

Scene Understanding from Image and Video: Segmentation, Depth Configuration and Inpainting

Maria Oliver Parera

TESI DOCTORAL UPF / 2018

Directores de la tesi
Dr. Coloma Ballester, Dr. Gloria Haro

Department of Information and Communication
Technologies



Sooner or later all things are
numbers.

TERRY PRATCHETT

Agraïments (Acknowledgements)

After five years of dedication to this thesis I would like to express my thankfulness to the people that have been closeby and have made it possible.

En primer lloc voldria donar les gràcies a les directores d'aquesta tesi. Moltes gràcies Glòria Haro i Coloma Ballester pel vostre temps, suport i ajut. També per l'esforç que heu fet perquè entengués cada detall d'aquest projecte i per ensenyar-me a treballar en el món de les imatges.

M'hauria agradat poder donar les gràcies al Prof. Vicent Caselles per haver dipositat la seva confiança en mi i haver-me donat l'oportunitat de començar aquesta etapa.

I also want to thanks all my colleagues from the image processing group at UPF. To my first two PhD mates Vadim and Roberto. Thanks for the funny times we had and for your friendship. A Lara por su gran ayuda en una parte de esta tesi y por animarse a venir siempre que le propongo alguna excursión. También a Pablo y a Mariella por haber dedicado su tiempo a ayudarme a entender partes importantes de esta tesi. To Baptiste, for your help in implementations of parts of this thesis. También quisiera agradecer a mis compañeros de grupo: Olga, Juan Fran, Juan, Felipe, Rida, Alexa, Joan, Ferran, Vanel y Pablo Zinemanas. Finalment gràcies als meus dos nous companys de doctorat, a la Patrícia i a l'Adrià Arbués, que han fet aquest any més entretingut. També vull agrair a l'Enric

Meinhardt la seva ajuda en parts d'aquesta tesis.

Thanks to the workmates from the department: al Javier Vázquez per les llargues converses al passadís i ser una gran font d'informació sobre qualsevol conferència. A la Itziar, per proposar-me excursions més esbojarrades que les meves. Thanks to Kalpi for her friendship from the first day I arrived at the university and being in touch despite the distance. Gràcies també al Jordi Pons, per la seva companyia als ICASSP. En especial, gracias Adrián Martín por todo tu apoyo durante este tiempo, ya sea en la distancia o en persona, por todas las charlas y tu gran ayuda.

Gràcies a la gent del màster i del grau, per continuar mantenint el contacte tot i que tots hem anat en direccions molt diferents: al Marc Fraile i al Jaume, per trobar temps per quedar cada cop que tornen a casa. Thanks to Sandra, Pedram and Silvia to find time for meeting as oftenly as possible. I especialment a la Carme, per haver insistit, tot i la meva resistència, a introduir-me al CAU.

Voldria donar les gràcies a la gent del conservatori per la seva comprensió, especialment a la Carme, al Jordi i a l'Adrià.

Voldria agrair a la gent de Terrassa tot el seu suport, especialment a la Laura, als dos Paus, la Raquel i el Jordi, perquè tot i que ja no tenim tant temps per veure'ns sé que sempre puc comptar amb vosaltres. També a l'Eyra, la Neus, l'Aaron, la Laura, l'Adriana, la Tanit, la Judit, a l'Anna, al Keko i al Torrents.

Voldria agrair el seu suport a la família. Especialment, a les meves cosines la Isaura i la Griselda, per haver sigut com dues germanes grans. Gràcies també a la Isaura i en Jordi per haver-me acollit al Canadà. També a la Griselda per haver-me escoltant durant aquests anys. Gràcies al meu germà Josep per haver-me aguantat durant tota la seva vida. I als meus pares, per haver-me fet créixer com a persona i haver-se preocupat sempre de la meva educació, per haver-me donat suport quan se'm va acudir estudiar mates i continuar amb un doctorat. Finalment, donar les gràcies als avis per haver-nos transmès les ganes d'aprendre i animar-nos sempre a continuar estudiant.

Abstract

In this thesis we aim at analyzing images and videos at the object level, with the goal of decomposing the scene into complete objects that move and interact among themselves. The thesis is divided in three parts. First, we propose a segmentation method to decompose the scene into shapes. Then, we propose a probabilistic method, which works with shapes or objects at two different depths, to infer which objects are in front of the others, while completing the ones which are partially occluded. Finally, we propose two video related inpainting methods. On one hand, we propose a binary video inpainting method that relies on the optical flow of the video in order to complete the shapes across time taking into account their motion. On the other hand, we propose a method for optical flow inpainting that takes into account the information from the frames.

Resum

Aquesta tesi té per objectiu analitzar imatges i vídeos a nivell d'objectes, amb l'objectiu de descompondre l'escena en objectes complets que es mouen i interaccionen entre ells. La tesi està dividida en tres parts. En primer lloc, proposem un mètode de segmentació per descompondre l'escena en les formes que la componen. A continuació, proposem un mètode probabilístic, que considera les formes o objectes en dues profunditats de l'escena diferents, i infereix quins objectes estan davant dels altres, completant també els objectes parcialment ocults. Finalment, proposem dos mètodes relacionats amb el vídeo inpainting. Per una banda, proposem un mètode per vídeo inpainting binari que utilitza el flux òptic del vídeo per completar les formes al llarg del temps, tenint en compte el seu moviment. Per l'altra banda, proposem un mètode per inpainting de flux òptic que té en compte la informació provinent dels frames.

Preface

The visual information we extract from our perception of the world is formed by a continuum of objects interacting among them, instead of the small particles that form it. Objects play a crucial role in our understanding of the environment: we perceive them as the single entities that allow us to interact with our surroundings. When we take a picture or film a video we want to capture the reality we are observing. In this way, although we store the information in pixels, we look for the objects conforming it. That is, we look for the identifiable portions of the image that can be interpreted as single units. Obviously, these basic units will change depending on the application. For example, if we are working on face recognition we will be interested in the eyes, mouth and nose of the face, but, if we work in people tracking we would be more interested in considering the whole person as a single unity.

Changing from the pixel unity to objects level has a lot of advantages in many areas, not only in computer vision but also in robotics or industrial engineering. For instance, although robots see the world through sensors that receive pixel-data, we expect them to perceive the surrounding world in the same way we do. We need the robots to be able to recognize and interact with the objects conforming the scenes. Besides, objects have a number of attributes such as volume and shape, texture or interrelation properties such as adjacency or T-junctions. These attributes make objects to be richer instances than individual pixels, which can help in a classification process.

This manuscript is focused on working from the object level per-

spective. We propose a segmentation model to decompose the image scene into regions or shapes. Then, we propose to solve two other problems which have as input an image or video classified in objects or shapes.

Image Segmentation consists in partitioning the image into regions that share common features, such as color or texture. To this goal we propose a variational method that considers adaptive patches to characterize, in an affine invariant way, the local structure of each region of the image. The patches are computed using an affine covariant structure tensor defined at every pixel of the image domain, so that they can automatically adapt its shape and size. The proposed segmentation model uses an L^1 -norm fidelity term and the total variation of relaxed fuzzy membership functions as an approximation to the length of the boundaries of the segmented regions. The output of the method is a partition of the image in regions together with a representative patch of the texture of each region.

The **Scene Structure** problem involves the recovery of the relative order structure of the objects from a planar image, where some objects may occlude others. We propose to estimate the interpretation of the scene by integrating some global and local cues while also providing both the complete disoccluded objects that form the scene and their ordering according to depth. Our method first computes several distal scenes which are compatible with the proximal planar image. To compute these different hypothesized scenes, we propose a perceptually inspired object disocclusion method, which works by minimizing the Euler's elastica as well as by incorporating the reliability of partially occluded contours and the convexity of the disoccluded objects. Then, to estimate the perceptually preferred scene we rely on a Bayesian model and define probabilities taking into account the global complexity of the objects in the hypothesized scenes as well as the effort of bringing these objects in their relative position in the planar image.

Inpainting is the problem of recovering an image or video that is partially damaged. It is also used to remove undesired objects from the image or video and recovering the occluded objects. In video inpainting the missing information in the frames produces an incomplete optical flow. Therefore, video inpainting involves an extra challenge: recovering the optical flow. First, we propose a variational model for the completion of moving shapes through binary video inpainting that works by smoothly recovering the objects into the inpainting hole, taking into account the optical flow and motion occlusions. We solve it by a dynamic shape analysis algorithm based on threshold dynamics. The resulting inpainting algorithm diffuses the available information along the space and the visible trajectories of the pixels in time. Finally, we present an automatic method for optical flow inpainting. Given a video, each frame domain is endowed with a Riemannian metric based on the video pixel values. The missing optical flow is recovered by solving the Absolutely Minimizing Lipschitz Extension (AMLE) Partial Differential Equation on the Riemannian manifold. An efficient numerical algorithm is proposed using eikonal operators on finite graphs for nonlinear elliptic Partial Differential Equations.

Manuscript Outline

This document is organized in three parts. Each of them is devoted to analyze one of the problems.

In Part I we approach the segmentation problem. In Chapter 1 related works to the segmentation problem are reviewed, together with a motivation to use patch-based methods comparison. Chapter 2 is devoted to introduce the affine invariant tensors that are used to compute the patches and perform the segmentation. In Chapter 3 we explain our model, which uses the patch distance measure presented in Chapter 2, to decide to which region each pixel belongs to. The optimization algorithm is also explained in Chapter 3. Chapters 4

and 5 are respectively dedicated to the results and conclusions of the proposed method. Finally, as our algorithm involves the computation of the vector median, Appendix A is dedicated to the explanation of a fast algorithm to compute it.

Part II is devoted to the method that computes the most probable scene structure of an image. In Chapter 6 we do a review of the psychophysical studies that are involved in the image recovery of the 3D information given a single image. As our model also proposes to disocclude the occluded objects we also do a review of inpainting problems. In Chapter 7 we present the disocclusion method used to obtain the completed objects of the scene, together with a probabilistic method, which decides which is the most probable scene interpretation. The detailed algorithm is presented in Chapter 8. In Chapter 9 we show results on synthetic and real images and, finally, the conclusions are presented in Chapter 10.

The last Part of this Thesis, Part III, is devoted to present the methods for video inpainting. Chapter 11 introduces the binary video inpainting and optical flow problems. Chapter 12 is fully dedicated to the binary video inpainting problem. We present our method, which completes the shapes in a smooth way using both, the temporal and the spatial information. As we are considering binary videos it includes an extra difficulty for the minimization part. For this reason, we propose to use the Allen-Cahn equation to be able to find a solution of our problem. We explain the proposed minimization strategy, which is based on Threshold Dynamics. And, finally we provide some details on the code and examples on different applications. In Chapter 13 we present our model of optical flow inpainting. We propose to use the information from the frames in order to complete the optical flow on the missing areas using a geodesic distance. Finally, in Chapter 14 we provide the conclusions of this part together with some proposals of future work.

