}(\mathbf{x}) \text{ (as } \mathcal{R} \rightarrow +\infty\text{)},$$

we show that

$$\lim_{\mathcal{R} \rightarrow +\infty} \mathcal{R} \text{rdg}_{\text{peak}}(\mathbf{x}) = 1.$$

From Eq. (C.10), we can write

$$\mathcal{R} \text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho \Delta G_\sigma(\rho) \mathcal{R} (\mathcal{R} + \rho) E \left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho} \right) d\rho$$

Since

$$\int_0^\infty \rho \Delta G_\sigma(\rho) d\rho = 0,$$

it is also true that

$$\int_0^\infty \frac{\pi}{2} \mathcal{R}^2 \rho \Delta G_\sigma(\rho) d\rho = 0$$

Hence,

$$\mathcal{R} \text{rdg}_{\text{peak}}(\mathbf{x}) = 4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\mathcal{R} (\mathcal{R} + \rho) E \left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho} \right) - \frac{\pi}{2} \mathcal{R}^2 \right] d\rho$$

Thanks to the power series expansions of E in Eq. (C.6), we can derive from Eq. (C.10),

$$\begin{aligned}h(\mathcal{R}, \rho) &= \mathcal{R} (\mathcal{R} + \rho) E \left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho} \right) - \frac{\pi}{2} \mathcal{R}^2 \\ &= \mathcal{R} (\mathcal{R} + \rho) \left(\frac{\pi}{2} - \frac{\pi\rho\mathcal{R}}{2(\mathcal{R} + \rho)^2} - \frac{3\pi\rho^2\mathcal{R}^2}{8(\mathcal{R} + \rho)^4} \right) - \frac{\pi}{2} \mathcal{R}^2 + O \left(\frac{1}{\mathcal{R}} \right) \\ &= \frac{\pi\mathcal{R}\rho^2(\mathcal{R}^2 + 4\rho^2 + 8\mathcal{R}\rho)}{8(\mathcal{R} + \rho)^3} + O \left(\frac{1}{\mathcal{R}} \right)\end{aligned}$$

and thus,

$$\lim_{\mathcal{R} \rightarrow +\infty} h(\mathcal{R}, \rho) = \frac{\pi}{8} \rho^2.$$

For every $\rho \in [0, +\infty)$, $\rho \Delta G_\sigma(\rho) h(\mathcal{R}, \rho)$ is dominated by $\max(1, \rho^3) |\Delta G_\sigma(\rho)|$, which is L^1 -integrable. Hence, by the DCT,

$$\begin{aligned} \lim_{\mathcal{R} \rightarrow +\infty} 4 \int_0^\infty \rho \Delta G_\sigma(\rho) h(\mathcal{R}, \rho) \, d\rho &= 4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left(\lim_{\mathcal{R} \rightarrow +\infty} h(\mathcal{R}, \rho) \right) \, d\rho \\ &= \frac{\pi}{2} \int_0^\infty \rho^3 \Delta G_\sigma(\rho) \, d\rho \\ &= 1 \end{aligned}$$

Thus,

$$\lim_{\mathcal{R} \rightarrow +\infty} \mathcal{R} \text{rdg}_{\text{peak}}(\mathbf{x}) = 1,$$

which proves Proposition 11.

C.1.9 End point - AOF

The AOF being rotation-invariant, we perform the following derivations considering that \mathbf{t} is aligned with the horizontal axis (see Fig. 6.8a). This simplifies the derivation and remains valid for any orientation of \mathbf{t} . Considering Eq. (6.32), the circle of center \mathbf{s} and radius r is split into three arcs:

- the one from angle $-\alpha$ to α , on which $\nabla D = [-\cos \theta, -\sin \theta]^\top$,
- the one from angle α to $\pi - \alpha$, on which $\nabla D = [-\cos \alpha, -\sin \alpha]^\top$,
- and the one from angle $\pi - \alpha$ to $2\pi - \alpha$, on which $\nabla D = [-\cos \alpha, \sin \alpha]^\top$

In this case, Eq. (6.4) gives:

$$\begin{aligned} \text{aof}_{\text{end}}(\mathbf{s}) &= \frac{1}{2\pi} \left(\int_{-\alpha}^{\alpha} \begin{bmatrix} -\cos \theta \\ -\sin \theta \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \, d\theta + \int_{\alpha}^{\pi} \begin{bmatrix} -\cos \alpha \\ -\sin \alpha \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \, d\theta \right. \\ &\quad \left. + \int_{\pi}^{2\pi-\alpha} \begin{bmatrix} -\cos \alpha \\ \sin \alpha \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \, d\theta \right) \\ &= \frac{1}{2\pi} \left(- \int_{-\alpha}^{\alpha} 1 \, d\theta - \int_{\alpha}^{\pi} \cos(\theta - \alpha) \, d\theta - \int_{\pi}^{2\pi-\alpha} \cos(\theta + \alpha) \, d\theta \right) \\ &= -\frac{1}{\pi} (\alpha + \sin \alpha) \end{aligned}$$

which proves Proposition 12.

C.1.10 End point - Ridgeness

Starting from a polar transformation of Eq. (6.6), as in Section C.1.9, we use the rotation invariance property of the ridgeness. The integral w.r.t to angle θ is split according to the three arcs. Using Eq. (6.31), we write

$$\begin{aligned} \text{rdg}_{\text{end}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D\left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \left(\int_{-\alpha}^{\alpha} D(\mathbf{s}) - \rho d\theta + \int_{\alpha}^{\pi} D(\mathbf{s}) + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}^\top \begin{bmatrix} -\cos \alpha \\ -\sin \alpha \end{bmatrix} d\theta \right. \\ &\quad \left. + \int_{\pi}^{2\pi-\alpha} D(\mathbf{s}) + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}^\top \begin{bmatrix} -\cos \alpha \\ \sin \alpha \end{bmatrix} d\theta \right) d\rho \\ &= \int_0^\infty \rho \Delta G_\sigma(\rho) \left(2\pi D(\mathbf{s}) + \int_{-\alpha}^{\alpha} \rho d\theta + \int_{\alpha}^{\pi} \rho \cos(\theta - \alpha) d\theta \right. \\ &\quad \left. + \int_{\pi}^{2\pi-\alpha} \rho \cos(\theta + \alpha) d\theta \right) d\rho \end{aligned}$$

Using definite integrals $\int_0^{+\infty} \rho \Delta G_\sigma(\rho) d\rho = 0$ and $\int_0^{+\infty} \rho^2 \Delta G_\sigma(\rho) d\rho = \frac{1}{2\sigma\sqrt{2\pi}}$, we finally obtain

$$\text{rdg}_{\text{end}}(\mathbf{s}) = \int_0^\infty \rho^2 \Delta G_\sigma(\rho) (2\alpha + 2 \sin \alpha) d\rho = \frac{\sqrt{2\pi}}{2\pi\sigma} (\alpha + \sin \alpha)$$

which proves Proposition 13.

C.1.11 Ligature point - AOF

Consider Fig. 6.9a. Since the average outward flux is rotation-invariant, we can perform the derivation assuming that \mathbf{t} is aligned with the horizontal axis, for convenience. Under this assumption, $\mathbf{p} = \mathbf{s} + [\mathcal{A}, \mathcal{B}]^\top$ and $\mathbf{q} = \mathbf{s} + \mathcal{R}[\mathcal{A}, -\mathcal{B}]^\top$. The distance defined in Eq. (6.35) now reads

$$D\left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) = \begin{cases} \left\| \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta - \mathcal{B} \end{bmatrix} \right\| & \text{if } 0 \leq \theta \leq \pi \\ \left\| \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta + \mathcal{B} \end{bmatrix} \right\| & \text{if } \pi \leq \theta \leq 2\pi \end{cases} \quad (\text{C.11})$$

and its gradient is

$$\nabla D\left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) = \begin{cases} \left\| \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta - \mathcal{B} \end{bmatrix} \right\|^{-1} \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta - \mathcal{B} \end{bmatrix} & \text{if } 0 < \theta < \pi \\ \left\| \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta + \mathcal{B} \end{bmatrix} \right\|^{-1} \begin{bmatrix} r \cos \theta - \mathcal{A} \\ r \sin \theta + \mathcal{B} \end{bmatrix} & \text{if } \pi < \theta < 2\pi \end{cases}$$

Plugging it into Eq (6.4), it comes

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{2\pi} \left[\int_0^\pi \nabla D\left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right. \\ &\quad \left. + \int_\pi^{2\pi} \nabla D\left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right] \\ &= \frac{1}{2\pi} \left[\int_0^\pi \frac{r - \mathcal{A} \cos \theta - \mathcal{B} \sin \theta}{\sqrt{r^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2r(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)}} d\theta \right. \\ &\quad \left. + \int_\pi^{2\pi} \frac{r - \mathcal{A} \cos \theta + \mathcal{B} \sin \theta}{\sqrt{r^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2r(\mathcal{A} \cos \theta - \mathcal{B} \sin \theta)}} d\theta \right] \end{aligned}$$

Using property $\cos(\pi - x) = -\cos x$ and $\sin(\pi - x) = \sin x$, we get

$$\text{aof}_{\text{ligature}}(\mathbf{s}) = \frac{1}{\pi} \int_0^\pi \frac{r - \mathcal{A} \cos \theta - \mathcal{B} \sin \theta}{\sqrt{r^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2r(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)}} d\theta$$

which proves Proposition 14. From Eq (6.36), $\mathcal{A} = \mathcal{R} \cos \beta$ and $\mathcal{B} = \mathcal{R} \sin \beta$, which gives

$$\text{aof}_{\text{ligature}}(\mathbf{s}) = \frac{1}{\pi} \int_0^\pi \frac{r - \mathcal{R} \cos(\theta - \beta)}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta)}} d\theta.$$

By translation,

$$\text{aof}_{\text{ligature}}(\mathbf{s}) = \frac{1}{\pi r} \int_{-\beta}^{\pi - \beta} \frac{r^2 - r\mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta.$$

Using identity $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and a change of variable,

$$\text{aof}_{\text{ligature}}(\mathbf{s}) = \frac{2}{\pi r} \int_{-\frac{\beta}{2}}^{\frac{\pi}{2} - \frac{\beta}{2}} \frac{r^2 - r\mathcal{R} + 2r\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta.$$

The integral can be split as

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{2}{\pi r} \left(\int_{-\frac{\beta}{2}}^0 \frac{r^2 - r\mathcal{R} + 2r\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta + \int_0^{\frac{\pi}{2} - \frac{\beta}{2}} \frac{r^2 - r\mathcal{R} + 2r\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta \right) \\ &= \frac{2}{\pi r} \left(\int_0^{\frac{\beta}{2}} \frac{r^2 - r\mathcal{R} + 2r\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta + \int_0^{\frac{\pi}{2} - \frac{\beta}{2}} \frac{r^2 - r\mathcal{R} + 2r\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta \right) \end{aligned}$$

Using Eqs. (6.20) and (C.3), it can be expressed with incomplete elliptic integrals with purely imaginary modulus,

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{\pi r} \left(|\mathcal{R} - r| \left(E\left(\frac{\beta}{2}, ik\right) + E\left(\frac{\pi}{2} - \frac{\beta}{2}, ik\right) \right) \right. \\ &\quad \left. - \frac{\mathcal{R}^2 - r^2}{|\mathcal{R} - r|} \left(F\left(\frac{\beta}{2}, ik\right) + F\left(\frac{\pi}{2} - \frac{\beta}{2}, ik\right) \right) \right) \end{aligned}$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{|\mathcal{R} - r|}$. Using transformations in Eq. (C.4), we convert them to elliptic integrals with real modulus

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{\pi r} \left[(\mathcal{R} + r) \left(2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \right. \\ &\quad \left. - (\mathcal{R} - r) \left(2K(k) - F\left(\frac{\beta}{2}, k\right) - F\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \right] \end{aligned} \tag{C.12}$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$, which proves Proposition 15.