Contributions

Publications

- Oliver M., Raad L., Ballester C., Haro G., Motion Inpainting by an Image-Based Geodesic AMLE Method. *IEEE International Conference on Image Processing*, 2018. (Accepted)
- Oliver M., Haro G., Fedorov V., Ballester C., L1 Patch-Based Image Partitioning Into Homogeneous Textured Regions. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1558–1562, 2018.
- Oliver M., Palomares R. P., Ballester C., Haro G., Spatio-Temporal Binary Video Inpainting Via Threshold Dynamics. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1822–1826, 2017.
- Oliver M., Haro G., Dimiccoli M., Mazin B., Ballester C., A Computational Model for Amodal Completion. *Journal of Mathematical Imaging and Vision*. 56(3):511–534, 2016.

Conference Presentations

- Oliver M., Haro G., Fedorov V., Ballester C., L1 Patch-Based Image Partitioning Into Homogeneous Textured Regions. *SIAM Conference on Imaging Science*, June 2018, Bologna (Italy) **(Poster presentation)** BEST POSTER AWARD (2nd position)
- Oliver M., Palomares R. P., Ballester C., Haro G., Spatio-temporal binary video inpainting via threshold dynamics. *Annual Catalan Meeting on Computer Vision*, September 2017, Barcelona (Spain). **(Poster presentation)**

- Oliver M., Haro G., Dimiccoli M., Baptiste M., Ballester C., A Computational Model of Amodal Completion. *Annual Catalan Meeting on Computer Vision*, September 2016. Barcelona (Spain). **(Poster presentation)**
- Oliver M., Haro G., Dimiccoli M., Baptiste M., Ballester C., A Computational Model of Amodal Completion. *SIAM Conference on Imaging Science, minisymposium: Geometry-based Models in Image Processing*. May 2016, Albuquerque (New Mexico) **(Oral presentation)**

Contents

Preface	v
Manuscript Outline	vii
Contributions	ix
I IMAGE SEGMENTATION	1
1 Introduction	3
2 Affine Covariant Structure Tensors	7
2.1 Normalization to a Disc	9
2.2 An Affine Invariant Patch Similarity	10
3 The Segmentation Model	13
3.1 Optimization Algorithm	16
3.1.1 \mathbf{v} -subproblem	17
3.1.2 ω -subproblem: dual formulation algorithm . .	18
3.1.3 \mathbf{p} -subproblem	18
3.2 Initialization	19
3.3 Final segmented image and discs	20
4 Experimental Results	23
5 Conclusions and Future Work	33

A	Fast Weighted Median Vector Algorithm	35
II	SCENE STRUCTURE RECONSTRUCTION	45
6	Introduction: Scene Structure	47
6.1	Disocclusion	52
6.2	Depth ordering: a Bayesian approach	53
7	An Elastica Based Model for Scene Structure	55
7.1	Elastica-based object disocclusion	59
7.1.1	Initialization of the inpainting mask	60
7.2	Elastica-based probabilistic model	63
7.2.1	Conditional Probability: Relative Position Complexity	65
7.2.2	Prior probability: Objects Complexity	67
8	Algorithm and implementation details	71
8.1	Complete Algorithm	71
8.2	Inpainting Algorithm	73
8.3	Likelihood Estimation	75
9	Experimental results	77
9.1	Synthetic images	79
9.2	Real images	83
9.3	Shapes in front of a background	91
9.4	Discussion of failure cases	94
10	Conclusions and Future Work	99
III	VIDEO INPAINTING	101
11	Introduction	103

12 Video Shape Inpainting	109
12.1 A Threshold Dynamics Strategy	113
12.1.1 Numerical Details on the Diffusion Step	116
12.2 Applications	118
12.2.1 Damaged objects recovering	119
12.2.2 Objects removal	121
13 Optical Flow Inpainting	135
13.1 The geodesic AMLE on a finite graph	136
13.1.1 Numerical Multiscale Approach	139
13.1.2 Neighbourhood	141
13.1.3 Influence of the Image to Compute the Metric	143
13.2 Applications	145
14 Conclusions and Future Work	151
Bibliography	153

PART I:
IMAGE SEGMENTATION

The beauty of a living thing is not
the atoms that go into it, but the
way those atoms are put together.

CARL SAGAN

1

Introduction

In this Chapter we explain the segmentation problem and a review of some works related to it. We also provide a motivation to the use of patches.

Image simplification (or segmentation) is one of the central problems in image analysis and computer vision. The goal is to partition the image into regions which share common features – such as color, intensity, texture, or depth – while at the same time locate the most regular and accurate contours that define the sharp boundaries of these regions. Often, a representative feature of every region is also extracted. This information can be used, for example, for image cartooning, or image interpretation.

In the image segmentation literature variational approaches are among the most popular (Vese and Chan (2002); Strelakovsky and Cremers (2014); Xu et al. (2016a); Gu et al. (2017); Syu et al. (2017); Garamendi and Schiavi (2017)). From the literature, it is now well known that a good segmentation can be obtained by minimizing an appropriate energy functional. The Mumford-Shah functional is one of the most popular with this underlying variational criterion (see, e.g., Morel and Solimini (1994)). Let us briefly recall that Mumford and Shah (1989) defined the segmentation problem as a joint smoothing and edge detection problem by finding a piecewise smooth approximation v of the original image u , together with its disconti-

nuity set B , by minimizing:

$$E(B, v) = \int_{\Omega} (v - u)^2 dx + \mu \int_{\Omega \setminus B} |\nabla v|^2 dx + \lambda \ell(B) \quad (1.1)$$

where Ω is the image domain, B denotes the set where v is discontinuous and μ and λ are positive weighting parameters. The fidelity term forces to minimize the color value difference of each pixel from the original image and the segmented one; while the regularity term constraints v to be smooth everywhere except for the discontinuity set B , whose Hausdorff length should be as short as possible (third term). Due to the theoretical and numerical complexity of the Mumford-Shah functional, a simplification of the previous problem has been used where v is considered piecewise constant in $\Omega \setminus B$. Then the piecewise version of the functional reads as

$$E(B, v) = \int_{\Omega} (v - u)^2 dx + \lambda \ell(B). \quad (1.2)$$

The data term in the Mumford-Shah functional measures the similarity of the original image and the simplified one in an excessively local sense (at a pixel level). However, the gray level or color value of a single pixel is neither discriminative nor robust enough to be used as a comparison measure, specially for natural textured images. Consequently, the use of patches has become a common practice for establishing image similarities and correspondences in different image processing and computer vision applications, such as inpainting (Arias et al. (2011)), denoising (Buades et al. (2005); Kheradmand and Milanfar (2014); Fedorov and Ballester (2017)) or stereo matching (Einecke and Eggert (2015)).

Traditionally, these patches have been defined as squared or circular windows. One of the main problems with these windows is that if the center of the patch is close to an object boundary the patch contains mixed information from different objects. Simple improvements are the bilateral weights proposed by Tomasi and Manduchi (1998) or adaptive patches that try to follow the local geometry of

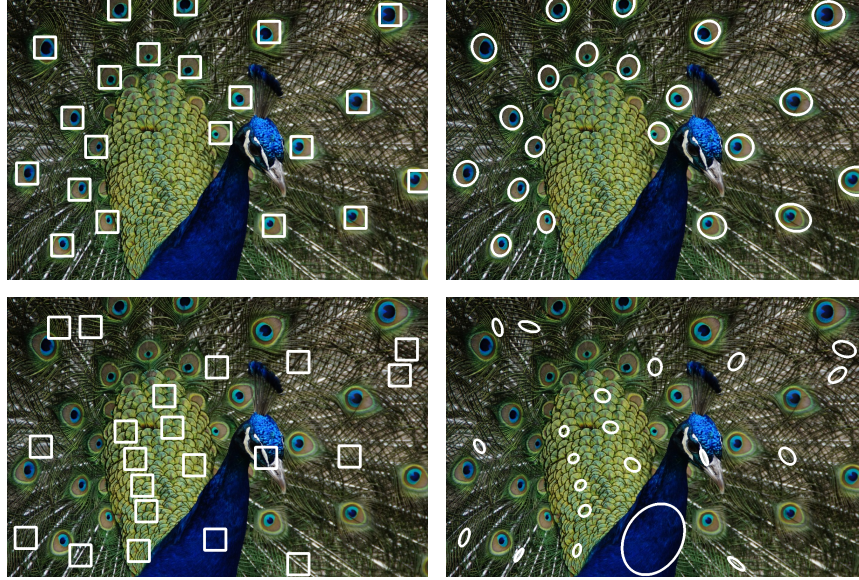
the image (Deledalle et al. (2012); Grewenig et al. (2011)). However, patches of fixed size have two main problems:

- Poor discrimination power when applied to textural structures not observable within the size of the neighborhood because of the wrong scale selected.
- Lack of robustness to transformations of the local texture due to perspective transformations or changes in the point of view.

In contrast, the patches proposed by Fedorov et al. (2015) and Fedorov (2016) are ellipses on the image domain that automatically adapt their shape and orientation to the local structure of the image. These kind of patches are defined using affine covariant structure tensors and, in combination with the affine invariant similarity measure introduced in Fedorov et al. (2015), allow to identify similar local image patterns that have suffered different affine transformations. In Figure 1.1 we present a comparison between the usual fixed-size squared windows and the affine covariant structure tensors: we compare patches associated to the same points $x \in \Omega$. We can observe that the elliptical tensor-based patches adapt better to the shapes.

Rousson et al. (2003); Sagiv et al. (2006) and Houhou et al. (2009) have used the classical structure tensor for texture segmentation purposes, however these methods are not robust to affine transformations in the texture.

We propose to use the L^1 version of the patch-based affine invariant similarity measure, proposed by Fedorov et al. (2015), in a Mumford-Shah-based segmentation functional to partition the image into regions that share the same local structure up to an affine distortion. Thanks to the use of the L^1 norm in our fidelity term, the proposed model also provides the representative sharp texture for each region, which consists of a patch containing contrast preserving texture as a result of a weighted median vector process. As a further consequence of the use of the L^1 norm the segmentation is robust to impulse noise and outliers. The L^1 norm was also used for segmentation purposes in Li et al. (2016) and Jung (2017).



(a) Fixed-size patches of size 30. (b) Affine covariant patches.

Figure 1.1: Comparison among fixed-size patches and adaptive patches.

Following the idea of Mory and Ardon (2007) and Li et al. (2016) we relax the functional by using fuzzy membership functions, which assume that each point can be in several regions simultaneously with a certain probability. As a generalization of the notion of characteristic functions, fuzzy membership functions satisfy the following two constraints:

- Nonnegativity constraint: the membership is non-negative for all pixels.
- Sum-to-one constraint: the sum of all the membership functions at each pixel equals one.

Moreover, fuzzy membership functions are considered to belong to $BV(\Omega; [0, 1])$, which is a convex set and guarantees the convergence and stability of many numerical optimization methods.