C.1.12 Ligature point - AOF - Asymptotical behavior

We consider the second-order Taylor approximation of D , in the polar coordinate system centered at s . From Eq. (C.11), we derive the approximate distance on the circle of radius r centered at s :

$$\begin{aligned}
& \tilde{D} \left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \\
&= D(s) + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla D(s) + \frac{1}{2} \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla^2 D(s) \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \\
&= \begin{cases} \mathcal{R} + \frac{r}{\mathcal{R}} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} -\mathcal{A} \\ -\mathcal{B} \end{bmatrix} + \frac{r^2}{2\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & -\mathcal{A}\mathcal{B} \\ -\mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & \text{if } 0 \leq \theta \leq \pi \\ \mathcal{R} + \frac{r}{\mathcal{R}} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} -\mathcal{A} \\ \mathcal{B} \end{bmatrix} + \frac{r^2}{2\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & \mathcal{A}\mathcal{B} \\ \mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & \text{if } \pi \leq \theta \leq 2\pi \end{cases} \\
&= \begin{cases} \mathcal{R} + \frac{r}{\mathcal{R}}(-\mathcal{A} \cos \theta - \mathcal{B} \sin \theta) + \frac{r^2}{2\mathcal{R}^3}(\mathcal{B} \cos \theta - \mathcal{A} \sin \theta)^2 & \text{if } 0 \leq \theta \leq \pi \\ \mathcal{R} + \frac{r}{\mathcal{R}}(-\mathcal{A} \cos \theta + \mathcal{B} \sin \theta) + \frac{r^2}{2\mathcal{R}^3}(\mathcal{B} \cos \theta + \mathcal{A} \sin \theta)^2 & \text{if } \pi \leq \theta \leq 2\pi \end{cases} \tag{C.13}
\end{aligned}$$

The approximate gradient on the circle is

$$\nabla \tilde{D} \left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) = \begin{cases} \frac{1}{\mathcal{R}} \begin{bmatrix} -\mathcal{A} \\ -\mathcal{B} \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \mathcal{B}^2 & -\mathcal{A}\mathcal{B} \\ -\mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & \text{if } 0 \leq \theta \leq \pi \\ \frac{1}{\mathcal{R}} \begin{bmatrix} -\mathcal{A} \\ \mathcal{B} \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \mathcal{B}^2 & \mathcal{A}\mathcal{B} \\ \mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} & \text{if } \pi \leq \theta \leq 2\pi \end{cases}$$

Plugging it into Eq. (6.26), we obtain

$$\begin{aligned}
\widetilde{\text{aof}}_{\text{ligature}}(\mathbf{x}) &= \frac{1}{2\pi} \int_0^{2\pi} \nabla \tilde{D} \left(\mathbf{x} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \\
&= \frac{1}{2\pi} \left[\int_0^\pi \frac{1}{\mathcal{R}} \begin{bmatrix} -\mathcal{A} \\ -\mathcal{B} \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & -\mathcal{A}\mathcal{B} \\ -\mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right. \\
&\quad \left. + \int_\pi^{2\pi} \frac{1}{\mathcal{R}} \begin{bmatrix} -\mathcal{A} \\ \mathcal{B} \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & \mathcal{A}\mathcal{B} \\ \mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right]
\end{aligned}$$

Using $\cos(x + \pi) = -\cos x$ and $\sin(x + \pi) = -\sin x$, it comes

$$\begin{aligned}
\widetilde{\text{aof}}_{\text{ligature}}(\mathbf{x}) &= \frac{1}{2\pi} \left[\int_0^\pi \frac{1}{\mathcal{R}} \begin{bmatrix} -\mathcal{A} \\ -\mathcal{B} \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & -\mathcal{A}\mathcal{B} \\ -\mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right. \\
&\quad \left. + \int_0^\pi \frac{1}{\mathcal{R}} \begin{bmatrix} \mathcal{A} \\ -\mathcal{B} \end{bmatrix}^\top \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} + \frac{r}{\mathcal{R}^3} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \mathcal{B}^2 & \mathcal{A}\mathcal{B} \\ \mathcal{A}\mathcal{B} & \mathcal{A}^2 \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right] \\
&= \frac{1}{2\pi} \int_0^\pi -\frac{2\mathcal{B}}{\mathcal{R}} \sin \theta + \frac{r}{\mathcal{R}} d\theta \\
&= \frac{1}{\sqrt{\mathcal{A}^2 + \mathcal{B}^2}} \left(\frac{r}{2} - \frac{2\mathcal{B}}{\pi} \right)
\end{aligned}$$

which proves Proposition 16. In order to prove that the AOF at a ligature point is asymptotically equivalent to its second-order approximation

$$\text{aof}_{\text{ligature}}(\mathbf{s}) \sim \widetilde{\text{aof}}_{\text{ligature}}(\mathbf{s}) \text{ (as } \mathcal{A} \rightarrow +\infty\text{)},$$

we show that

$$\lim_{\mathcal{A} \rightarrow +\infty} \frac{2\pi\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\pi r - 4\mathcal{B}} \text{aof}_{\text{ligature}}(\mathbf{s}) = 1.$$

Using series expansion in Eqs. (C.6) and (C.7), we derive the elliptic terms of Eq. (C.12)

$$\begin{aligned} 2K(k) - F\left(\frac{\beta}{2}, k\right) - F\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) &= \frac{\pi}{2} + k^2 \left(\frac{\pi}{8} + \frac{\sin \beta}{4}\right) + k^4 \left(\frac{9\pi}{128} + \frac{3\sin \beta}{16}\right) + O(k^6) \\ 2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) &= \frac{\pi}{2} - k^2 \left(\frac{\pi}{8} + \frac{\sin \beta}{4}\right) - k^4 \left(\frac{3\pi}{128} + \frac{\sin \beta}{16}\right) + O(k^6) \end{aligned}$$

which yields

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{(\mathcal{R} + r)^4} \left(-\frac{2\sin \beta}{\pi} \mathcal{R}^4 + \left(\frac{1}{2} - \frac{8\sin \beta}{\pi}\right) \mathcal{R}^3 r \right. \\ &\quad \left. + \frac{23}{4} \mathcal{R}^2 r^2 + 4\mathcal{R}r^3 + r^4 \right) + O\left(\frac{1}{\mathcal{R}^2}\right) \end{aligned}$$

From definitions of \mathcal{R} and β in Eq. (6.36), it comes

$$\begin{aligned} \text{aof}_{\text{ligature}}(\mathbf{s}) &= \frac{1}{\left(\sqrt{\mathcal{A}^2 + \mathcal{B}^2} + r\right)^4} \left((\mathcal{A}^2 + \mathcal{B}^2)^{\frac{3}{2}} \left(\frac{r}{2} - \frac{2\mathcal{B}}{\pi}\right) + (\mathcal{A}^2 + \mathcal{B}^2) \left(\frac{23}{4}r^2 - \frac{8\mathcal{B}}{\pi}r\right) \right. \\ &\quad \left. + 4r^3 \sqrt{\mathcal{A}^2 + \mathcal{B}^2} + r^4 \right) + O\left(\frac{1}{\mathcal{A}^2}\right) \end{aligned}$$

Thus,

$$\lim_{\mathcal{A} \rightarrow +\infty} \frac{2\pi\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\pi r - 4\mathcal{B}} \text{aof}_{\text{ligature}}(\mathbf{s}) = \lim_{\mathcal{A} \rightarrow +\infty} \frac{(\mathcal{A}^2 + \mathcal{B}^2)^2}{\left(\sqrt{\mathcal{A}^2 + \mathcal{B}^2} + r\right)^4} + O\left(\frac{1}{\mathcal{A}}\right) = 1$$

which proves Proposition 17.

C.1.13 Ligature point - Ridgeness

From Eqs (6.38) and (C.11), it comes

$$\begin{aligned} \text{rdg}_{\text{ligature}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} D\left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix}\right) d\theta d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\int_0^\pi \sqrt{\rho^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2\rho(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)} d\theta \right. \\ &\quad \left. + \int_\pi^{2\pi} \sqrt{\rho^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2\rho(\mathcal{A} \cos \theta - \mathcal{B} \sin \theta)} d\theta \right] d\rho \end{aligned}$$

Using properties $\cos(\pi - x) = -\cos x$ and $\sin(\pi - x) = \sin x$, we have

$$\text{rdg}_{\text{ligature}}(\mathbf{s}) = -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^\pi \sqrt{\rho^2 + \mathcal{A}^2 + \mathcal{B}^2 - 2\rho(\mathcal{A} \cos \theta + \mathcal{B} \sin \theta)} d\theta d\rho$$

which proves Proposition 18. From Eq (6.36), $\mathcal{A} = \mathcal{R} \cos \beta$ and $\mathcal{B} = \mathcal{R} \sin \beta$, which gives

$$\text{rdg}_{\text{ligature}}(\mathbf{s}) = -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^\pi \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho \mathcal{R} \cos(\theta - \beta)} d\theta d\rho$$

By translation,

$$\text{rdg}_{\text{ligature}}(\mathbf{s}) = -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_{-\beta}^{\pi - \beta} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho \mathcal{R} \cos \theta} d\theta d\rho$$

Using transformation $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$,

$$\text{rdg}_{\text{ligature}}(s) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \int_{-\frac{\beta}{2}}^{\frac{\pi}{2} - \frac{\beta}{2}} \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} \, d\theta \, d\rho.$$

Splitting the integral w.r.t θ , we obtain

$$\begin{aligned} \text{rdg}_{\text{ligature}}(s) &= -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\int_{-\frac{\beta}{2}}^0 \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} \, d\theta \right. \\ &\quad \left. + \int_0^{\frac{\pi}{2} - \frac{\beta}{2}} \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} \, d\theta \right] \, d\rho \\ &= -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\int_0^{\frac{\beta}{2}} \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} \, d\theta \right. \\ &\quad \left. + \int_0^{\frac{\pi}{2} - \frac{\beta}{2}} \sqrt{(\rho - \mathcal{R})^2 + 4\rho\mathcal{R}\sin^2\theta} \, d\theta \right] \, d\rho \end{aligned}$$

which can be expressed with incomplete elliptic integrals of the second kind with purely imaginary modulus

$$\text{rdg}_{\text{ligature}}(s) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) |\mathcal{R} - \rho| \left(E\left(\frac{\beta}{2}, k\right) + E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \, d\rho$$

with $k = \frac{2\sqrt{\rho\mathcal{R}}}{|\mathcal{R} - \rho|}$. Using transformations in Eq. (C.4), we convert them to elliptic integrals of the second kind with real modulus

$$\text{rdg}_{\text{ligature}}(s) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) (\mathcal{R} + \rho) \left(2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right) \, d\rho \quad (\text{C.14})$$

with $k = \frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}$, which proves Proposition 19.