2

Affine Covariant Structure Tensors

In this Chapter we define the Affine Covariant Structure Tensors and the associated elliptical patches. We also show how to normalize these patches to a disc and provide a measure to compare them.

Let us consider an image $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ endowed with a Riemannian metric which, in our case, is given by an approximate structure tensor field. In other words, we consider an image-dependent tensor field T_u as a function that associates a structure tensor (a symmetric, positive semi-definite 2×2 matrix) to each point $x \in \Omega$. The structure tensor field is said to be *affine covariant* if, for any affinity A ,

$$T_{u_A}(x) = A^t T_u(Ax) A, \quad (2.1)$$

where $u_A(x) := u(Ax)$ denotes the affinely transformed version of u .

For each $x \in \Omega$, the affine structure tensor $T_u(x)$ has associated with it an elliptical region:

$$\mathcal{E}_{T_u}(x, r) = \{y : \langle T_u(x)(y - x), (y - x) \rangle \leq r^2\}, \quad (2.2)$$

where x is the center of the region and r is a free parameter that controls, together with the local texture, the size of the affine covariant neighborhood. Moreover, as this neighborhood comes from a structure tensor that is affine covariant, we have that

$$A \mathcal{E}_{T_u}(x, r) = \mathcal{E}_{T_u}(Ax, r). \quad (2.3)$$

This means that the structure tensors can be used to define affine covariant regions which transform properly via an affinity.

The affine covariant structure tensors and their associated neighborhoods, introduced in Fedorov et al. (2015), are computed using the following iterative scheme:

$$\begin{cases} T_u^{(k)}(x) = \frac{\int_{\mathcal{E}_{T_u^{(k-1)}}(x,r)} \nabla u(y) \otimes \nabla u(y) dy}{\text{Area}(\mathcal{E}_{T_u^{(k-1)}}(x,r))} \\ \mathcal{E}_{T_u^{(k)}}(x,r) = \{y : \langle T_u^{(k)}(x)(y-x), (y-x) \rangle \leq r^2\} \end{cases} \quad (2.4)$$

with the following initialization:

$$\mathcal{E}_{T_u^{(0)}}(x,r) = \{y : \langle \nabla u(x), (y-x) \rangle < r\}, \quad (2.5)$$

where ∇u denotes the gradient, \otimes the tensor product and $k \in \mathbb{N}$. From now on we will denote by $T_u(x)$ the affine invariant tensor $T_u^{(k)}$ for a fixed number of iterations ($k = 30$) and a given radius r ($r > 0$, which is left as a free parameter). Fedorov et al. (2015) state that after a few iterations T_u^k , obtained in (2.4), might cycle over a finite number of affine covariant tensors. Typically, a single tensor is found, but for some situations, like in corners, the process can iterate among two tensors or even three in some occasional situations. In any case, all the tensors found are guaranteed to be affine covariant. Therefore, the purpose of the iterative algorithm is not to ensure affine covariance, but to attenuate the dependency on the first iteration, which only depends on a single gradient and it is very sensitive to noise. We denote by $\mathcal{E}_{T_u}(x)$ the affine covariant neighborhood, and each patch of the image u will be denoted by $p_u(x)$, for $x \in \Omega$, and defined by the function $p_u(x) := p_u(x, \cdot)$ given by:

$$p_u(x, y) = u(y), \quad y \in \mathcal{E}_{T_u}(x). \quad (2.6)$$

As the computed patches may have different shapes and orientations we would not be able to compare them by using a simple

L^2 -distance. For this reason we propose to take advantage of the affine similarity measure defined in Section 2.2 and the transformation provided by the tensors to normalize the adaptive patches to a disc of fixed area.

2.1 Normalization to a Disc

Fedorov (2016) show that, given two affine covariant structure tensors, if $u_A = u \circ A$ we can extract the affine transformation between the corresponding elliptical patches up to a rotation. Indeed, for any affine transformation A there exists an orthogonal matrix R such that

$$A = T_u(Ax)^{-\frac{1}{2}} R T_{u_A}(x)^{\frac{1}{2}} \quad (2.7)$$

For its relevance, let us formalize and proof this last result.

Lemma 1. *Let u and v be two images, such that $u(z) = v(Az)$ for all $z \in \mathbb{R}^N$ and for a given invertible matrix A . Then, $A = T_v(y)^{-\frac{1}{2}} R T_u(x)^{\frac{1}{2}}$ for $y = Ax$ and some orthogonal matrix R .*

Proof. To prove that $A = T_v(y)^{-\frac{1}{2}} R T_u(x)^{\frac{1}{2}}$ is equivalent to proof that $T_v(y)^{\frac{1}{2}} A T_u(x)^{-\frac{1}{2}}$ is orthogonal. As $T_v(y)$ is affine covariant, by equation (2.1) we have that

$$T_u = A^t T_v A \quad (2.8)$$

$$T_u^{\frac{1}{2}} T_u^{\frac{1}{2}} = A^t T_v^{\frac{1}{2}} T_v^{\frac{1}{2}} A \quad (2.9)$$

$$Id = \left(T_u^{-\frac{1}{2}} A^t T_v^{\frac{1}{2}} \right) \left(T_v^{\frac{1}{2}} A^t T_u^{-\frac{1}{2}} \right) \quad (2.10)$$

$$Id = \left(T_v^{\frac{1}{2}} A T_u^{-\frac{1}{2}} \right)^t \left(T_v^{\frac{1}{2}} A^t T_u^{-\frac{1}{2}} \right) \quad (2.11)$$

From (2.10) to (2.11) we have used the symmetry of the structure tensors. Equation (2.11), proves that $T_v^{\frac{1}{2}} A T_u^{-\frac{1}{2}}$ is an orthogonal matrix. \square

Equation (2.7) provides an intuition of the geometric relationship between the structure tensors, the associated ellipses and the affinity. The application of (2.7) can be decomposed in three steps:

1. $T_{u_A}(x)^{\frac{1}{2}}$ transforms $\mathcal{E}_{T_u(x)}$ into a circle of radius r .
2. R is an appropriate rotation applied to the normalized patch.
3. $T_u(Ax)^{-\frac{1}{2}}$ maps the rotated normalized patch to the neighborhood $\mathcal{E}_u(Ax)$.

To fully determine the affinity A one needs to find the rotation R that aligns the image content of both discs. For this aim, the rotation is decomposed as

$$R = R_u(Ax)R_{u_A}^{-1}(x), \quad (2.12)$$

where $R_u(Ax)$ and $R_{u_A}^{-1}(x)$ are estimated from the image content of the discs. In practice, the rotation is calculated by aligning the dominant orientation of the normalized patches to the horizontal axis. The dominant orientations are computed using the histograms of oriented gradients as in the SIFT keypoints proposed by Lowe (2004). Notice that we can normalize each patch to a disc Δ_t using

$$\tilde{\mathbf{p}}_u(x) := \tilde{\mathbf{p}}_u(x, h) = u \left(x + T_u(x)^{-\frac{1}{2}} R_u(x)^{-1} h \right), \quad (2.13)$$

where $h \in \Delta_t$, $T_u(x)$ is the structure tensor associated to x and $R_u(x) = R(\theta)$ the rotation matrix of angle θ , being θ the local dominant orientation at x . Figure 2.1 presents the discs of several patches.

2.2 An Affine Invariant Patch Similarity

To compare two discs we give a step back and return to the patches, where we will derive a multiscale Patch Similarity Measure and see that, in practice, it is equivalent to a Disc Comparison Measure.

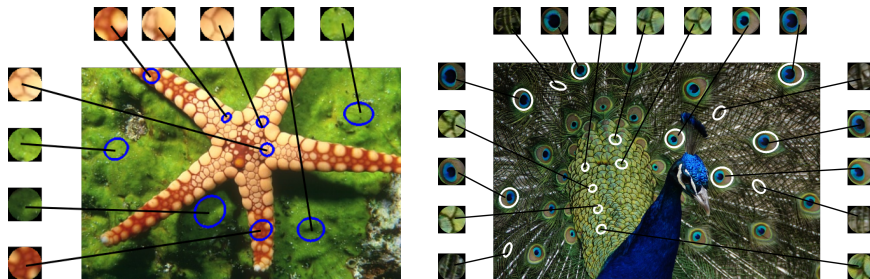


Figure 2.1: Adaptive patches associated to the affine invariant structure tensor for some pixels of the image, together with their normalization into a disc already rotated according to the local dominant orientation.

We are interested in comparing the neighborhoods $\mathbf{p}_u(x)$ and $\mathbf{p}_u(y)$ centered at x and y , respectively. In order to compare them we need a mapping among them. Equations (2.7) and (2.12) suggest the following definition:

$$P_R(x, y) = T_u(y)^{-\frac{1}{2}} R_u(y) R_u^{-1}(x) T_u(x)^{\frac{1}{2}}, \quad (2.14)$$

which can be interpreted as an affinity that maps the elliptical patch associated to $T_u(x)$ into the one associated to $T_u(y)$. The multiscale similarity measure we use to compare the patches is

$$\mathcal{D}_t^{\alpha, q}(\mathbf{p}_u(x), \mathbf{p}_u(y)) = \int_{\Delta_t} g_t(h) \left\| u\left(x + T_u(x)^{-\frac{1}{2}} R_u(x) h\right) - u\left(y + T_u(y)^{-\frac{1}{2}} R_u(y) h\right) \right\|_{L^q}^q dh, \quad (2.15)$$

where $q > 0$, $\|\cdot\|_{L^q}$ denotes the norm in L^q , $t > 0$ represents the scale of the patch and allows to control the support in the patch comparison, g_t is a Gaussian weighting function, and Δ_t is a disc of radius proportional to t where g_t has effective support. In practice, we transform both patches to a disc of radius t and compare the aligned normalized patches.

The similarity measure (2.15) was derived by Fedorov et al. (2015) as a computationally tractable approximation of the linear case of the

multiscale similarity measure introduced in Ballester et al. (2014). The authors show that all scale spaces of similarity measures $\mathcal{D}(t, x, y)$ satisfying a certain set of axioms are solutions of a family of degenerate parabolic partial differential equations. In their paper images are considered defined on Riemannian manifolds endowed with a metric defined by a tensor field. As we use the tensor T_u , which is affine covariant, like the Riemannian metric on the image plane, the associated similarity measure is also affine invariant.

3

The Segmentation Model

In this Chapter we explain the patch-affine segmentation model together with its optimization algorithm. As the proposed model considers characteristic functions, which belong to a non-convex set, we propose to relax them using fuzzy membership functions. Then, we propose a threshold to recover the characteristic functions of the original model. Moreover, as the model is not jointly convex we also propose an initialization.

Let $u : \Omega \rightarrow \mathbb{R}^M$ be a given image defined on $\Omega \subset \mathbb{R}^2$ with values in \mathbb{R}^M , where $M = 1$ for gray level images and $M = 3$ for standard color images.