C.1.14 Ligature point - Ridgeness - Asymptotical behavior

From Eqs. (6.30) and (C.13), the approximate ridgeness for ligature point s is derived as

$$\begin{aligned} \widetilde{\text{rdg}}_{\text{ligature}}(s) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \int_0^{2\pi} \tilde{D} \left(s + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) \, d\theta \, d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\int_0^\pi \frac{\rho}{\mathcal{R}} (-\mathcal{A} \cos \theta - \mathcal{B} \sin \theta) + \frac{\rho^2}{2\mathcal{R}^3} (\mathcal{B} \cos \theta - \mathcal{A} \sin \theta)^2 \, d\theta \right. \\ &\quad \left. + \int_\pi^{2\pi} \frac{\rho}{\mathcal{R}} (-\mathcal{A} \cos \theta + \mathcal{B} \sin \theta) + \frac{\rho^2}{2\mathcal{R}^3} (\mathcal{B} \cos \theta + \mathcal{A} \sin \theta)^2 \, d\theta + 2\pi\mathcal{R} \right] \, d\rho \end{aligned}$$

Using $\cos(x + \pi) = -\cos x$ and $\sin(x + \pi) = -\sin x$, it comes

$$\begin{aligned} \widetilde{\text{rdg}}_{\text{ligature}}(s) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \left[\int_0^\pi \frac{\rho}{\mathcal{R}} (-\mathcal{A} \cos \theta - \mathcal{B} \sin \theta) + \frac{\rho^2}{2\mathcal{R}^3} (\mathcal{B} \cos \theta - \mathcal{A} \sin \theta)^2 \, d\theta \right. \\ &\quad \left. + \int_0^\pi \frac{\rho}{\mathcal{R}} (\mathcal{A} \cos \theta - \mathcal{B} \sin \theta) + \frac{\rho^2}{2\mathcal{R}^3} (\mathcal{B} \cos \theta + \mathcal{A} \sin \theta)^2 \, d\theta + 2\pi\mathcal{R} \right] \, d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \left(2\pi\mathcal{R} + \int_0^\pi -2\frac{\rho\mathcal{B}}{\mathcal{R}} \sin \theta + \frac{\rho^2}{2\mathcal{R}^3} (\mathcal{A}^2 \sin^2 \theta + \mathcal{B}^2 \cos^2 \theta) \, d\theta \right) \, d\rho \\ &= \int_0^\infty \Delta G_\sigma(\rho) \left(-2\pi\mathcal{R}\rho + 4\frac{\mathcal{B}}{\mathcal{R}}\rho^2 - \frac{\pi}{2\mathcal{R}}\rho^3 \right) \, d\rho \end{aligned}$$

Using definite integrals $\int_0^\infty \rho \Delta G_\sigma(\rho) d\rho = 0$, $\int_0^\infty \rho^2 \Delta G_\sigma(\rho) d\rho = \frac{1}{2\sigma\sqrt{2\pi}}$ and $\int_0^\infty \rho^3 \Delta G_\sigma(\rho) d\rho = \frac{2}{\pi}$, we finally obtain

$$\widetilde{\text{rdg}}_{\text{ligature}}(s) = \frac{1}{\sqrt{\mathcal{A}^2 + \mathcal{B}^2}} \left(\mathcal{B} \frac{\sqrt{2\pi}}{\pi\sigma} - 1 \right)$$

which proves Proposition 20. In order to prove that the ridgeness of a ligature point is asymptotically equivalent to its second-order approximation

$$\text{rdg}_{\text{ligature}}(s) \sim \widetilde{\text{rdg}}_{\text{ligature}}(s) \quad (\text{as } \mathcal{A} \rightarrow +\infty),$$

we show that

$$\lim_{\mathcal{A} \rightarrow +\infty} \frac{\pi\sigma\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \text{rdg}_{\text{ligature}}(s) = 1.$$

Let us consider Eq. (C.14). For convenience, we introduce

$$f(\mathcal{A}, \rho) = \left(\sqrt{\mathcal{A}^2 + \mathcal{B}^2} + \rho \right) \left[2E(k) - E\left(\frac{\beta}{2}, k\right) - E\left(\frac{\pi}{2} - \frac{\beta}{2}, k\right) \right]$$

with $k = \frac{2\sqrt{\rho\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}}{\rho + \sqrt{\mathcal{A}^2 + \mathcal{B}^2}}$ and $\beta = \tan^{-1}\left(\frac{\mathcal{B}}{\mathcal{A}}\right)$, so that

$$\text{rdg}_{\text{ligature}}(s) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) f(\mathcal{A}, \rho) d\rho.$$

We can write

$$\frac{\pi\sigma\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \text{rdg}_{\text{ligature}}(s) = -\frac{4\pi\sigma}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \int_0^\infty \rho \Delta G_\sigma(\rho) f(\mathcal{A}, \rho) \sqrt{\mathcal{A}^2 + \mathcal{B}^2} d\rho$$

Since

$$\int_0^\infty \rho \Delta G_\sigma(\rho) d\rho = 0,$$

it is also true that

$$\int_0^\infty \frac{\pi}{2} (\mathcal{A}^2 + \mathcal{B}^2) \rho \Delta G_\sigma(\rho) d\rho = 0.$$

Hence,

$$\frac{\pi\sigma\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \text{rdg}_{\text{ligature}}(s) = -\frac{4\pi\sigma}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \int_0^\infty \rho \Delta G_\sigma(\rho) \sqrt{\mathcal{A}^2 + \mathcal{B}^2} \left(f(\mathcal{A}, \rho) - \frac{\pi}{2} \sqrt{\mathcal{A}^2 + \mathcal{B}^2} \right) d\rho$$

Thanks to the power series expansions in Eqs. (C.6) and (C.7), we can derive from Eq. (C.14),

$$h(\mathcal{A}, \rho) = \sqrt{\mathcal{A}^2 + \mathcal{B}^2} \left(f(\mathcal{A}, \rho) - \frac{\pi}{2} \sqrt{\mathcal{A}^2 + \mathcal{B}^2} \right) = \frac{(\mathcal{A}^2 + \mathcal{B}^2)^{\frac{3}{2}} (\pi\rho^2 - 8\mathcal{B}\rho)}{8 \left(\sqrt{\mathcal{A}^2 + \mathcal{B}^2} + \rho \right)^3} + O\left(\frac{1}{\mathcal{A}}\right)$$

and thus,

$$\lim_{\mathcal{A} \rightarrow +\infty} h(\mathcal{A}, \rho) = \frac{\pi\rho^2 - 8\mathcal{B}\rho}{8}.$$

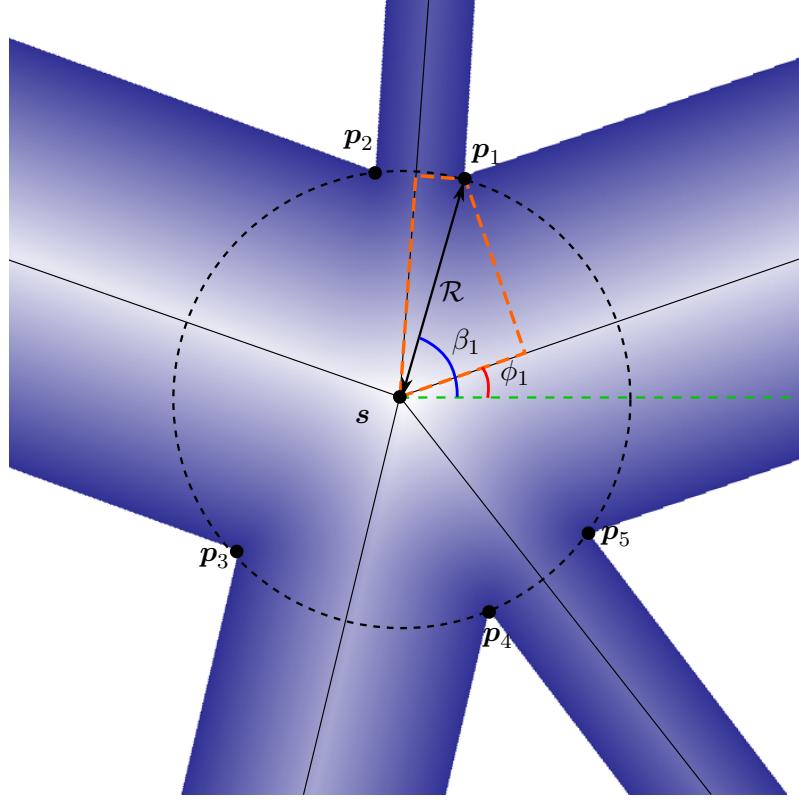


Figure C.2: Junction point with equidistant corners

For every $\rho \in [0, +\infty)$, $\rho \Delta G_\sigma(\rho) h(\mathcal{A}, \rho)$ is dominated by $(\max(1, \rho^3) + \mathcal{B} \max(1, \rho^2)) |\Delta G_\sigma(\rho)|$, which is L^1 -integrable. Hence, by the DCT,

$$\begin{aligned} \lim_{\mathcal{A} \rightarrow +\infty} -\frac{4\pi\sigma}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \int_0^\infty \rho \Delta G_\sigma(\rho) h(\mathcal{A}, \rho) d\rho &= -\frac{4\pi\sigma}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \int_0^\infty \rho \Delta G_\sigma(\rho) \left(\lim_{\mathcal{A} \rightarrow +\infty} h(\mathcal{A}, \rho) \right) d\rho \\ &= -\frac{4\pi\sigma}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \int_0^\infty \left(\frac{\pi\rho^2 - 8\mathcal{B}\rho}{8} \right) \Delta G_\sigma(\rho) d\rho \\ &= 1 \end{aligned}$$

Hence,

$$\lim_{\mathcal{A} \rightarrow +\infty} \frac{\pi\sigma\sqrt{\mathcal{A}^2 + \mathcal{B}^2}}{\mathcal{B}\sqrt{2\pi} - \pi\sigma} \text{rdg}_{\text{ligature}}(s) = 1$$

which proves Proposition 21.

C.1.15 Junction point - AOF

Assume that $x_0 = \mathbf{0}$. There are n corners p_i , equidistant from junction point s . In the neighborhood of x_0 , the distance is:

$$D(\mathbf{x}) = \min_{i=1 \dots n} D_i(\mathbf{x})$$

with $D_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_i\|$. Switching to polar coordinates, corners are defined as $\mathbf{p}_i = s + [\mathcal{R} \cos \beta_i, \mathcal{R} \sin \beta_i]^\top$, where β_i is the absolute angle formed with the x -axis. We denote by α_i the relative angle formed by x_0 and two successive corners \mathbf{p}_i and \mathbf{p}_{i+1} : $\alpha_i = \beta_{i+1} - \beta_i$. The distance is defined piecewise over radial intervals around s . For a point \mathbf{x} in the neighborhood of s s.t.

$$\mathbf{x} = s + [\rho \cos \theta, \rho \sin \theta]^\top,$$

assuming that $\rho \leq \mathcal{R}$ and $\theta \in [\phi_i, \phi_{i+1}]$, with $\phi_i = \frac{\beta_{i-1} + \beta_i}{2}$, \mathbf{x} is in the radial range of points closest to \mathbf{p}_i . This is exemplified in Fig. 6.10, where the area outlined with dotted orange lines is the set of points closest to \mathbf{x}_1 than any other \mathbf{x}_i . It is spanned by angle range $[\phi_1, \phi_2]$. Using Eq. (6.41), the gradient of the distance on the circle of radius r is

$$\begin{aligned}\nabla D_i \left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) &= \left\| \begin{bmatrix} r \cos \theta - \mathcal{R} \cos \beta_i \\ r \sin \theta - \mathcal{R} \sin \beta_i \end{bmatrix} \right\|^{-1} \begin{bmatrix} r \cos \theta - \mathcal{R} \cos \beta_i \\ r \sin \theta - \mathcal{R} \sin \beta_i \end{bmatrix} \\ &= \frac{1}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta_i)}} \begin{bmatrix} r \cos \theta - \mathcal{R} \cos \beta_i \\ r \sin \theta - \mathcal{R} \sin \beta_i \end{bmatrix}\end{aligned}$$

Plugging it into Eq (6.4), it comes

$$\begin{aligned}\text{aof}_{\text{junction}}(\mathbf{s}) &= \frac{1}{2\pi} \int_0^{2\pi} \nabla D \left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \\ &= \frac{1}{2\pi} \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \nabla D_i \left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \nabla D_{i+1} \left(\mathbf{s} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right] \\ &= \frac{1}{2\pi} \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \frac{r - \mathcal{R} \cos(\theta - \beta_i)}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta_i)}} d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \frac{r - \mathcal{R} \cos(\theta - \beta_{i+1})}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos(\theta - \beta_{i+1})}} d\theta \right]\end{aligned}$$