In order to define our segmentation model, we associate to each pixel an affine invariant patch. Let \mathcal{P}_u be the set of all patches, also called manifold of patches, obtained from image u and defined using the affine covariant tensor metric $T_u(x)$ associated to u . That is,

$$\mathcal{P}_u = \{\mathbf{p}_u(x), \quad x \in \Omega\}. \quad (3.1)$$

As noticed in Chapter 2, thanks to the tensors, these elliptical patches can be considered defined on the *normalized disc* Δ_t .

We propose to simplify the set of all patches \mathcal{P}_u by estimating an optimal finite set of representative patches $\{\mathbf{p}_{\Omega_1}, \dots, \mathbf{p}_{\Omega_N}\}$, for a given $N \in \mathbb{N}$, where $\Omega = \cup_{i=1}^N \Omega_i$ is a partition of the image domain

into N disjoint open regions Ω_i , such that each region contains the pixels with similar patches and \mathbf{p}_{Ω_i} is the patch associated to the region Ω_i . In other words, we aim that Ω_i contains all the pixels with local homogeneous texture regardless of differences in the point of view or suffered perspective distortion. We propose to do it by minimizing the following energy:

$$E(\mathbf{p}, B) = \ell(B) + \lambda \int_{\Omega} \mathcal{D}_t^{\mathbf{a},1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}) dx \quad (3.2)$$

where $\mathbf{p} = \sum_i \mathbf{p}_{\Omega_i} \chi_{\Omega_i}$ is a piecewise constant patch function, i. e., it associates a homogeneous texture to each point x of a connected component Ω_i of $\Omega \setminus B$, and $B = \cup_{i=1}^N \partial\Omega_i$, being $\partial\Omega_i$ the boundary and χ_{Ω_i} the characteristic function of Ω_i . By analogy to (2.15) and by an abuse of notation we have denoted the patch similarity measure by

$$\begin{aligned} \mathcal{D}_t^{\mathbf{a},1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}) = \\ \int_{\Delta_t} g_t(h) \left\| u(x + T_u(x)^{-\frac{1}{2}} R(x)h) - \mathbf{p}_{\Omega_i}(h) \right\|_{L^1} dh \end{aligned} \quad (3.3)$$

for $x \in \Omega_i$. Our choice of $q = 1$ allows to obtain pure representative patches for each region.

As $\Omega \setminus B = \cup_{i=1}^N \Omega_i$, with $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j$, we can rewrite (3.2) as follows

$$\begin{aligned} E(\mathbf{p}, \boldsymbol{\chi}) = \sum_{i=1}^N \left[\int_{\Omega} |\nabla \chi_{\Omega_i}(x)| dx + \right. \\ \left. \lambda \int_{\Omega} \int_{\Delta_t} g_t(h) \left\| u(x + T_u(x)^{-\frac{1}{2}} R(x)h) - \mathbf{p}_{\Omega_i}(h) \right\|_{L^1} \chi_{\Omega_i}(x) dh dx \right], \end{aligned} \quad (3.4)$$

where $\boldsymbol{\chi} = (\chi_{\Omega_1}, \dots, \chi_{\Omega_N})$, with $\chi_{\Omega_i} \in BV(\Omega; \{0, 1\})$, and such that $\sum_i \chi_i(x) = 1, \forall x \in \Omega$. Again, $\mathbf{p} = \sum_i \mathbf{p}_{\Omega_i} \chi_{\Omega_i}$ is piecewise constant, i.e., the same patch \mathbf{p}_{Ω_i} is associated to all the pixels that belong to region Ω_i . Let us recall that the space $BV(\Omega; [0, 1])$ is the set of

functions $f \in L^1(\Omega)$ whose partial derivatives in the sense of distributions are measures with finite Total Variation (TV), where the Total Variation, for $f \in L^1_{\text{loc}}(\Omega)$ is defined as

$$TV(\chi) = \sup \left\{ - \int_{\Omega} f \operatorname{div} \phi dx : \phi \in C_c^{\infty}(\Omega; \mathbb{R}^N), \right. \\ \left. |\phi(x)| \leq 1, \forall x \in \Omega \right\}. \quad (3.5)$$

Notice that if $f \in C^1(\Omega; [0, 1])$, then

$$TV(f) = \int_{\Omega} |\nabla f(x)| dx. \quad (3.6)$$

In this case, the first term of the Total Variation is equal to the length of the boundary $\partial\Omega_i$. Energy (3.4) measures both the smoothness of the segmentation boundaries and the fidelity of the approximating piecewise patch function \mathbf{p} to the manifold of patches \mathcal{P}_u of the input image u . The output of the method is a partition of the image into regions with homogeneous texture together with a patch representative of the texture of each region.

The variational model (3.4) is defined using characteristic functions $\chi_{\Omega_i} \in BV(\Omega; \{0, 1\})$, and the constraint $\sum_i \chi_i(x) = 1$ for each pixel $x \in \Omega$ implies that each pixel only belongs to a unique region Ω_i . But the set $BV(\Omega; \{0, 1\})$ is not convex and, moreover, the Euler-Lagrange equation for non-continuous functions leads to difficulties in numerical implementations. Thus, following the idea proposed by Mory and Ardon (2007) and Li et al. (2016) we relax the characteristic functions to be fuzzy membership functions, which are associated to the notion of fuzzy sets¹, firstly introduced by Zadeh (1965). Fuzzy

¹A fuzzy set is a pair (X, m) , where X is a set and $m : X \rightarrow [0, 1]$ a membership function. The value $m(x)$, $x \in X$, is called the grade of membership of x in (X, m) .

membership functions belong to the set

$$\mathcal{C} = \left\{ (\omega_1, \dots, \omega_N) \mid \omega_i \in BV(\Omega; [0, 1]), 0 \leq \omega_i(x) \leq 1, \sum_{i=1}^N \omega_i(x) = 1, \forall x \in \Omega \right\}. \quad (3.7)$$

Hence, $\omega_i(x)$ describes the fuzzy membership of a pixel x that may well belong simultaneously to more than one region; in other words, $\omega_i(x)$ can be understood as the probability that x belongs to the region Ω_i . Let $\boldsymbol{\omega}$ be $\boldsymbol{\omega} = (\omega_1, \dots, \omega_N)$ which is often denoted as an N -phase fuzzy membership function. In this framework, our model (3.4) writes as

$$\begin{aligned} \min_{(\mathbf{p}, \boldsymbol{\omega}) \in L^1(\Omega; L^1(\Delta_t)) \times \mathcal{C}} \bar{E}(\mathbf{p}, \boldsymbol{\omega}) &= \underbrace{\sum_{i=1}^N \int_{\Omega} |\nabla \omega_i(x)| dx}_{E_s(\boldsymbol{\omega})} \\ &+ \lambda \underbrace{\sum_{i=1}^N \int_{\Omega} \mathcal{D}_t^{\mathbf{a}, 1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}) \omega_i(x) dx}_{E_d(\mathbf{p}, \boldsymbol{\omega})}. \end{aligned} \quad (3.8)$$

The energy formulation (3.8) is convex with respect to \mathbf{p} and $\boldsymbol{\omega}$ separately but not jointly.

3.1 Optimization Algorithm

To minimize the functional (3.8), we introduce an auxiliary variable $\mathbf{v} = (v_1, \dots, v_N) \in \mathcal{C}$ representing the fuzzy membership function $\boldsymbol{\omega}$ and we penalize its deviation from $\boldsymbol{\omega}$ by a quadratic term as follows

$$\begin{aligned} \min \tilde{E}(\mathbf{p}, \boldsymbol{\omega}, \mathbf{v}) &= E_s(\boldsymbol{\omega}) + \lambda E_d(\mathbf{p}, \mathbf{v}) \\ &+ \underbrace{\frac{1}{2\theta} \sum_{i=1}^N \int_{\Omega} (\omega_i(x) - v_i(x))^2 dx}_{E_c(\boldsymbol{\omega}, \mathbf{v})}, \end{aligned} \quad (3.9)$$

where $\theta > 0$ is small enough to enforce \mathbf{v} to be the closest possible to $\boldsymbol{\omega}$. This energy can be minimized by alternatively fixing two variables and minimizing with respect to the third one since the functional \tilde{E} is convex w.r.t each variable, and iterate until convergence. The optimization scheme can be summarized as follows:

$$\begin{aligned}\mathbf{v}^{k+1} &= \arg \min_{\mathbf{v}} E(\boldsymbol{\omega}^k, \mathbf{p}^k, \mathbf{v}) \\ \boldsymbol{\omega}^{k+1} &= \arg \min_{\boldsymbol{\omega}} E(\boldsymbol{\omega}, \mathbf{p}^k, \mathbf{v}^{k+1}) \\ \mathbf{p}^{k+1} &= \arg \min_{\mathbf{p}} E(\boldsymbol{\omega}^{k+1}, \mathbf{p}, \mathbf{v}^{k+1})\end{aligned}\quad (3.10)$$

In the following we describe the optimization algorithm we use to minimize each of them.

3.1.1 v-subproblem

The subproblem for $\mathbf{v} = (v_1, \dots, v_N)$ is

$$\min_{\mathbf{v} \in \mathcal{C}} \left(\lambda E_d(\mathbf{p}, \mathbf{v}) + \frac{1}{2\theta} E_c(\boldsymbol{\omega}, \mathbf{v}) \right). \quad (3.11)$$

Observe that this problem is separable with respect to the variables v_i . Actually, we can obtain a closed formula of the solution of (3.11) together with a projection onto the convex set \mathcal{C} :

$$v_i(x) = \omega_i(x) - \lambda\theta \mathcal{D}_t^{\mathbf{a},1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}(x)), \quad \forall i. \quad (3.12)$$

To include the projection onto \mathcal{C} , the set of fuzzy membership functions, the expression (3.12) is replaced by

$$v_i(x) = \min\{\max\{\omega_i(x) - \lambda\theta \mathcal{D}_t^{\mathbf{a},1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}(x)), 0\}, 1\} \quad (3.13)$$

with a final normalization step $\sum_{i=1}^N v_i(x) = 1, \forall x \in \Omega$.

3.1.2 ω -subproblem: dual formulation algorithm

The subproblem for ω is

$$\min_{\omega} \left(E_s(\omega) + \frac{1}{2\theta} E_c(\omega, \mathbf{v}) \right). \quad (3.14)$$

As the problem (3.14) is separable in the variables ω_i , we can solve each problem independently, that is:

$$\min_{\omega_i} \int_{\Omega} |\nabla \omega_i(x)| dx + \frac{1}{2\theta} \int_{\Omega} (\omega_i(x) - v_i(x))^2 dx. \quad (3.15)$$

This minimization is done using a dual formulation and the algorithm proposed by Chambolle (2004):

Proposition 1. *The solution of Eq. (3.15) is given by*

$$\omega_i(x) = v_i(x) + \theta \operatorname{div}(\xi(x)) \quad (3.16)$$

where the vector function ξ is obtained by the following iterative fixed-point scheme:

$$\xi^{n+1}(x) = \frac{\xi^n(x) + \tau \nabla (\theta \operatorname{div}(\xi(x)) + v_i(x))}{1 + \tau |\nabla (\theta \operatorname{div}(\xi(x)) + v_i(x))|} \quad (3.17)$$

taking $\xi^0 = 0$ and $\tau \leq 1/8$.