By translation, it comes

$$\begin{aligned}\text{aof}_{\text{junction}}(\mathbf{s}) &= \frac{1}{2\pi} \sum_{i=1}^n \left[\int_0^{\frac{\alpha_i}{2}} \frac{r - \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta + \int_{-\frac{\alpha_i}{2}}^0 \frac{r - \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta \right] \\ &= \frac{1}{\pi} \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} \frac{r - \mathcal{R} \cos \theta}{\sqrt{r^2 + \mathcal{R}^2 - 2r\mathcal{R} \cos \theta}} d\theta\end{aligned}$$

which proves Proposition 22. Using identity $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$ and a change of variable,

$$\text{aof}_{\text{junction}}(\mathbf{s}) = \frac{2}{\pi} \sum_{i=1}^n \int_0^{\frac{\alpha_i}{4}} \frac{r - \mathcal{R} + 2\mathcal{R} \sin^2 \theta}{\sqrt{(r - \mathcal{R})^2 + 4r\mathcal{R} \sin^2 \theta}} d\theta$$

Using Eqs. (6.21) and (C.3), it can be expressed using incomplete elliptic integrals with purely imaginary modulus:

$$\text{aof}_{\text{junction}}(\mathbf{s}) = \frac{1}{\pi r} \sum_{i=1}^n \frac{r^2 - \mathcal{R}^2}{|r - \mathcal{R}|} F \left(\frac{\alpha_i}{4}, ik_1 \right) + |r - \mathcal{R}| E \left(\frac{\alpha_i}{4}, ik_1 \right)$$

with $k_1 = \frac{2\sqrt{r\mathcal{R}}}{|\mathcal{R} - r|}$. We convert them to elliptic integrals with real modulus using rule (C.4):

$$\text{aof}_{\text{junction}}(\mathbf{s}) = \frac{1}{\pi r} \sum_{i=1}^n \left[(\mathcal{R} + r) \left(E(k) - E \left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k \right) \right) - (\mathcal{R} - r) \left(K(k) - F \left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k \right) \right) \right] \quad (\text{C.15})$$

with $k = \frac{2\sqrt{r\mathcal{R}}}{\mathcal{R} + r}$, which proves Proposition 23.

C.1.16 Junction point - AOF - Asymptotical behavior

We consider the second-order Taylor approximation of D_i , as defined in Eq. (6.41), in the polar coordinate system centered at s . We have

$$\begin{aligned}\nabla D_i(s) &= [-\cos \beta_i, -\sin \beta_i]^\top \\ \nabla^2 D_i(s) &= \frac{1}{\mathcal{R}} \begin{bmatrix} \sin^2 \beta_i & -\cos \beta_i \sin \beta_i \\ -\cos \beta_i \sin \beta_i & \cos^2 \beta_i \end{bmatrix}\end{aligned}$$

So, the approximated distance on the arc of radius r centered at s , for $\theta \in [\phi_i, \phi_{i+1}]$ is

$$\begin{aligned}\tilde{D}_i\left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) &= \mathcal{R} + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla D_i(s) + \frac{1}{2} \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}^\top \nabla^2 D_i(s) \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= \mathcal{R} - r \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \cos \beta_i \\ \sin \beta_i \end{bmatrix} \\ &\quad + \frac{r^2}{2\mathcal{R}} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top \begin{bmatrix} \sin^2 \beta_i & -\cos \beta_i \sin \beta_i \\ -\cos \beta_i \sin \beta_i & \cos^2 \beta_i \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \\ &= \mathcal{R} - r \cos(\theta - \beta_i) + \frac{r^2}{2\mathcal{R}} \sin^2(\theta - \beta_i)\end{aligned}\tag{C.16}$$

and its gradient is

$$\begin{aligned}\nabla \tilde{D}_i\left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) &= \nabla D_i(s) + \nabla^2 D_i(s) \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= - \begin{bmatrix} \cos \beta_i \\ \sin \beta_i \end{bmatrix} + \frac{r}{\mathcal{R}} \begin{bmatrix} \sin^2 \beta_i & -\cos \beta_i \sin \beta_i \\ -\cos \beta_i \sin \beta_i & \cos^2 \beta_i \end{bmatrix} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}\end{aligned}$$

Plugging it into Eq. (6.26), we obtain

$$\begin{aligned}\widetilde{\text{aof}}_{\text{junction}}(s) &= \frac{1}{2\pi} \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \nabla \tilde{D}_i\left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \nabla \tilde{D}_{i+1}\left(s + \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}\right) \cdot \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} d\theta \right] \\ &= \frac{1}{2\pi} \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} -\cos(\theta - \beta_i) + \frac{r}{\mathcal{R}} \sin^2(\theta - \beta_i) d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} -\cos(\theta - \beta_{i+1}) + \frac{r}{\mathcal{R}} \sin^2(\theta - \beta_{i+1}) d\theta \right]\end{aligned}$$

By translation,

$$\begin{aligned}\widetilde{\text{aof}}_{\text{junction}}(s) &= \frac{1}{\pi} \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} -\cos \theta + \frac{r}{\mathcal{R}} \sin^2 \theta d\theta \\ &= \frac{1}{\pi} \sum_{i=1}^n -\sin\left(\frac{\alpha_i}{2}\right) + \frac{r}{4\mathcal{R}} (\alpha_i - \sin \alpha_i) \\ &= -\frac{1}{\pi} s_2 + \frac{r}{\mathcal{R}} \left(1 - \frac{1}{2\pi} s_1\right)\end{aligned}$$

where s_1 and s_2 are defined in Eq. (6.45), which proves Proposition 24. In order to prove that the AOF at a junction point is asymptotically equivalent to its second-order approximation

$$\text{aof}_{\text{junction}}(s) \sim \widetilde{\text{aof}}_{\text{junction}}(s) \text{ (as } \mathcal{R} \rightarrow +\infty\text{)},$$

we show that

$$\lim_{\mathcal{R} \rightarrow +\infty} \frac{2\pi\mathcal{R}}{-2s_2\mathcal{R} + 2\pi r - rs_1} \text{aof}_{\text{junction}}(\mathbf{s}) = 1.$$

Using series expansions in Eqs. (C.6) and (C.7), we derive the elliptic terms of Eq. (C.15)

$$\begin{aligned} K(k) - F\left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k\right) &= \frac{\alpha_i}{4} + k^2 \left(\frac{1}{8} \sin\left(\frac{\alpha_i}{2}\right) + \frac{\alpha_i}{16} \right) + k^4 \left(\frac{3 \sin \alpha_i}{256} + \frac{3}{32} \sin\left(\frac{\alpha_i}{2}\right) + \frac{9\alpha_i}{256} \right) + O(k^6) \\ E(k) - E\left(\frac{\pi}{2} - \frac{\alpha_i}{4}, k\right) &= \frac{\alpha_i}{4} - k^2 \left(\frac{1}{8} \sin\left(\frac{\alpha_i}{2}\right) + \frac{\alpha_i}{16} \right) - k^4 \left(\frac{\sin \alpha_i}{256} + \frac{1}{32} \sin\left(\frac{\alpha_i}{2}\right) + \frac{3\alpha_i}{256} \right) + O(k^6) \end{aligned}$$

which yields

$$\begin{aligned} \text{aof}_{\text{junction}}(\mathbf{s}) &= \sum_{i=1}^n \left\{ \frac{\alpha_i}{2\pi} - \frac{1}{\pi(\mathcal{R}+r)^4} \left[\mathcal{R}^4 \left(\frac{\alpha_i}{2} + \sin\left(\frac{\alpha_i}{2}\right) \right) + \right. \right. \\ &\quad \left. \left. + \mathcal{R}^3 r \left(\frac{7\alpha_i}{4} + 4 \sin\left(\frac{\alpha_i}{2}\right) + \frac{\sin \alpha_i}{4} \right) + \mathcal{R}^2 r^2 \left(\frac{\alpha_i}{8} - \frac{\sin \alpha_i}{8} \right) \right] \right\} + O\left(\frac{1}{\mathcal{R}^2}\right) \\ &= 1 - \frac{1}{\pi(\mathcal{R}+r)^4} \left[\mathcal{R}^4 (\pi + s_2) + \mathcal{R}^3 r \left(4s_2 + \frac{7\pi}{2} + \frac{s_1}{4} \right) + \mathcal{R}^2 r^2 \left(\frac{\pi}{4} - \frac{s_1}{8} \right) \right] + O\left(\frac{1}{\mathcal{R}^2}\right) \\ &= \frac{-8s_2\mathcal{R}^4 + (4\pi - 2s_1 - 32s_2)\mathcal{R}^3 r + (46\pi - s_1)\mathcal{R}^2 r^2 + 32\pi\mathcal{R}r^3 + 8\pi r^4}{8\pi(\mathcal{R}+r)^4} + O\left(\frac{1}{\mathcal{R}^2}\right) \end{aligned}$$

Thus,

$$\lim_{\mathcal{R} \rightarrow +\infty} \frac{2\pi\mathcal{R}}{-2s_2\mathcal{R} + 2\pi r - rs_1} \text{aof}_{\text{junction}}(\mathbf{s}) = \lim_{\mathcal{R} \rightarrow +\infty} \frac{\mathcal{R}^4}{(\mathcal{R}+r)^4} + O\left(\frac{1}{\mathcal{R}}\right) = 1$$

which proves Proposition 25.

C.1.17 Junction point - Ridgeness

Using Eqs (6.6) and (6.41), the ridgeness at a junction point can be expressed in polar coordinates as

$$\begin{aligned} \text{rdg}_{\text{junction}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} D_i \left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} D_{i+1} \left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) d\theta \right] d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos(\theta - \beta_i)} d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos(\theta - \beta_{i+1})} d\theta \right] d\rho \end{aligned}$$

By translation,

$$\begin{aligned} \text{rdg}_{\text{junction}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_0^{\frac{\beta_{i+1}-\beta_i}{2}} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos \theta} d\theta \right. \\ &\quad \left. + \int_{-\frac{\beta_{i+1}-\beta_i}{2}}^0 \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos \theta} d\theta \right] d\rho \\ &= -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} \sqrt{\rho^2 + \mathcal{R}^2 - 2\rho\mathcal{R} \cos \theta} d\theta d\rho \end{aligned}$$

which proves Proposition 26. Using transformation $\cos \theta = 1 - 2 \sin^2 \frac{\theta}{2}$, it comes

$$\text{rdg}_{\text{junction}}(\mathbf{s}) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \int_0^{\frac{\alpha_i}{4}} \sqrt{(\mathcal{R} - \rho)^2 + 4\rho\mathcal{R} \sin^2 \theta} \, d\theta \, d\rho,$$

which can be expressed with incomplete elliptic integrals of the second kind with purely imaginary modulus

$$\text{rdg}_{\text{junction}}(\mathbf{s}) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n |\mathcal{R} - \rho| E \left(\frac{\alpha_i}{4}, i \frac{2\sqrt{\rho\mathcal{R}}}{|\mathcal{R} - \rho|} \right) \, d\rho.$$

Using transformations in Eq. (C.4), we obtain:

$$\text{rdg}_{\text{junction}}(\mathbf{s}) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n (\mathcal{R} + \rho) \left(E \left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho} \right) - E \left(\frac{\pi}{2} - \frac{\alpha_i}{4}, \frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho} \right) \right) \, d\rho, \quad (\text{C.17})$$

which proves Proposition 27.

C.1.18 Junction point - Ridgeness - Asymptotical behavior

Using Eqs (6.30) and (C.16), the approximate ridgeness at a junction point is

$$\begin{aligned} \widetilde{\text{rdg}}_{\text{junction}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \tilde{D}_i \left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) \, d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \tilde{D}_{i+1} \left(\mathbf{s} + \begin{bmatrix} \rho \cos \theta \\ \rho \sin \theta \end{bmatrix} \right) \, d\theta \right] \, d\rho \\ &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_{\beta_i}^{\frac{\beta_i+\beta_{i+1}}{2}} \mathcal{R} - \rho \cos(\theta - \beta_i) + \frac{\rho^2}{2\mathcal{R}} \sin^2(\theta - \beta_i) \, d\theta \right. \\ &\quad \left. + \int_{\frac{\beta_i+\beta_{i+1}}{2}}^{\beta_{i+1}} \mathcal{R} - \rho \cos(\theta - \beta_{i+1}) + \frac{\rho^2}{2\mathcal{R}} \sin^2(\theta - \beta_{i+1}) \, d\theta \right] \, d\rho \end{aligned}$$

By translation, we get

$$\begin{aligned} \widetilde{\text{rdg}}_{\text{junction}}(\mathbf{s}) &= - \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \left[\int_0^{\frac{\beta_{i+1}-\beta_i}{2}} \mathcal{R} - \rho \cos \theta + \frac{\rho^2}{2\mathcal{R}} \sin^2 \theta \, d\theta \right. \\ &\quad \left. + \int_{-\frac{\beta_{i+1}-\beta_i}{2}}^0 \mathcal{R} - \rho \cos \theta + \frac{\rho^2}{2\mathcal{R}} \sin^2 \theta \, d\theta \right] \, d\rho \\ &= -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \int_0^{\frac{\alpha_i}{2}} \mathcal{R} - \rho \cos \theta + \frac{\rho^2}{2\mathcal{R}} \sin^2 \theta \, d\theta \, d\rho \\ &= -2 \int_0^\infty \rho \Delta G_\sigma(\rho) \sum_{i=1}^n \mathcal{R} \frac{\alpha_i}{2} - \rho \sin \left(\frac{\alpha_i}{2} \right) + \frac{\rho^2}{2\mathcal{R}} \left(\frac{\alpha_i}{4} - \frac{1}{4} \sin \alpha_i \right) \, d\rho. \end{aligned}$$

Using Eq. (6.45), this simplifies to

$$\begin{aligned} \widetilde{\text{rdg}}_{\text{junction}}(\mathbf{s}) &= \int_0^\infty \rho \Delta G_\sigma(\rho) \left(-2\pi\mathcal{R} + 2\rho s_2 - \frac{\rho^2}{4\mathcal{R}} (2\pi - s_1) \right) \, d\rho \\ &= \frac{1}{\sigma\sqrt{2\pi}} s_2 - \frac{1}{\mathcal{R}} \left(1 - \frac{1}{2\pi} s_1 \right), \end{aligned}$$

which proves Proposition 28. In order to prove that the ridgeness of a junction point is asymptotically equivalent to its second-order approximation

$$\text{rdg}_{\text{junction}}(\mathbf{s}) \sim \widetilde{\text{rdg}}_{\text{junction}}(\mathbf{s}) \quad (\text{as } \mathcal{R} \rightarrow +\infty),$$

we show that

$$\lim_{\mathcal{R} \rightarrow +\infty} \frac{2\pi\sigma\mathcal{R}}{s_2\mathcal{R}\sqrt{2\pi} - 2\pi\sigma + \sigma s_1} \text{rdg}_{\text{junction}}(\mathbf{s}) = 1.$$

Let us consider Eq. (C.17). For convenience, we introduce

$$f(\mathcal{R}, \rho) = (\mathcal{R} + \rho) \sum_{i=1}^n \left\{ E\left(\frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}\right) - E\left(\frac{\pi}{2} - \frac{\alpha_i}{4}, \frac{2\sqrt{\rho\mathcal{R}}}{\mathcal{R} + \rho}\right) \right\},$$

so that

$$\text{rdg}_{\text{junction}}(\mathbf{s}) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) f(\mathcal{R}, \rho) d\rho.$$

Since

$$\int_0^\infty \rho \Delta G_\sigma(\rho) d\rho = 0,$$

it is also true that

$$\int_0^\infty \frac{\pi}{2} \mathcal{R} \rho \Delta G_\sigma(\rho) d\rho = 0.$$

Hence,

$$\text{rdg}_{\text{junction}}(\mathbf{s}) = -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left(f(\mathcal{R}, \rho) - \frac{\pi}{2} \mathcal{R} \right) d\rho$$

Let

$$h(\mathcal{R}, \rho) = f(\mathcal{R}, \rho) - \frac{\pi}{2} \mathcal{R}.$$

Thanks to the power series expansions in Eqs. (C.6) and (C.7), it comes

$$\begin{aligned} h(\mathcal{R}, \rho) &= \sum_{i=1}^n \left\{ \frac{1}{(\mathcal{R} + \rho)^3} \left[\left(\frac{1}{2} \sin\left(\frac{\alpha_i}{2}\right) + \frac{\alpha_i}{4} \right) \mathcal{R}^4 + \left(\sin\left(\frac{\alpha_i}{2}\right) + \frac{\alpha_i}{2} \right) \mathcal{R}^3 \rho \right. \right. \\ &\quad \left. \left. + \left(\frac{\alpha_i}{16} + \frac{\sin \alpha_i}{16} \right) \mathcal{R}^2 \rho^2 \right] - \frac{\mathcal{R}}{2} \sin\left(\frac{\alpha_i}{2}\right) + \frac{\alpha_i \rho}{4} \right\} - \frac{\pi \mathcal{R}}{2} + O\left(\frac{1}{\mathcal{R}^2}\right) \\ &= \frac{1}{(\mathcal{R} + \rho)^3} \left[\frac{\pi \mathcal{R}^4}{2} + \left(\frac{3\pi}{2} - \frac{s_2}{2} \right) \mathcal{R}^3 \rho \right. \\ &\quad \left. + \left(\frac{5\pi}{8} - \frac{s_1}{16} - \frac{s_2}{2} \right) \mathcal{R}^2 \rho^2 + \left(\frac{\pi}{2} - \frac{s_2}{2} \right) \mathcal{R} \rho^3 + \frac{\pi \rho^4}{2} \right] - \frac{\pi \mathcal{R}}{2} + O\left(\frac{1}{\mathcal{R}^2}\right) \\ &= -\frac{\rho(8s_2\mathcal{R}^3 + (14\pi + s_1 + 8s_2)\mathcal{R}^2\rho + 8s_2\mathcal{R}\rho^2 - 8\pi\rho^3)}{16(\mathcal{R} + \rho)^3} + O\left(\frac{1}{\mathcal{R}^2}\right) \end{aligned}$$

and thus,

$$\lim_{\mathcal{R} \rightarrow +\infty} h(\mathcal{R}, \rho) = -\frac{s_2}{2} \rho.$$

For every $\rho \in [0, +\infty)$, $\rho \Delta G_\sigma(\rho) h(\mathcal{R}, \rho)$ is dominated by $s_2 \max(1, \rho^3) |\Delta G_\sigma(\rho)|$, which is L^1 -integrable. Hence, by the DCT,

$$\begin{aligned} \lim_{\mathcal{R} \rightarrow +\infty} \text{rdg}_{\text{junction}}(\mathbf{s}) &= \lim_{\mathcal{R} \rightarrow +\infty} -4 \int_0^\infty \rho \Delta G_\sigma(\rho) h(\mathcal{R}, \rho) d\rho \\ &= -4 \int_0^\infty \rho \Delta G_\sigma(\rho) \left(\lim_{\mathcal{R} \rightarrow +\infty} h(\mathcal{R}, \rho) \right) d\rho \\ &= 2s_2 \int_0^\infty \rho^2 \Delta G_\sigma(\rho) d\rho \\ &= \frac{s_2 \sqrt{2\pi}}{2\pi\sigma} \end{aligned}$$

Hence,

$$\begin{aligned} \lim_{\mathcal{R} \rightarrow +\infty} \frac{2\pi\sigma\mathcal{R}}{s_2\mathcal{R}\sqrt{2\pi} - 2\pi\sigma + \sigma s_1} \text{rdg}_{\text{junction}}(s) &= \lim_{\mathcal{R} \rightarrow +\infty} \frac{2\pi\sigma\mathcal{R}}{s_2\mathcal{R}\sqrt{2\pi} - 2\pi\sigma + \sigma s_1} \cdot \lim_{\mathcal{R} \rightarrow +\infty} \text{rdg}_{\text{junction}}(s) \\ &= \frac{2\pi\sigma}{s_2\sqrt{2\pi}} \cdot \frac{s_2\sqrt{2\pi}}{2\pi\sigma} = 1 \end{aligned}$$

which proves Proposition 29.

C.2 Deformable skeleton: calculus of variations with variable endpoints

This section gives the derivation used to obtain evolution equation (6.58). Consider the general energy functional as an integral along the open curve \mathbf{c} ,

$$E[\mathbf{c}] = \int_0^1 h(\mathbf{c}(u), \mathbf{c}'(u)) du. \quad (\text{C.18})$$

where $h : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ is continuously differentiable with respect to each of its two parameters. A necessary condition for $E[\mathbf{c}]$ being a local extremum is that its first variation vanishes

$$\left. \frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \right|_{\epsilon=0} = 0. \quad (\text{C.19})$$

for any small and smooth variation $\boldsymbol{\eta} : [0, 1] \rightarrow \mathbb{R}^2$. Using the chain rule, we have

$$\left. \frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \right|_{\epsilon=0} = \int_0^1 \boldsymbol{\eta} \cdot \frac{\partial h}{\partial \mathbf{c}} + \boldsymbol{\eta}' \cdot \frac{\partial h}{\partial \mathbf{c}'} du.$$

The mathematical derivation in Section 4.3 assumed that \mathbf{c} was closed or had fixed endpoints ($\boldsymbol{\eta}(0) = \boldsymbol{\eta}(1) = 0$). In the present case, the curve is open, and has free endpoints. No boundary condition are set on $\mathbf{c}(0)$ and $\mathbf{c}(1)$. Thus, variation $\boldsymbol{\eta}$ is not assumed to vanish at $u = 0$ and $u = 1$. Integration by parts on the second term yields

$$\left. \frac{dE[\mathbf{c} + \epsilon\boldsymbol{\eta}]}{d\epsilon} \right|_{\epsilon=0} = \left[\boldsymbol{\eta} \cdot \frac{\partial h}{\partial \mathbf{c}'} \right]_{u=0}^{u=1} + \int_0^1 \boldsymbol{\eta} \cdot \left(\frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} \right) du.$$

The problem of setting the necessary conditions for the first variation to vanish, in the case of free endpoints, is studied in [van Brunt, 2004, Chapter 7]. It follows that the three conditions should be verified

$$\left\{ \begin{array}{l} \left. \frac{\partial h}{\partial \mathbf{c}'} \right|_{u=0} = 0 \\ \left. \frac{\partial h}{\partial \mathbf{c}'} \right|_{u=1} = 0 \\ \frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} = 0 \quad \forall u \in [0, 1] \end{array} \right. \quad (\text{C.20})$$

We can build an evolution equation minimizing energy (C.18) by making the curve time-dependent and, as in Eq. (4.10), set the time-derivative equal to the opposite of the functional derivatives involved in (C.20). Notice the extra term at endpoints $\mathbf{c}(0, t)$ and $\mathbf{c}(1, t)$,