3.1.3 \mathbf{p} -subproblem

The subproblem for \mathbf{p} is

$$\min_{\mathbf{p}} E_d(\mathbf{p}, \mathbf{v}) = \sum_{i=1}^N \int_{\Omega} \mathcal{D}_t^{\mathbf{a},1}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}(x)) v_i(x) dx. \quad (3.18)$$

with $\mathbf{p} = \sum_i \mathbf{p}_{\Omega_i} \chi_{\Omega_i}$. For each region Ω_i , the unknown patch \mathbf{p}_{Ω_i} is given by a vector of size the number of pixels contained in the disc Δ_t . Thus, the solution for subproblem (3.18) is given by a weighted median vector. To compute the weighted median vector we extend the proposal of Barni et al. (1993) and Barni (1997) that suggest a fast algorithm to compute the median vector filter. Their approach consists in

1. Componentwise apply the scalar median filter: let \mathbf{z}_m be the output of such operation.
2. For each vector \mathbf{z}_i calculate $d_i = f(\mathbf{z}_i) - f(\mathbf{z}_m)$, where $f(\mathbf{z}) = \sum_{i=1}^N \|\mathbf{z} - \mathbf{z}_i\|_{L^1}$.
3. The median vector is the point that minimizes d_i .

In Appendix A we explain in more detail the derivation of the fast algorithm for the median vector filter proposed by Barni et al. (1993) and Barni (1997).

3.2 Initialization

As the functional (3.9) is not jointly convex the final result has a high dependence on the initialization. Following the idea of Li et al. (2016) we initialize our algorithm using the fuzzy c-means algorithm proposed by Bezdek et al. (1993) which, applied over the set of patches of the input image, turns into:

$$\min J(\omega, \mathbf{p}_{\Omega_i}; \mathbf{p}_u(x)) = \min \sum_{k=1}^{\#px} \sum_{i=1}^N \omega_{i,k} \mathcal{D}^{a,2}(\mathbf{p}_u(x), \mathbf{p}_{\Omega_i}). \quad (3.19)$$

Its minimum is computed as:

$$\begin{aligned} \omega_{\Omega_i}(x_k) &= \sum_{j=1}^N \frac{\mathcal{D}_t^{a,2}(\mathbf{p}_{\Omega_i}, \mathbf{p}_u(x_k))}{\mathcal{D}_t^{a,2}(\mathbf{p}_{\Omega_j}, \mathbf{p}_u(x_k))} \quad \forall i, k \\ \mathbf{p}_{\Omega_i}(x) &= \frac{\sum_k (\omega_{\Omega_i})^2 \mathbf{p}_u(x_k)}{\sum_{k=1}^N (\omega_{\Omega_i}(x_k))^2} \quad \forall i. \end{aligned} \quad (3.20)$$

Afterwards, by applying (3.18), the median patch that corresponds to each initial region will be computed. Figure 3.1 displays the initialization for the left image of Figure 2.1.



Figure 3.1: Fuzzy c-means initialization.

3.3 Final segmented image and discs

To solve our problem (3.4) we relaxed it by using fuzzy membership functions (3.8). Then, a segmentation is provided by:

$$I(x) = \sum_i \bar{p}_{\Omega_i} \omega_{\Omega_i}(x). \quad (3.21)$$

where $\omega_i(x) \in [0, 1]$ and \bar{p} is the mean of the patch associated to the region Ω_i . Therefore, for each pixel x we are considering the probability of belonging to the region Ω_i . In order to impose that each pixel belongs to a unique region we propose to select, for each pixel, the maximum value of the membership functions at that pixel.

$$\chi_{\Omega_i}(x) = \begin{cases} 1, & \text{if } i = \arg \max_j \omega_j(x) \\ 0, & \text{else} \end{cases}, \quad (3.22)$$

which is a translation of the assumption that each pixel belongs only to the region with highest membership value. Therefore, the final segmentation is provided by

$$I(x) = \sum_i \bar{p}_{\Omega_i} \chi_{\Omega_i}(x) \quad (3.23)$$

where, again, \bar{p} is the mean of the patch associated to the region Ω_i . We also provide a disc that contains the texture associated to each region. Figure 3.2 shows an example of the output using fuzzy membership function (Fig 3.2a), recovering the characteristic functions and the discs (Fig 3.2b).



(a) Fuzzy membership functions and corresponding output.



(b) Characteristic functions, corresponding output and representative discs.

Figure 3.2: Output of the fuzzy membership functions and the extracted characteristic functions, together with the discs associated to each region.

4

Experimental Results

In this Chapter we present some results on the Berkeley and Weizmann datasets. We also present a study of the parameters related to the patches and discs in order to fix them.

In Chapter 3 we have presented a segmentation model that works by comparing adaptive patches. The model depends on the following four free parameters:

- Balance among data and regularity term λ , we fix it to 0.04 for all images.
- Radius used in the patch computation r , which is related to the final size of the patch.
- Length, $2t$, of each side of the square where the discs are embedded. Let us recall that each computed patch is interpolated to a disc of fixed size in order to be able to compare the different patches. Also, the parameter t let us decide the resolution of the disc Δ_t . With an abuse of notation, in this Chapter we will denote by t the diameter of the disc instead of its radius.
- Number of regions n : this parameter is left free for every image. Anyway, the Tables presented in this section have been obtained with a fixed value for all the images of the dataset.

Let us comment on the radius r and the diameter of the discs t . The radius of the patch strongly depends on the local content of the image. We use a larger or smaller value of r depending on the blur, noise or the kind of texture. Actually, equation (2.2) already gives an intuition of it, as we are considering a balance among the distance of the pixels to the center of the patch, and the gradient of the central pixel. Therefore, for low gradient values the ellipse automatically grows far until it reaches the size of the radius, while for large gradients the ellipse can not grow so far. In Figure 4.1 we show the behavior of the patches at textured and homogeneous regions, for a fixed value of r . We can observe as, for a fixed value of r , the ellipses automatically and intrinsically adapt their shape and size to the local content of the image. For instance, in the textured regions of the image the ellipses are smaller than in the homogeneous regions. Moreover, we also see that for small values of r the ellipses are too small in the highly textured regions, while for bigger values of r the ellipses at the homogeneous regions grow a lot. The parameter r also affects the content of the discs. For low values of r we obtain blurred discs, as we are using limited values to do the interpolation, while the discs obtained with larger values of r the texture is sharper.

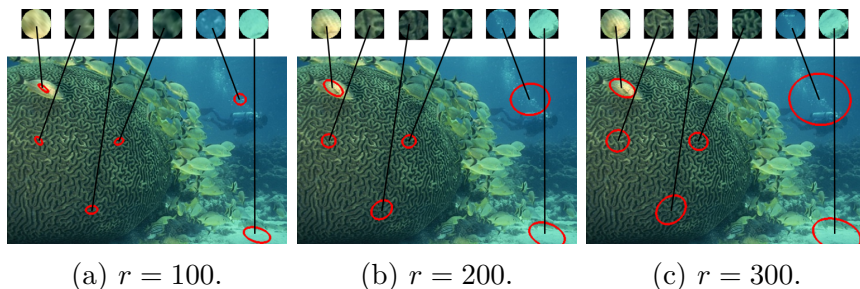


Figure 4.1: Radius (r) comparison of the patches and discs.

In order to fix the size of the disc, we have compared the results of 100 images, randomly selected, from the Berkeley dataset using different values of t ($t = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 51\}$) and r

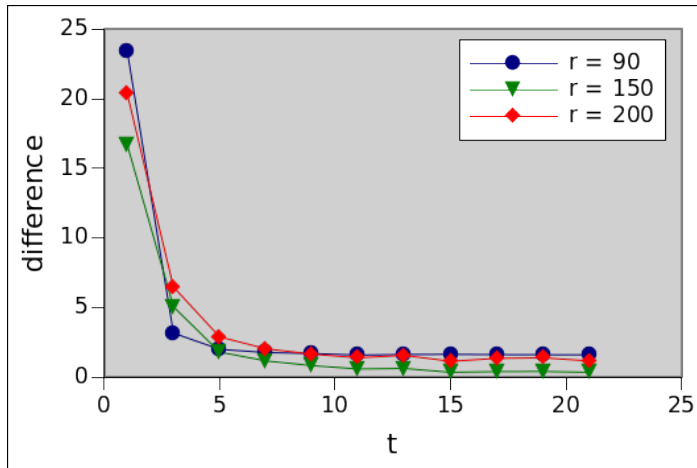


Figure 4.2: Segmentation difference (larger values mean more different) between results obtained with the discs of size $t = 51$ (more resolution) and the discs of the specified t . We also evaluated different values of r .

($r = \{90, 150, 200\}$). The comparison is performed by labeling in the same way all the images and computing a simple error difference. If I and \tilde{I} denote two of such segmentations results, using the same segmentation labels, for each pixel we consider:

$$d(I(x), \tilde{I}(x)) = \begin{cases} 1, & \text{if } I(x) \neq \tilde{I}(x) \\ 0, & \text{if } I(x) = \tilde{I}(x), \end{cases} \quad (4.1)$$

then, we sum over all the pixels and normalize:

$$D(I, \tilde{I}) = 100 \cdot \frac{\sum_{x=1}^N d(I(x), \tilde{I}(x))}{N}, \quad (4.2)$$

where N is the number of pixels of the image. That is, we compute the percentage of pixels that differ on each pair of images.

Figure 4.2 shows the quantitative results of the comparison. Each cell of the table contains the mean of the difference between the segmentation results obtained with discs of size $t = 51$ and the size t

specified at each row. The radius r used is specified in each column. We can observe that the difference among the results using the different values of t is very small, specially for t bigger than 7, which is a usual minimum size for a patch. This is closely related to the fact that the image content in the interpolated discs is similar, no matter the size of the disc. This fact is illustrated in Figure 4.3, where we show four discs of the image presented in Figure 4.1. We can see that for a fixed value of r , and changing the value of t , the results appear the same but at a different scale. The interpolation only changes when we use a different value of r . Finally, notice the bad results for $t = 1$, where the patch is degenerated to a point.