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{c}(u, t)}{\partial t} = -\frac{\partial h}{\partial \mathbf{c}} + \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} \quad \forall u \in (0, 1) \\ \frac{\partial \mathbf{c}(0, t)}{\partial t} = \left(-\frac{\partial h}{\partial \mathbf{c}} + \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} \right) \Big|_{u=0} - \left. \frac{\partial h}{\partial \mathbf{c}'} \right|_{u=0} \\ \frac{\partial \mathbf{c}(1, t)}{\partial t} = \left(-\frac{\partial h}{\partial \mathbf{c}} + \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} \right) \Big|_{u=1} - \left. \frac{\partial h}{\partial \mathbf{c}'} \right|_{u=1} \\ \mathbf{c}(u, 0) = \mathbf{c}_0(u) \quad \forall u \in [0, 1]. \end{array} \right. \quad (\text{C.21})$$

We now address the minimization of line integral,

$$E[\mathbf{c}] = \int_0^1 g(\mathbf{c}) \|\mathbf{c}'\| du, \quad (\text{C.22})$$

with a C^1 cost function $g : \mathcal{D} \rightarrow \mathbb{R}^+$. This amounts to consider general energy (C.18) with $h(\mathbf{c}, \mathbf{c}') = g(\mathbf{c}) \|\mathbf{c}'\|$. The partial derivatives involved in the necessary conditions (C.20) are expanded to

$$\frac{\partial g}{\partial \mathbf{c}'} = g(\mathbf{c}) \frac{\mathbf{c}'}{\|\mathbf{c}'\|} = g(\mathbf{c}) \mathbf{t}, \quad (\text{C.23})$$

and, as shown in [Caselles et al., 1997],

$$\frac{\partial h}{\partial \mathbf{c}} - \frac{d}{du} \left\{ \frac{\partial h}{\partial \mathbf{c}'} \right\} = (\nabla g(\mathbf{c}) \cdot \mathbf{n} - g(\mathbf{c}) \kappa) \|\mathbf{c}'\| \mathbf{n}. \quad (\text{C.24})$$

Plugging Eqs (C.23) and (C.24) into system (C.21), we obtain evolution equation

$$\begin{cases} \frac{\partial \mathbf{c}(u, t)}{\partial t} &= (g(\mathbf{c}(u, t)) \kappa(u, t) - \nabla g(\mathbf{c}(u, t)) \cdot \mathbf{n}(u, t)) \|\mathbf{c}'(u, t)\| \mathbf{n}(u, t) & \forall u \in (0, 1) \\ \frac{\partial \mathbf{c}(0, t)}{\partial t} &= -(\nabla g(\mathbf{c}(0, t)) \cdot \mathbf{n}(0, t)) \|\mathbf{c}'(0, t)\| \mathbf{n} + g(\mathbf{c}(0, t)) \mathbf{t}(0, t) \\ \frac{\partial \mathbf{c}(1, t)}{\partial t} &= -(\nabla g(\mathbf{c}(1, t)) \cdot \mathbf{n}(1, t)) \|\mathbf{c}'(1, t)\| \mathbf{n} - g(\mathbf{c}(1, t)) \mathbf{t}(1, t) \\ \mathbf{c}(u, 0) &= \mathbf{c}_0(u) \end{cases} \quad \forall u \in [0, 1]. \quad (\text{C.25})$$

where the curvature-dependent term vanishes at endpoints, since $\kappa(0) = \kappa(1) = 0$.

Bibliography

- [Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, F., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.
- [Adams, 2004] Adams, C. (2004). *The Knot Book: An elementary introduction to the mathematical theory of knots*. American Mathematical Society.
- [Aeschliman et al., 2010] Aeschliman, C., Park, J., and Kak, A. (2010). A probabilistic framework for joint segmentation and tracking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1371–1378, San Francisco, USA.
- [Alzer and Qiu, 2004] Alzer, H. and Qiu, S.-L. (2004). Monotonicity theorems and inequalities for the complete elliptic integrals. *Journal of Computational and Applied Mathematics*, 172(2):289–312.
- [Amenta et al., 2001] Amenta, N., Sunghee, C., and Kolluri, R. (2001). The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153.
- [Anderson et al., 1992] Anderson, G., Vamanamurthy, M., and Vuorinen, M. (1992). Functional inequalities for hypergeometric functions and complete elliptic integrals. *SIAM Journal on Mathematical Analysis*, 31:512–524.
- [Andreopoulos and Tsotsos, 2013] Andreopoulos, A. and Tsotsos, J. (2013). 50 years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891.
- [Appia and Yezzi, 2010] Appia, V. and Yezzi, A. (2010). Fully isotropic fast marching methods on cartesian grids. In *European Conference on Computer Vision*, volume 6311 of *LNCS*, pages 73–85. Springer-Verlag.
- [Arcelli and Sanniti di Baja, 1985] Arcelli, C. and Sanniti di Baja, G. (1985). A width-independent fast thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):463–474.
- [Arcelli and Sanniti di Baja, 1992] Arcelli, C. and Sanniti di Baja, G. (1992). Ridge points in Euclidean distance maps. *Pattern Recognition Letters*, 13(4):237–243.
- [Arnold, 1994] Arnold, V. (1994). Plane curves, their invariants, perestroikas and classification. *Advances in Soviet Mathematics: Singularities and Bifurcations*, 21:33–91.
- [Attali et al., 1995] Attali, D., Sanniti di Baja, G., and Thiel, E. (1995). Pruning discrete and semicontinuous skeletons. In *Image Analysis and Processing*, pages 488–493.
- [Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.
- [Bai and Latecki, 2008] Bai, X. and Latecki, L. (2008). Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1282–1292.

- [Bai et al., 2007] Bai, X., Latecki, L., and Liu, W. (2007). Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):449–462.
- [Baker et al., 2011] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.
- [Barron et al., 1994] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77.
- [Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.
- [Benmansour and Cohen, 2009] Benmansour, F. and Cohen, L. (2009). Fast object segmentation by growing minimal paths from a single point on 2D or 3D images. *Journal of Mathematical Imaging and Vision*, 33(2):209–221.
- [Bentley and Ottmann, 1979] Bentley, J. and Ottmann, T. (1979). Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, C-28(9):643–647.
- [Bertalmio et al., 2000] Bertalmio, M., Sapiro, G., and Randall, G. (2000). Morphing active contours. 22(7):733–737.
- [Bertrand and Malandain, 1994] Bertrand, G. and Malandain, G. (1994). A new characterization of three-dimensional simple points. *Pattern Recognition Letters*, 15(2):169–175.
- [Bishop, 2006] Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.
- [Blum, 1967] Blum, H. (1967). *A transformation for extracting new descriptors of shape*, pages 362–380. MIT Press.
- [Blum and Nagel, 1978] Blum, H. and Nagel, R. (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, 10:167–180.
- [Borgefors et al., 2001] Borgefors, G., Ramella, G., and Sanniti di Baja, G. (2001). Hierarchical decomposition of multiscale skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1296–1312.
- [Bougleux et al., 2008] Bougleux, S., Peyré, G., and Cohen, L. (2008). Anisotropic geodesics for perceptual grouping and domain meshing. *European Conference on Computer Vision (ECCV)*, 2:129–142.
- [Bouix et al., 2005] Bouix, S., Siddiqi, K., and Tannenbaum, A. (2005). Flux driven automatic centerline extraction. *Medical Image Analysis*, 9(3):209–221.
- [Bouwmans, 2014] Bouwmans, T. (2014). Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31–66.
- [Bouwmans et al., 2008] Bouwmans, T., El Baf, F., and Vachon, B. (2008). Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, 1(3):219–337.
- [Bovik et al., 1990] Bovik, A. C., Clark, M., and Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73.
- [Boykov and Funka-Lea, 2006] Boykov, Y. and Funka-Lea, G. (2006). Graph cuts and efficient N-D segmentation. *International Journal of Computer Vision*, 70(2):109–131.

- [Boykov and Kolmogorov, 2004] Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.
- [Boykov et al., 2001] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.
- [Brandt and Algazi, 1992] Brandt, J. and Algazi, V. (1992). Continuous skeleton computation by Voronoi diagram. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 55(3):329–337.
- [Brlek et al., 2005] Brlek, S., Labelle, G., and Lacasse, A. (2005). The discrete Green theorem and some applications in discrete geometry. *Theoretical Computer Science*, 346(2-3):200–225.
- [Brox et al., 2006] Brox, T., Bruhn, A., and Weickert, J. (2006). Variational motion segmentation with level sets. In *European Conference on Computer Vision (ECCV)*, volume 3951 of *LNCS*, pages 471–483, Graz, Austria. Springer.
- [Brox and Cremers, 2009] Brox, T. and Cremers, D. (2009). On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional. *International Journal of Computer Vision*, 84(2):184–193.
- [Brox and Malik, 2011] Brox, T. and Malik, J. (2011). Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513.
- [Brutzer et al., 2011] Brutzer, S., Höferlin, B., and Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1937–1944.
- [Bunke and Allermann, 1983] Bunke, H. and Allermann, G. (1983). Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253.
- [Carlson, 1966] Carlson, B. (1966). Some inequalities for hypergeometric functions. *Proceedings of the American Mathematical Society*, 17:32–39.
- [Carlson, 2010] Carlson, B. (2010). Elliptic integrals. In Olver, F., Lozier, D., Boisvert, R., and Clark, C., editors, *NIST Handbook of Mathematical Functions*, chapter 19, pages 485–522. Cambridge University Press.
- [Caselles et al., 1997] Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79.
- [Chan and Vasconcelos, 2008] Chan, A. and Vasconcelos, N. (2008). Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):909–926.
- [Chan and Vese, 2001] Chan, T. and Vese, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.
- [Chandra and Kokkinos, 2016] Chandra, S. and Kokkinos, I. (2016). Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian CRFs. In *European Conference on Computer Vision (ECCV)*, pages 402–418.
- [Chaussard et al., 2011] Chaussard, J., Couprie, M., and Talbot, H. (2011). Robust skeletonization using the discrete λ -medial axis. *Pattern Recognition Letters*, 32(9):1384–1394.

- [Chazal and Lieutier, 2005] Chazal, F. and Lieutier, A. (2005). The λ -medial axis. *Graphical Models*, 67(4):304–331.
- [Chazelle and Edelsbrunner, 1992] Chazelle, B. and Edelsbrunner, H. (1992). An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM*, 39(1):1–54.
- [Chen et al., 2020a] Chen, B., Ling, H., Zeng, X., Gao, J., Xu, Z., and Fidler, S. (2020a). ScribbleBox: Interactive annotation framework for video object segmentation. In *European Conference on Computer Vision (ECCV)*, volume 13, pages 293–310.
- [Chen and Cohen, 2018] Chen, D. and Cohen, L. (2018). Fast asymmetric fronts propagation for image segmentation. *Journal of Mathematical Imaging and Vision*, 60(6):766–783.
- [Chen et al., 2017] Chen, D., Mirebeau, J.-M., and Cohen, L. (2017). Global minimum for a Finsler elastica minimal path approach. *International Journal of Computer Vision*, 122(3):458–483.
- [Chen et al., 2018] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.
- [Chen et al., 2020b] Chen, Y.-W., Tsai, Y.-H., Lin, Y.-Y., and Yang, M.-H. (2020b). VOSTR: Video object segmentation via transferable representations. *International Journal of Computer Vision*, 128(4):931–949.
- [Chiverton et al., 2008] Chiverton, J., Xie, X., and Mirmehdi, M. (2008). Tracking with active contours using dynamically updated shape information. In *British Machine Vision Conference (BMVC)*, pages 253–262, Leeds, UK.
- [Coeurjolly and Montanvert, 2007] Coeurjolly, D. and Montanvert, A. (2007). Optimal separable algorithms to compute the reverse Euclidean distance transformation and discrete medial axis in arbitrary dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):437–448.
- [Cohen, 1991] Cohen, L. (1991). On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218.
- [Cohen and Cohen, 1993] Cohen, L. and Cohen, I. (1993). Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147.
- [Cohen and Kimmel, 1997] Cohen, L. and Kimmel, R. (1997). Global minimum for active contour models: a minimal path approach. *International Journal of Computer Vision*, 24(1):57–78.
- [Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., and Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575.
- [Cordts et al., 2016] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The Cityscapes dataset for semantic urban scene understanding. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223.
- [Cormen et al., 2009] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. (2009). *Introduction to algorithms, 3rd edition*. The MIT Press.
- [Couprie et al., 2011] Couprie, C., Grady, L., Najman, L., and Talbot, H. (2011). Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1384–1399.
- [Crandall et al., 1992] Crandall, M., Ishii, H., and Lions, P.-L. (1992). User’s guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27:1–67.