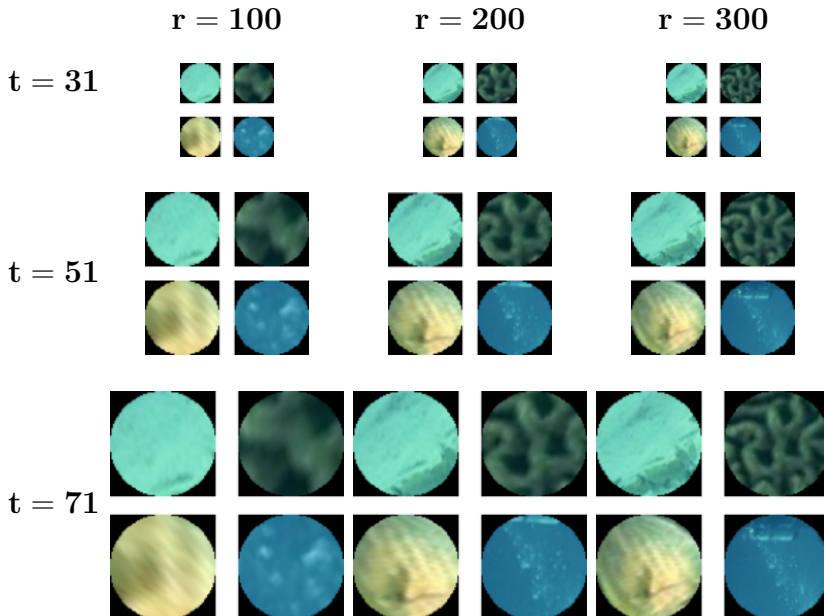


Figure 4.3: Interpolation of the patches into discs for different radius $r(100, 200, 300)$ and size of the disc t (31, 51, 71).

We tested our method on the Berkeley dataset proposed by Martin et al. (2001) and the Weizmann dataset proposed by Alpert et al. (2007).

In Tables 4.2, 4.3 and 4.4 we show a comparison of our method with two methods that are also based on fuzzy membership functions: the implementation of the piecewise constant Mumford-Shah method proposed in Li et al. (2010) and the method proposed by Li et al. (2016).

In order to quantitatively compare our method with respect to the works mentioned we have used the F-measure, first defined in Rijsbergen (1979), which is the harmonic mean of the precision (P) and recall (R):

$$F = 2 \cdot \frac{P \cdot R}{P + R}. \quad (4.3)$$

To explain the meaning and computation of these two values, let us consider an image with two regions Ω_1 and Ω_2 of Ω . In Table 4.1 we summarize the four possible options of membership of a pixel x with respect to the classification obtained and the correct one.

		Predicted	
		$x \in \Omega_1$	$x \in \Omega_2$
Actual	$x \in \Omega_1$	✓(TP)	✗(FN)
	$x \in \Omega_2$	✗(FP)	✓(TN)

Table 4.1: Summary of the four possible classifications of a pixel.

Then, precision (P) and recall (R) are measured as

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad (4.4)$$

that is, precision (also called predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. In terms of segmentation results, precision is the proportion of correctly positive

labeled pixels, while recall is the fraction of pixels correctly labeled, with respect to the ground-truth. In the case that the image is segmented in more than two regions we compute the mean, over all the regions, of the precision and recall. From these values we get the F-measure error of the image.

In Table 4.2 we present quantitative results on images of the Weizmann Segmentation dataset (Alpert et al. (2007)) and Berkeley dataset (Martin et al. (2001)). These results are computed with discs of size $t = 17$ and the radius r is either 100, 150, 200 or 250. The output discs, which contain the texture of the regions, have size $t = 51$. Since Weizmann dataset is mainly made of images of one or two disconnected objects in front of a background, we have fixed $n = 2$ for all the images. As Berkeley dataset also contains images with a small number of objects in front of a background, we tried experiments with three and six regions, but as it can be observed in Figure 4.4 when we fix the number of regions to six we start to obtain an oversegmentation or decomposition in the so-called superpixels instead of the expected segmentation, which is not in the scope of this work. This effect is due to the fact that our segmentation regions can be made of several connected components. For this reason we show quantitative and qualitative results on images segmented with three regions.



Figure 4.4: Some results on Berkeley dataset using 6 regions.

We can observe in Table 4.2 that the behaviour is similar for all

the datasets, but when we observe some qualitative results (presented in Tables 4.3 and 4.4 for the Weizmann and Berkeley datasets, respectively) we can observe that, thanks to the use of patches, our results better capture the texture of the objects, producing results more robust to the illumination change and the heterogeneity of the objects. Moreover, the boundaries of the regions are smoother and it produces less region outliers, compared to the other two methods. Let us finally recall that we are not using texture descriptors, but an $L1$ -norm difference of the color within the patches, therefore these results are promising and could be improved by using a data term that takes into account texture descriptors instead of the color values of the disc.

Quantitative study	Alpert et al. (2007)			Martin et al. (2001)		
	R	P	F	R	P	F
Li et al. (2016)	0.5158	0.3601	0.3630	0.3624	0.4621	0.4062
Li et al. (2010)	0.5271	0.3627	0.3657	0.3864	0.4935	0.4334
Ours ($r=100$)	0.5292	0.3655	0.3677	0.3829	0.4869	0.4287
Ours ($r=150$)	0.5157	0.3744	0.3661	0.3849	0.4878	0.4303
Ours ($r=200$)	0.5014	0.3766	0.3582	0.3867	0.4900	0.4323
Ours ($r=250$)	0.4900	0.3868	0.3553	0.3747	0.4619	0.4135

Table 4.2: Results in Weizmann and Berkeley datasets. Recall (R), Precision (P) and F-measures (F) for our method, with different values of r , and the methods proposed in Li et al. (2010) and Li et al. (2016).














































	Input	Li et al. (2016)	Li et al. (2010)	Ours	Discs
Image 1					
		0.4131	0.4203	0.4438	
Image 2					
		0.3900	0.3924	0.4340	
Image 3					
		0.7178	0.7462	0.7537	
Image 4					
		0.7584	0.7647	0.7784	
Image 5					
		0.4726	0.4706	0.4944	
Image 6					
		0.3408	0.3470	0.3801	
Image 7					
		0.5037	0.5558	0.5875	
Image 8					
		0.3809	0.3822	0.3922	
Image 9					
		0.2738	0.2813	0.2911	

Table 4.3: Some results on the Weizmann dataset (Alpert et al. (2007)) of our method compared with Li et al. (2016) and Li et al. (2010) Below each picture we also provide its F-measure.














































	Input	Li et al. (2016)	Li et al. (2010)	Ours	Discs
Image 1					
		0.3528	0.3852	0.4457	
Image 2					
		0.3513	0.3656	0.3915	
Image 3					
		0.7350	0.7415	0.7439	
Image 4					
		0.3859	0.3968	0.4848	
Image 5					
		0.7255	0.7284	0.7534	
Image 6					
		0.4945	0.4963	0.5325	
Image 7					
		0.2557	0.5616	0.5853	
Image 8					
		0.4236	0.5579	0.5661	
Image 9					
		0.0.9553	0.9823	0.9881	

Table 4.4: Some results on the Berkeley dataset (Martin et al. (2001)) of our method compared with Li et al. (2016) and Li et al. (2010) Below each picture we also provide its F-measure.

5

Conclusions and Future Work

We propose a new variational formulation for image segmentation that uses similarity among shape and size adaptive patches in an L^1 fidelity term and the total variation of fuzzy membership functions as relaxed length of the boundaries of the segmentation regions. The result is a partition of the image in regions of local homogeneous texture regardless of differences in the point of view or suffered local perspective or affine distortion, together with a patch, associated to each region, which contains the representative texture of its corresponding region.

Despite providing a patch with the representative texture our method does not use pure texture features, thus it does not deal well with images that only contain texture, without color contrast, therefore as a future work we propose to compare texture features extracted from the patches in order to segment the images. As it has been shown in Chapter 4, the radius has a strong dependency on how textured is the image. Therefore another improvement of the method would be to automatically compute the appropriate radius size for each image.

Moreover, the proposed method, built on Riemannian metrics intrinsic to the input image and on an L^1 data term, could be used to synthesize a textured output by using the representative tensor metric of each region to fill-in the region with the texture of the representative patch. This is an interesting direction for future research.

A

Fast Weighted Median Vector Algorithm

In this Appendix we present the derivation of the Fast Algorithm for the Median Vector Filter proposed by Barni et al. (1993) and Barni (1997). We also show their direct extension to Weighted Vector Median and propose an example to illustrate the main steps.

Let us start by the definition of median vector attributed to Astola et al. (1990):

Definition 1 (Median Vector (Astola et al. (1990))). Given N vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ in \mathbb{R}^p , the vector median is \mathbf{x}_{vm} such that:

$$\mathbf{x}_{\text{vm}} \in \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \quad (\text{A.1})$$

and for all $j = 1, \dots, N$:

$$\sum_{i=1}^N \|\mathbf{x}_{\text{vm}} - \mathbf{x}_i\|_1 \leq \sum_{i=1}^N \|\mathbf{x}_j - \mathbf{x}_i\|_1, \quad j = 1, \dots, N. \quad (\text{A.2})$$

Viero et al. (1994) extended this concept to the case where each given vector has a different weight:

Definition 2 (Weighted Median Vector Filter (Viero et al. (1994))). Let $\mathbf{x}_1, \dots, \mathbf{x}_N$ be vectors and let $\alpha_1, \dots, \alpha_N$ be their corresponding

nonnegative weights, such that $\sum_{i=1}^N \alpha_i = 1$. The weighted vector median is the vector \mathbf{x}_{wvm} such that

$$\mathbf{x}_{\text{wvm}} \in \{\mathbf{x}_i, i = 1, \dots, N\} \quad (\text{A.3})$$

and for all $j = 1, \dots, N$

$$\sum_{i=1}^N \alpha_i \|\mathbf{x}_{\text{wvm}} - \mathbf{x}_i\|_1 \leq \sum_{i=1}^N \alpha_i \|\mathbf{x}_j - \mathbf{x}_i\|_1 \quad (\text{A.4})$$

Barni et al. (1993) presented a fast algorithm to compute the vector median and we propose to extend it to the weighted case. Their intuition relies on the fact that both, the scalar vector median¹, and the median vector minimize the same cost function:

$$f(\mathbf{x}) = \sum_{i=1}^N \|\mathbf{x} - \mathbf{x}_i\|_1 \quad (\text{A.5})$$

where N is the number of vectors of the set. The only difference between both solutions is that the scalar vector median is the unconstrained minimum of (A.5), while the vector median is a constrained minimum: we impose the solution to belong to the initial set of vectors. Consequently, we can compute the unconstrained minimum of energy (A.5), which we denote by \mathbf{x}_m and, afterwards, search the vector from our set which minimizes the following distance:

$$d_i = f(\mathbf{x}_i) - f(\mathbf{x}_m), \quad \forall i. \quad (\text{A.6})$$

Function $f(\mathbf{x})$ is continuous and piecewise linear since the partial derivative with respect to the components of the vector x are piecewise constant, except in a set of measure zero. Thereby, $d_i = f(\mathbf{x}_i) - f(\mathbf{x}_m)$ can be computed integrating the gradient of $f(\mathbf{x})$ along the path that joints \mathbf{x}_i to \mathbf{x}_m , which is made of segments parallel to

¹The scalar vector median consists in applying the scalar median to each component of the vector.

the coordinate axes. This can be fastly computed by exploding the histograms of the vector components.

Barni et al. (1993) and Barni (1997) algorithm relies on the fact that the scalar vector median is easy and fast to compute. Let us illustrate the algorithm with one example, which we introduce progressively in order to illustrate each step of the algorithm.

Example: Computation of \mathbf{x}_m .

Let us consider the following three vectors in \mathbb{N}^3 with their associated weights α_i :

$$\left\{ \begin{array}{ll} x_1 = (2, 5, 3) & \alpha_1 = 0.3 \\ x_2 = (1, 3, 0) & \alpha_2 = 0.4 \\ x_3 = (1, 2, 2) & \alpha_3 = 0.3 \end{array} \right. \quad (\text{A.7})$$

Using the weights, we obtain the following channel-histograms, from which we get the scalar vector median filter x_m :

$$\left. \begin{array}{l} h_1 = (0, 0.7, 0.3, 0, 0, 0) \\ h_2 = (0, 0, 0.3, 0.4, 0, 0.3) \\ h_3 = (0.4, 0, 0.3, 0.3, 0, 0) \end{array} \right\} \rightarrow \mathbf{x}_m = (1, 3, 2) \quad (\text{A.8})$$

Once the scalar vector median is computed, we calculate the integral of the gradient along the path that joints the scalar median vector \mathbf{x}_m to every vector \mathbf{x}_i from our set. This path is always formed by p segments parallel to the axes, where p is the dimension of the vector space. Let $\{\mathbf{u}_0 = \mathbf{x}_m, \mathbf{u}_1, \dots, \mathbf{u}_{p-1}, \mathbf{u}_p = \mathbf{x}_i\}$ be the extreme points of such segments.

Example: (Cont.) Path from \mathbf{x}_m to \mathbf{x}_1 .

In our example, if we consider the path going from \mathbf{x}_m to \mathbf{x}_1 we have to consider the following extreme segments:

$$\{\mathbf{u}_0 = (1, 3, 2), \mathbf{u}_1 = (2, 3, 2), \mathbf{u}_2 = (2, 5, 2), \mathbf{u}_3 = (2, 5, 3)\} \quad (\text{A.9})$$

Now, we can compute the distance as

$$d_i = f(\mathbf{x}_i) - f(\mathbf{x}_m) = \int_{\mathbf{u}_0}^{\mathbf{u}_1} \nabla f(\mathbf{x}) d\mathbf{x} + \cdots + \int_{\mathbf{u}_{p-1}}^{\mathbf{u}_p} \nabla f(\mathbf{x}) d\mathbf{x} \quad (\text{A.10})$$

Example: (Cont.) Splitting of the distance into integrals.

In our particular case:

$$\begin{aligned} d_1 = f(\mathbf{x}_1) - f(\mathbf{x}_m) &= \int_{\mathbf{u}_0}^{\mathbf{u}_1} \nabla f(\mathbf{x}) d\mathbf{x} + \int_{\mathbf{u}_1}^{\mathbf{u}_2} \nabla f(\mathbf{x}) d\mathbf{x} \\ &+ \int_{\mathbf{u}_2}^{\mathbf{u}_3} \nabla f(\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (\text{A.11})$$

Let us now consider two different situations: $x_{i,j} > x_{m,j}$ and $x_{i,j} < x_{m,j}$. The equality case as gives 0. We start considering the situation where $x_{i,j} > x_{m,j}$. In this case we can rewrite each integral of (A.10) as:

$$\int_{\mathbf{u}_{j-1}}^{\mathbf{u}_j} \nabla f(\mathbf{x}) d\mathbf{x} = \sum_{k=1}^{\Delta_{i,j}} \int_{\mathbf{u}_{j-1}^{k-1}}^{\mathbf{u}_{j-1}^k} \nabla f(\mathbf{x}) d\mathbf{x}, \quad (\text{A.12})$$

where $\mathbf{u}_j^k = (x_{i,1}, x_{i,2}, \dots, x_{i,j}, x_{m,j+1} + k, x_{m,j+2}, \dots, x_{m,p})$ and $\Delta_{i,j} = x_{i,j} - x_{m,j}$. Let us comment more on equation (A.12): Observe that the integral that goes from \mathbf{u}_{j-1} to \mathbf{u}_j only changes one of the coordinates of the vector, therefore we are splitting the integral into unit values and summing the gradient on that single pixel. About \mathbf{u}_j^k observe that it is gradually moving the coordinate j from \mathbf{x}_i to \mathbf{x}_m .

Example: (Cont.) Computation of each integral.

$$\int_{\mathbf{u}_0}^{\mathbf{u}_1} \nabla f(\mathbf{x}) d\mathbf{x} = \sum_{k=1}^1 \int_{\mathbf{u}_0^{k-1}}^{\mathbf{u}_0^k} \nabla f(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{u}_0^0}^{\mathbf{u}_0^1} \nabla f(\mathbf{x}) d\mathbf{x} \quad (\text{A.13})$$

$$\begin{aligned} \int_{\mathbf{u}_1}^{\mathbf{u}_2} \nabla f(\mathbf{x}) d\mathbf{x} &= \sum_{k=1}^2 \int_{\mathbf{u}_1^{k-1}}^{\mathbf{u}_1^k} \nabla f(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{u}_1^0}^{\mathbf{u}_1^1} \nabla f(\mathbf{x}) d\mathbf{x} \\ &\quad + \int_{\mathbf{u}_1^1}^{\mathbf{u}_1^2} \nabla f(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (\text{A.14})$$

$$\int_{\mathbf{u}_2}^{\mathbf{u}_3} \nabla f(\mathbf{x}) d\mathbf{x} = \sum_{k=1}^1 \int_{\mathbf{u}_2^{k-1}}^{\mathbf{u}_2^k} \nabla f(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{u}_2^0}^{\mathbf{u}_2^1} \nabla f(\mathbf{x}) d\mathbf{x} \quad (\text{A.15})$$

where $\mathbf{u}_0^0 = (1, 3, 2)$, $\mathbf{u}_0^1 = (2, 3, 2)$, $\mathbf{u}_1^0 = (2, 3, 2)$, $\mathbf{u}_1^1 = (2, 4, 2)$, $\mathbf{u}_1^2 = (2, 5, 2)$, $\mathbf{u}_2^1 = (2, 5, 2)$, $\mathbf{u}_2^2 = (2, 5, 3)$.

We will now only focus on one of the integrals from equation (A.12). Let us firstly observe that, as we are working at pixel level, the gradient is constant, moreover if h_j denotes the histogram associated to channel j , and, particularly, $h_{j,r}$ denotes the number of samples such that $\mathbf{x}_{i,j} = r, \forall i$ and the samples are quantized in b bins levels, then:

$$d_{j,k} := \int_{\mathbf{u}_j^{k-1}}^{\mathbf{u}_j^k} \nabla f(\mathbf{x}) d\mathbf{x} = \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} - \sum_{r=x_{m,j}+k}^{b-1} h_{j,r} \quad (\text{A.16})$$

In order to compute (A.10) efficiently, Barni (1997) proposes to substitute (A.16) into (A.12):

$$\begin{aligned} &\sum_{k=1}^{\Delta_{i,j}} \left(\sum_{r=0}^{x_{m,j}+k-1} h_{j,r} - \sum_{r=x_{m,j}+k}^{b-1} h_{j,r} \right) = \\ &\sum_{k=1}^{\Delta_{i,j}} \left(\sum_{r=0}^{x_{m,j}+k-1} h_{j,r} - \left(\sum_{r=0}^{b-1} h_{j,r} - \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} \right) \right) = \\ &2 \sum_{k=1}^{\Delta_{i,j}} \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} - \Delta_{i,j} |h_j|, \end{aligned} \quad (\text{A.17})$$

where $|h_j|$ denotes the area of the histogram associated to channel j . Let us now analyze in more detail the part which contains the sums:

$$\begin{aligned}
& \sum_{k=1}^{\Delta_{i,j}} \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} = \\
& \sum_{r=0}^{x_{m,j}} h_{j,r} + \sum_{r=0}^{x_{m,j}+1} h_{j,r} + \cdots + \sum_{r=0}^{x_{m,j}+\Delta_{i,j}-1} h_{j,r} = \\
& \Delta_{i,j} \sum_{r=0}^{x_{m,j}} h_{j,r} + (\Delta_{i,j} - 1)h_{j,x_{m,j}+1} + \cdots + h_{j,x_{m,j}+\Delta_{i,j}-1}.
\end{aligned} \tag{A.18}$$

Therefore, we can rewrite (A.17) as:

$$\begin{aligned}
& 2 \left(\Delta_{i,j} \sum_{r=0}^{x_{m,j}} h_{j,r} + (\Delta_{i,j} - 1)h_{j,x_{m,j}+1} + \cdots + h_{j,x_{m,j}+\Delta_{i,j}-1} \right) - \Delta_{i,j}|h_j| = \\
& \Delta_{i,j} \left(2 \sum_{r=0}^{x_{m,j}} h_{j,r} - |h_j| \right) + 2 \left((\Delta_{i,j} - 1)h_{j,x_{m,j}+1} + \cdots + h_{j,x_{m,j}+\Delta_{i,j}-1} \right),
\end{aligned} \tag{A.19}$$

which gives us a recursive method to compute the distances.

Let us now analyze the case where $x_{i,j} < x_{m,j}$. Again, we start with an example.

Example: (Cont.) Path from \mathbf{x}_m to \mathbf{x}_2 , which involves $x_{i,j} < x_{m,j}$.

We will use $x_2 = (1, 3, 0)$ from the same example, therefore we obtain the following path:

$$\{\mathbf{u}_0 = (1, 3, 2), \mathbf{u}_1 = (1, 3, 2), \mathbf{u}_2 = (1, 3, 2), \mathbf{u}_3 = (1, 3, 0)\} \quad (\text{A.20})$$

and the distance is computed as:

$$d_2 = f(\mathbf{u}_3) - f(\mathbf{u}_2) = \int_{\mathbf{u}_2}^{\mathbf{u}_3} \nabla f(\mathbf{x}) d\mathbf{x}. \quad (\text{A.21})$$

as all the other integrals have the same integration limits and, therefore, equal to 0.

In this situation, each integral from (A.10) is written as:

$$\int_{\mathbf{u}_{j-1}}^{\mathbf{u}_j} \nabla f(\mathbf{x}) d\mathbf{x} = \sum_{k=-1}^{\Delta_{i,j}} \int_{\mathbf{u}_{j-1}^{k-1}}^{\mathbf{u}_{j-1}^k} \nabla f(\mathbf{x}), \quad (\text{A.22})$$

where $\mathbf{u}_j^k = (x_{i,1}, x_{i,2}, \dots, x_{i,j}, x_{m,j+1} + k, x_{m,j+2}, \dots, x_{m,p})$ and $\Delta_{i,j} = x_{i,j} - x_{m,j}$. Let us observe that $\Delta_{i,j}$ is negative, therefore in the summation, we are decreasing the indices.