- [Cuturi, 2013] Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2292–2300.
- [Demirci et al., 2006] Demirci, M., Shokoufandeh, A., Keselman, Y., Bretzner, L., and Dickinson, S. (2006). Object recognition as many-to-many feature matching. *International Journal of Computer Vision*, 69(2):203–222.
- [Dimitrov et al., 2003] Dimitrov, P., Damon, J., and Siddiqi, K. (2003). Flux invariants for shape. In *Computer Vision and Pattern Recognition*, pages 835–841.
- [do Carmo, 1976] do Carmo, M. (1976). *Differential geometry of curves and surfaces*. Prentice-Hall.
- [Dolcetta et al., 2002] Dolcetta, I., Vita, S. F., and March, R. (2002). Area preserving curve shortening flows: from phase transitions to image processing. *Interfaces and Free Boundaries*, 4(4):325–344.
- [Doretto et al., 2003] Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109.
- [Elgammal et al., 2002] Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. (2002). Background and foreground modeling using nonparametric kernel density for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163.
- [Eppstein, 1998] Eppstein, D. (1998). Finding the k shortest paths. *SIAM Journal of Computing*, 28(2):652–673.
- [Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.
- [Falcão et al., 2004] Falcão, A., Stolfi, J., and Lotufo, R. (2004). The Image Foresting Transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29.
- [Falcão et al., 2000] Falcão, A., Udupa, J., and Miyazawa, F. (2000). An ultra-fast user-steered image segmentation paradigm: Live Wire on the Fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62.
- [Farabet et al., 2013] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929.
- [Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.
- [Ferradans et al., 2014] Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J. F. (2014). Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882.
- [Forsyth, 1960] Forsyth, A. (1960). *Calculus of variations*. Dover Publications, New York.
- [Freedman and Zhang, 2004] Freedman, D. and Zhang, T. (2004). Active contours for tracking distributions. *IEEE Transactions on Image Processing*, 13(4):518–526.
- [Fundana et al., 2008] Fundana, K., Overgaard, N., and Heyden, A. (2008). Variational segmentation of image sequences using region-based active contours and deformable shape priors. *International Journal of Computer Vision*, 80(3):289–299.
- [Gage, 1986] Gage, M. (1986). On an area-preserving evolution equation for plane curves. *Contemporary Mathematics*, 51:51–62.

- [Gan et al., 2017] Gan, C., Li, Y., Li, H., Sun, C., and Gong, B. (2017). VQS: Linking segmentations to questions and answers for supervised attention in VQA and question-focused semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1829–1838.
- [Gao et al., 2010] Gao, X., Xiao, B., Tao, D., and Li, X. (2010). A survey of graph edit distance. *Pattern Analysis and Applications*, 13(1):113–129.
- [García García and Bouwmans, 2020] García García, B. and Bouwmans, T. Rosales-Silva, A. (2020). Background subtraction in real applications: Challenges, current models and future directions. *Computer Science Review*, 35.
- [Ge and Fitzpatrick, 1996] Ge, Y. and Fitzpatrick, J. (1996). On the generation of skeletons from discrete Euclidean distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1055–1066.
- [Giblin and Kimia, 2004] Giblin, P. and Kimia, B. (2004). A formal classification of the 3D medial axis points and their local geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):238–251.
- [Goëau et al., 2011] Goëau, H., Bonnet, P., Joly, A., Boujema, N., Barthélémy, D., Molino, J.-F., Birnbaum, P., Mouysset, E., and Picard, M. (2011). The Image CLEF 2011 plant images classification task. In *CLEF*.
- [Goëau et al., 2012] Goëau, H., Bonnet, P., Joly, A., Yahiaoui, I., Barthélémy, D., Boujema, N., and Molino, J.-F. (2012). The Image CLEF 2012 plant identification task. In *CLEF*.
- [Grady, 2006] Grady, L. (2006). Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1768–1783.
- [Harris, 1978] Harris, F. (1978). On the use of windows for harmonic analysis with the Discrete Fourier Transform. *Proceedings of the IEEE*, 66(1):51–83.
- [He et al., 2020] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397.
- [He et al., 2004] He, X., Zemel, R., and Carreira-Perpiñán, M. (2004). Multiscale conditional random fields for image labeling. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 695–702.
- [Hesselink and Roerdink, 2008] Hesselink, W. and Roerdink, J. (2008). Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2204–2217.
- [Horn and Schunck, 1981] Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3):187–203.
- [Ivins and Porri, 1995] Ivins, J. and Porri, J. (1995). Active region models for segmenting textures and colours. *Image and Vision Computing*, 13(5):431–438.
- [Jacob et al., 2004] Jacob, M., Blu, T., and Unser, M. (2004). Efficient energies and algorithms for parametric snakes. *IEEE Transactions on Image Processing*, 13(9):1231–1244.
- [Jalba et al., 2016] Jalba, A., Sobiecki, A., and Telea, A. (2016). An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):30–45.
- [Jonker and Volgenant, 1987] Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.
- [Kass et al., 1988] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331.

- [Katz and Pizer, 2003] Katz, R. and Pizer, S. (2003). Untangling the Blum Medial Axis Transform. *International Journal of Computer Vision*, 55(2–3):139–153.
- [Keselman et al., 2003] Keselman, Y., Shokoufandeh, A., Demirci, M., and Dickinson, S. (2003). Many-to-many graph matching via metric embedding. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 850–857.
- [Kim et al., 2010] Kim, D., Yun, I., and Lee, S. (2010). Attributed relational graph matching based on the nested assignment structure. *Pattern Recognition*, 43(3):914–928.
- [Kim et al., 2005] Kim, J., Fisher, J., Yezzi, A., Çetin, M., and Willsky, A. (2005). A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502.
- [Kimmel et al., 1995] Kimmel, R., Shaked, D., Kiryati, N., and Bruckstein, A. (1995). Skeletonization via distance maps and level sets. *Computer Vision and Image Understanding*, 62(3):382–391.
- [Kipf and Welling, 2017] Kipf, T. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.
- [Kontschieder et al., 2014] Kontschieder, P., Rota Bulò, S., Pelillo, M., and Bischof, H. (2014). Structured labels in random forests for semantic labelling and object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2104–2116.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114.
- [Krizhevsky et al., 2017] Krizhevsky, A., Sutskever, I., and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.
- [Kuhn, 1955] Kuhn, H. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.
- [Lam et al., 1992] Lam, L., Lee, S., and Suen, C. (1992). Thinning methodologies - a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:869–885.
- [Lankton and Tannenbaum, 2008] Lankton, S. and Tannenbaum, A. (2008). Localizing region-based active contours. *IEEE Transactions on Image Processing*, 17(11):2029–2039.
- [Latecki et al., 2007] Latecki, L., Wang, Q., Koknar-Tezel, S., and Megalooikonomou, V. (2007). Optimal subsequence bijection. In *International Conference on Data Mining*, pages 565–570.
- [Lawler, 1976] Lawler, E. (1976). *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston.
- [Leborgne, 2016] Leborgne, A. (2016). *Appariement de formes basé sur une squelettisation hiérarchique*. PhD thesis, Université de Lyon (INSA de Lyon)).
- [Lee, 1982] Lee, D. (1982). Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4):363–369.
- [Lee, 2005] Lee, D.-S. (2005). Effective Gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832.
- [Levenshtein, 1966] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

- [Li et al., 2008] Li, C., Kao, C., Gore, J., and Ding, Z. (2008). Minimization of region-scalable fitting energy for image segmentation. *IEEE Transactions on Image Processing*, 17(10):1940–1949.
- [Li et al., 2006] Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. (2006). Lazy snapping. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2004*, 23(3):303–308.
- [Li et al., 2018] Li, Z., Chen, Q., and Koltun, V. (2018). Interactive image segmentation with latent diversity. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 577–585.
- [Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. (2014). Microsoft COCO: Common Objects in COntext. In *European Conference on Computer Vision (ECCV)*, volume 5, pages 740–755.
- [Ling et al., 2019] Ling, H., Gao, J., Kar, A., Chen, W., and Fidler, S. (2019). Fast interactive object annotation with curve-GCN. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5257–5266.
- [Liu et al., 2013] Liu, H., Wu, Z., Zhang, X., and Hsu, D. (2013). A skeleton pruning algorithm based on information fusion. *Pattern Recognition Letters*, 34(10):1138–1145.
- [Liu et al., 2020] Liu, L., Chen, D., Cohen, L., Wu, J., Pâques, M., and Shu, H. (2020). Anisotropic tubular minimal path model with fast marching front freezing scheme. *Pattern Recognition*, 104:107349.
- [Loncaric, 1998] Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001.
- [Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.
- [Lowe, 2004] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- [Macrini et al., 2011a] Macrini, D., Dickinson, S., Fleet, D., and Siddiqi, K. (2011a). Bone graphs: medial shape parsing and abstraction. *Computer Vision and Image Understanding*, 115(7):1044–1061.
- [Macrini et al., 2011b] Macrini, D., Dickinson, S., Fleet, D., and Siddiqi, K. (2011b). Object categorization using bone graphs. *Computer Vision and Image Understanding*, 115(8):1187–1206.
- [Macrini et al., 2008] Macrini, D., Siddiqi, K., and Dickinson, S. (2008). From skeletons to bone graphs: Medial abstraction for object recognition. In *Computer Vision and Pattern Recognition*, pages 1–8.
- [Malladi et al., 1995] Malladi, R., Sethian, J., and Vemuri, B. (1995). Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175.
- [Manjunath and Ma, 1996] Manjunath, B. and Ma, W. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.
- [Mansouri, 2002] Mansouri, A. (2002). Region tracking via level set PDEs without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):947–961.
- [Mansouri et al., 2006] Mansouri, A., Mitiche, A., and Vazquez, C. (2006). Multiregion competition: a level set extension of region competition to multiple region image partitioning. *Computer Vision and Image Understanding*, 101(3):137–150.
- [Marcus, 2018] Marcus, G. (2018). Deep learning: a critical appraisal. *arXiv*, 1801.00631.
- [Marie et al., 2016] Marie, R., Labbani-lgbida, O., and Mouaddib, E. (2016). The Delta Medial Axis: A fast and robust algorithm for filtered skeleton extraction. *Pattern Recognition*, 56:25–39.

- [Meijster et al., 2000] Meijster, A., Roerdink, J., and Hesselink, W. (2000). A general algorithm for computing distance transforms in linear time. *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 331–340.
- [Michailovich et al., 2007] Michailovich, O., Rathi, Y., and Tannenbaum, A. (2007). Image segmentation using active contours driven by the Bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, 16(11):2787–2801.
- [Miranda et al., 2012] Miranda, P., Falcão, A., and Spina, T. (2012). Riverbed: A novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Transactions on Image Processing*, 21(6):3042–3052.
- [Mirebeau and Portegies, 2019] Mirebeau, J.-M. and Portegies, J. (2019). Hamiltonian Fast Marching: A numerical solver for anisotropic and non-holonomic Eikonal PDEs. *Image Processing On Line*, 9:47–93.
- [Mitiche and Ben Ayed, 2011] Mitiche, A. and Ben Ayed, I. (2011). *Variational and Level Set Methods in Image Segmentation*. Springer, Topics in Signal Processing.
- [Mokhtarian and Mackworth, 1992] Mokhtarian, F. and Mackworth, A. (1992). A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805.
- [Montero and Lang, 2012] Montero, A. S. and Lang, J. (2012). Skeleton pruning by contour approximation and the integer medial axis transform. *Computers and Graphics*, 36(5):477–487.
- [Mortensen and Barrett, 1998] Mortensen, E. and Barrett, W. (1998). Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60(5):349–384.
- [Mumford and Shah, 1989] Mumford, D. and Shah, J. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685.
- [Munkres, 1957] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.
- [Nakhmani and Tannenbaum, 2012] Nakhmani, A. and Tannenbaum, A. (2012). Self-crossing detection and location for parametric active contours. *IEEE Transactions on Image Processing*, 21(7):3150–3156.
- [Neuman, 2011] Neuman, E. (2011). Inequalities and bounds for generalized complete elliptic integrals. *Journal of Mathematical Analysis and Applications*, 373(1):203–213.
- [Nguyen et al., 2007] Nguyen, H., Ji, Q., and Smeulders, A. (2007). Spatio-temporal context for robust multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):52–64.
- [Ogniewicz and Kübler, 1995] Ogniewicz, R. and Kübler, O. (1995). Hierarchic Voronoi Skeletons. *Pattern recognition*, 28(3):343–359.
- [Oh et al., 2019] Oh, S., Lee, J.-Y., Xu, N., and Kim, S. (2019). Fast user-guided video object segmentation by interaction-and-propagation networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5247–5256.
- [O'Mahony et al., 2019] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Velasco-Hernández, G.-A., Krpalkova, L., Riordan, D., and Walsh, J. (2019). Deep learning vs. traditional computer vision. In *Computer Vision Conference (CVC)*, volume 1, pages 128–144.
- [Osher and Sethian, 1988] Osher, S. and Sethian, J. (1988). Fronts propagation with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49.

- [Palágyi, 2008] Palágyi, K. (2008). A 3D fully parallel surface-thinning algorithm. *Theoretical Computer Science*, 406(1-2):119–135.
- [Papadakis and Bugeau, 2011] Papadakis, N. and Bugeau, A. (2011). Tracking with occlusions via graphcuts. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 33(1):17–28.
- [Paragios and Deriche, 2002] Paragios, N. and Deriche, R. (2002). Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247.
- [Paragios and Deriche, 2005] Paragios, N. and Deriche, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3):259–282.
- [Perera et al., 2000] Perera, A., Tsai, C.-L., Flatland, R., and Stewart, C. (2000). Maintaining valid topology with active contours: theory and application. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1496–1502, Hilton Head, SC, USA.
- [Péteri et al., 2010] Péteri, R., Fazekas, S., and Huiskes, M. (2010). DynTex : a comprehensive database of dynamic textures. *Pattern Recognition Letters*, 31(12):1627–1632.
- [Peyré et al., 2010] Peyré, G., Pechaud, M., Keriven, R., and Cohen, L. (2010). Geodesic methods in computer vision and graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(3-4):197–397.
- [Pizer et al., 1987] Pizer, S., Oliver, W., and Bloomberg, S. (1987). Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):505–511.
- [Pizer et al., 2003] Pizer, S., Siddiqi, K., Székely, G., Damon, J., and Zucker, S. (2003). Multiscale medial loci and their properties. *International Journal of Computer Vision*, 55(2-3):155–179.
- [Pudney, 1998] Pudney, C. (1998). Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, 72(3):404–413.
- [Puzicha et al., 1997] Puzicha, J., Hofmann, T., and Buhmann, J. (1997). Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 267–272.
- [Pérez et al., 2002] Pérez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *European Conference on Computer Vision (ECCV)*, volume 2350 of *LNCS*, pages 661–675, Copenhagen, Denmark. Springer.
- [Raussen, 2008] Raussen, M. (2008). *Elementary Differential Geometry: Curves and Surfaces*. Department of Mathematical Sciences, Aalborg University, Denmark.
- [Revaud et al., 2016] Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2016). DeepMatching: hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323.
- [Riesen et al., 2010] Riesen, K., Jiang, X., and Bunke, H. (2010). *Exact and Inexact Graph Matching: Methodology and Applications*, volume 40, pages 217–247. Springer, Boston.
- [Ronfard, 1994] Ronfard, R. (1994). Region based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251.
- [Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 234–241.
- [Rother et al., 2004] Rother, C., Kolmogorov, V., and Blake, A. (2004). GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314.

- [Royden and Fitzpatrick, 2010] Royden, H. and Fitzpatrick, P. (2010). *Real analysis, 4th edition*. Pearson.
- [Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. (2000). The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.
- [Sapiro and Tannenbaum, 1995] Sapiro, G. and Tannenbaum, A. (1995). Area and length preserving geometric invariant scale-spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):67–72.
- [Sebastian and Kimia, 2005] Sebastian, T. and Kimia, B. (2005). Curves vs. skeletons in object recognition. *Signal Processing*, 85(2):247–263.
- [Sebastian et al., 2004] Sebastian, T., Klein, P., and Kimia, B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571.
- [Serino and Sanniti di Baja, 2016] Serino, L. and Sanniti di Baja, G. (2016). A new strategy for skeleton pruning. *Pattern Recognition Letters*, 76:41–48.
- [Sethian, 1996] Sethian, J. (1996). A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Science*, 93(4):1591–1595.
- [Sethian, 1999] Sethian, J. (1999). *Level Sets Methods and Fast Marching Methods, 2nd edition*. Cambridge University Press.
- [Shelhamer et al., 2017] Shelhamer, E., Long, J., and Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651.
- [Shen et al., 2011] Shen, W., Bai, X., Hu, R., Wang, H., and Latecki, L.-J. (2011). Skeleton growing and pruning with bending potential ratio. *Pattern Recognition*, 44(2):196–209.
- [Shen et al., 2013] Shen, W., Bai, X., Hu, R., Wang, H., and Latecki, L.-J. (2013). Shape clustering: Common structure discovery. *Pattern Recognition*, 46(2):539–550.
- [Shotton et al., 2006] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision (ECCV)*, pages 1–15.
- [Siddiqi et al., 2002] Siddiqi, K., Bouix, S., Tannenbaum, A., and Zucker, S. (2002). Hamilton-Jacobi skeletons. *International Journal of Computer Vision*, 48(3):215–231.
- [Siddiqi et al., 1999] Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32.
- [Sofiiuk et al., 2020] Sofiiuk, K., Petrov, I., Barinova, O., and Konushin, A. (2020). F-BRS: Rethinking backpropagating refinement for interactive segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 8620–8629.
- [Spielman and Teng, 2004] Spielman, D. and Teng, S.-H. (2004). Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463.
- [Stauffer and Grimson, 1999] Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2246–2252, Ft Collins, USA.
- [Stauffer and Grimson, 2000] Stauffer, C. and Grimson, W. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757.
- [Tang, 1982] Tang, G. (1982). A discrete version of Green's theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):242–249.

- [Teague, 1980] Teague, M. (1980). Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930.
- [Tomizawa, 1971] Tomizawa, N. (1971). On some techniques useful for the solution of transportation problems. *Networks*, 1:173–194.
- [Torsello and Hancock, 2006] Torsello, A. and Hancock, E. (2006). Correcting curvature-density effects in the Hamilton-Jacobi skeleton. *IEEE Transactions on Image Processing*, 15(4):877–891.
- [Tsitsiklis, 1995] Tsitsiklis, J. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538.
- [Vamanamurthy and Vuorinen, 1994] Vamanamurthy, M. and Vuorinen, M. (1994). Inequalities for means. *Journal of Mathematical Analysis and Applications*, 183(1):155–166.
- [van Brunt, 2004] van Brunt, B. (2004). *The calculus of variations*. Springer.
- [Vezhnevets et al., 2012] Vezhnevets, A., Ferrari, V., and Buhmann, J. (2012). Weakly supervised structured output learning for semantic segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 845–852.
- [Vicente et al., 2008] Vicente, S., Kolmogorov, V., and Rother, C. (2008). Graph cut based image segmentation with connectivity priors. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, USA.
- [Wang et al., 2006] Wang, H., Wang, X., Zhou, Y., and Yang, J. (2006). Colour texture segmentation using quaternion-Gabor filters. In *IEEE International Conference on Image Processing (ICIP)*, page 745–748.
- [Wang et al., 2019] Wang, W., Shen, J., Porikli, F., and Yang, R. (2019). Semi-supervised video object segmentation with super-trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):985–998.
- [Weinzaepfel et al., 2013] Weinzaepfel, P., Revaud, J., Harchaoui, Z., and Schmid, C. (2013). DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1385–1392.
- [Wen et al., 2015] Wen, L., Du, D., Lei, Z., Li, S., and Yang, M.-H. (2015). JOTS: Joint online tracking and segmentation. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2226–2234.
- [Whitney, 1937] Whitney, H. (1937). On regular closed curves in the plane. *Compositio Mathematica*, 4:276–284.
- [Wolfram Research, a] Wolfram Research. The Wolfram Functions site: incomplete elliptic integral of the first kind. <http://functions.wolfram.com/EllipticIntegrals/EllipticF>.
- [Wolfram Research, b] Wolfram Research. The Wolfram Functions site: incomplete elliptic integral of the second kind. <http://functions.wolfram.com/EllipticIntegrals/EllipticE2>.
- [Wren et al., 1997] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: realtime tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.
- [Xu and Prince, 1998] Xu, C. and Prince, J. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369.
- [Xu et al., 2009] Xu, Y., Wang, B., Liu, W., and Bai, X. (2009). Skeleton graph matching based on critical points using path similarity. In *Asian Conference on Computer Vision (ACCV)*, pages 456–465.

- [Yang et al., 2016] Yang, C., Tiebe, O., Shirahama, K., and Grzegorzek, M. (2016). Object matching with hierarchical skeletons. *Pattern Recognition*, 55:183–197.
- [Yang et al., 2008] Yang, M., Kpalma, K., and Ronsin, J. (2008). A survey of shape feature extraction techniques. In *Pattern Recognition Techniques, Technology and Applications*, pages 43–90. IntechOpen.
- [Yao et al., 2020] Yao, R., Lin, G., Xia, S., Zhao, J., and Zhou, Y. (2020). Video object segmentation and tracking: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(4):1–39.
- [Yasseen et al., 2016] Yasseen, Z., Verroust-Blondet, A., and Nasri, A. (2016). Shape matching by part alignment using extended chordal axis transform. *Pattern Recognition*, 57:115–135.
- [Yen, 1971] Yen, J. (1971). Finding the K shortest loopless paths in a network. *Management Science*, 17(11):712–716.
- [Yilmaz et al., 2006] Yilmaz, A., Javed, O., and Shah, M. (2006). Object tracking: a survey. *ACM Computing Surveys*, 38(4):13.
- [Zahn and Roskies, 1972] Zahn, C. and Roskies, R. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21(3):269–281.
- [Zhang and Freedman, 2005] Zhang, T. and Freedman, D. (2005). Improving performance of distribution tracking through background mismatch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):282–287.
- [Zhang et al., 2018] Zhang, Y., Hare, J., and Prügel-Bennett, A. (2018). Learning to count objects in natural images for visual question answering. In *International Conference on Learning Representations (ICLR)*.
- [Zhao et al., 2019] Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232.
- [Zhong and Sclaroff, 2003] Zhong, J. and Sclaroff, S. (2003). Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. In *IEEE International Conference on Computer Vision (ICCV)*, pages 44–50.
- [Zhou et al., 2001] Zhou, F., Feng, J., and Shi, Q. (2001). Texture feature based on local Fourier transform. In *IEEE International Conference on Image Processing (ICIP)*, pages 610–613.
- [Zhu and Yuille, 1996] Zhu, S. and Yuille, A. (1996). Region competition: unifying snakes, region growing, Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900.