Example: (Cont.) Computation of the integrals

In our example we have that:

$$\begin{aligned} \int_{\mathbf{u}_2}^{\mathbf{u}_3} \nabla f(\mathbf{x}) d\mathbf{x} &= \sum_{k=-1}^{-2} \int_{\mathbf{u}_2^{k+1}}^{\mathbf{u}_2^k} \nabla f(\mathbf{x}) d\mathbf{x} = \\ &\int_{\mathbf{u}_2^0}^{\mathbf{u}_2^{-1}} \nabla f(\mathbf{x}) d\mathbf{x} + \int_{\mathbf{u}_2^{-1}}^{\mathbf{u}_2^{-2}} \nabla f(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (\text{A.23})$$

where $\mathbf{u}_2^0 = (1, 3, 2)$, $\mathbf{u}_2^{-1} = (1, 3, 1)$, $\mathbf{u}_2^{-2} = (1, 3, 0)$.

If we now focus on each integral we obtain that

$$\begin{aligned}
d_{j,k} &:= \int_{\mathbf{u}_j^{k+1}}^{\mathbf{u}_j^k} \nabla f(\mathbf{x}) d\mathbf{x} = \\
&\quad - \int_{\mathbf{u}_j^k}^{\mathbf{u}_j^{k+1}} \nabla f(\mathbf{x}) d\mathbf{x} = \\
&\quad \sum_{r=0}^{x_{m,j}+k} h_{j,r} - \sum_{r=x_{m,j}+k+1}^{b-1} h_{j,r} = \\
&\quad \sum_{r=0}^{x_{m,j}+k} h_{j,r} - \left(|h_j| - \sum_{r=0}^{x_{m,j}+k} h_{j,r} \right) = \\
&\quad 2 \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} + 2h_{j,x_{m,j}+k} - |h_j|.
\end{aligned} \tag{A.24}$$

By substituting this last result into (A.22) we obtain:

$$\begin{aligned}
&\sum_{k=-1}^{\Delta_{i,j}} \left(2 \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} + 2h_{j,x_{m,j}+k} - |h_j| \right) = \\
&2\Delta_{i,j}h_{j,x_{m,j}+k} - \Delta_{i,j}|h_j| + 2 \sum_{k=-1}^{\Delta_{i,j}} \sum_{r=0}^{x_{m,j}+k-1} h_{j,r}
\end{aligned} \tag{A.25}$$

which results into:

$$\begin{aligned}
d_j &:= 2\Delta_{i,j}h_{j,x_{m,j}+k} - \Delta_{i,j}|h_j| + 2\Delta_{i,j} \sum_{r=0}^{x_{m,j}+k-1} h_{j,r} + \\
&2(-1 - \Delta_{i,j})h_{j,x_{m,j}-1} + \\
&2(-2 - \Delta_{i,j})h_{j,x_{m,j}-2} + \cdots + 2(-1)h_{j,x_{m,j}-\Delta_{i,j}}.
\end{aligned} \tag{A.26}$$

Let us observe that in this case we are subtracting the extra terms of the summation.

In algorithm 1 we summarize this procedure.

Algorithm 1: Fast weighted vector median filter

Input : Set of vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$, the number of vectors n , scalar median vector filter x_m (obtained using the weights), $\text{hist}[j]$ vector containing the histogram of the j -th component of the image and $\text{histSum}[j] = \sum_{r=0}^{x_m[j]} \text{hist}[j][r]$

Output: Weighted Median Vector \mathbf{x}_{wvm}

```

foreach  $p$  in  $\mathbf{x}$  do
  for  $j$  in  $p$  do
     $\text{dif}[j] = x[p[j]] - x_m[j]$ ;
    if  $\text{dif}[j] > 0$  then
       $\text{sum}[p] += \text{dif}[j] * (2 * \text{histSum}[j] - n)$ ;
       $k = x_m[j]$ ;
      for  $m = 1$  to  $\text{dif}[j]$  do
         $k = k + 1$ ;
         $\text{sum}[p] = \text{sum}[p] + 2 * w[j] * (\text{dif}[j] - m) * \text{hist}[j][m[j]]$ ;
      end
    else
       $\text{sum}[p] += \text{dif}[j] * (2 * \text{histSum}[j] - n - 2 * \text{hist}[j][x_m[j]])$ ;
       $k = x_m[j]$ ;
      for  $m = -1$  to  $\text{dif}[j]$  do
         $k = k - 1$ ;
         $\text{sum}[p] = \text{sum}[p] + 2 * \text{hist}[j][k] * (m - \text{dif}[j])$ ;
      end
    end
  end
   $\mathbf{x}_{\text{wvm}} = \arg \min_p \text{sum}$ 
end

```

PART II:
SCENE STRUCTURE
RECONSTRUCTION

The whole is something else than
the sum of its parts.

KURT KOFFKA

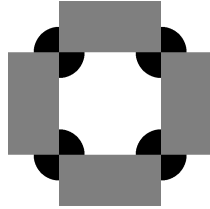
6

Introduction: Scene Structure

In this Chapter we describe the process of inferring the distal scene of a given proximal image. We also provide a psychophysical explanation and translate it contributing with a computational model to recover the most likely interpretation.

Visual completion is a pervasive process in our daily life that works by hallucinating contours and surfaces in the scene when there is not a physical magnitude for them. Whenever we look at an image, our brain unconsciously reconstructs the 3D scene by completing partially occluded objects while inferring their relative depth order into the scene. In Figure 6.1a, for instance, our brain prefers to interpret the scene as four discs partially occluded by four rectangles instead of, e.g., the more straightforward description of eight quarters of a disc and four rectangles fitting together. Also, in Figure 6.1b we perceive the branch in front of the arm of the bear.

Historically there have been two differentiated visual completion approaches: local and global completion. Local completion has been related to T-junctions and to the good continuation principle. When an object occludes another, the occluding and occluded boundaries form a configuration, called T-junction, which is the point where the visible part of the boundary of the occluded object terminates. T-junction configuration is shown in Figure 6.2a. Then, our visual system completes the occluded objects following the good continuation principle, that is, in such a way that the restored edges are



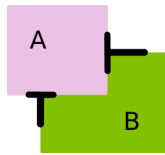
(a) Source: Kanizsa (1991)



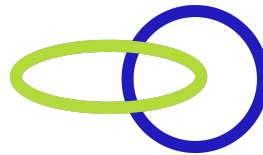
(b) Source: Martin et al. (2001)

Figure 6.1: Examples where our brain experiences visual completion.

the smoothest possible and the shapes as convex as possible. For instance, in figure 6.2b we observe an ellipse occluding a circle instead of any other configuration that would include sharper edges.



(a) T-junctions.



(b) Good continuation.

Figure 6.2: Local Completion Examples.

Global completion is driven by the simplicity principle which assumes that the visual system favors interpretations characterized by phenomenal simplicity, such as symmetry, repetition, regularity and familiarity or context properties. It typically leads towards the simplest completed shape, even though the good continuation principle may be violated, as shown by Koffka (1935), Kanizsa (1979) or Sekuler (1994). Figure 6.3a shows an example where two different completions occur depending on whether a global cue as symmetry is incorporated or only more local cues, while in Figure 6.3b, both interpretations coincide.

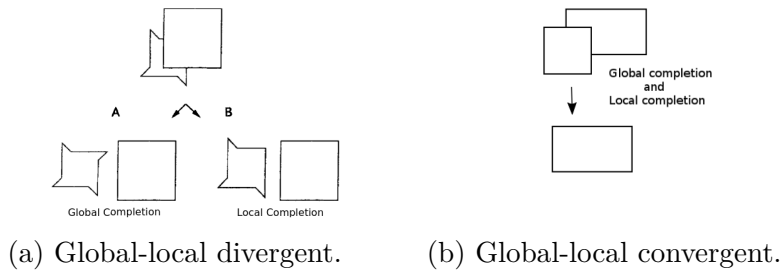


Figure 6.3: Examples of global-local completion processes. Images adapted from van Lier et al. (1995a).

Thanks to the studies of van Lier et al. (1995a,b) and Carrigan et al. (2015), it is acknowledged that occlusion patterns evoke both local and global completion processes and that the visual completion is the result of a competition between them. Moravec and Beck (1986) noticed that features favoring completion through good continuation are read out more quickly (in the very first second) than features favoring completion through symmetry, which are incorporated in the following nine seconds. The incorporation of different cues was also studied by Rubin (2001) who experimentally proved that local and global occlusion cues affect the perception of amodal completion at different stages of visual processing. Amodal completion occurs when portions of an object are hidden behind another object, but the former object is nevertheless perceived as a single continuous entity. Associated to the concept of amodal completion there is the modal completion, which occurs when portions of an object are camouflaged by an underlying surface, usually because this underlying surface happens to project the same luminance and color as the nearer object or background (Singh (2004)). For instance, in Fig. 6.4 we present an example for each type of completion. In Fig. 6.4a we perceive a circle in the middle of the half-moons which has the same color as the background, our mind induces illusory contours to be able to see the circle. On the other hand, in Fig. 6.4b our mind completes the

circles behind the square in an amodal way.

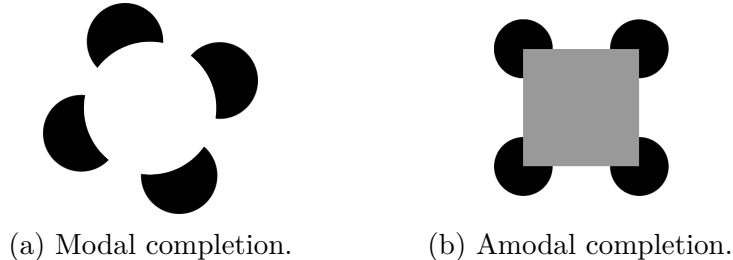


Figure 6.4: Modal and amodal completion of occluded regions. The modal completion induces illusory contours.

For the perception of amodal completion, Rubín proposed that the detection of local cues such as T-junctions generate a local pattern of activation which launches a process of propagation of the contour which is either enhanced or stopped depending on whether or not other global cues hold. As for global cues, the author focused in relatability and surface similarity, being cues that seem to be instantaneously used at first stages of occlusion perception.

Relatability was introduced by Kellman and Shipley (1991) as a necessary global condition for completion to occur: Two contours are said to be relatable if they can be connected with a smooth contour without inflection points. We illustrate this concept in Figure 6.5: in Figure 6.5a as there is the presence of two relatable contours, which are made of two pairs of relatable end-points (in the upper and lower part of the gray shape, respectively), we perceive an ellipse occluded by a rectangle; while in Figure 6.5b there is no pair of relatable contours so we perceive three different shapes.

In computer vision, a pioneering contribution to the recovery of image plane geometry was proposed by Nitzberg et al. (1993). The authors proposed a variational model for segmenting the image into objects which should be ordered according to their depth in the scene, providing the so-called 2.1 sketch. The minimization of their functional is able to find the occluding and the occluded objects, while

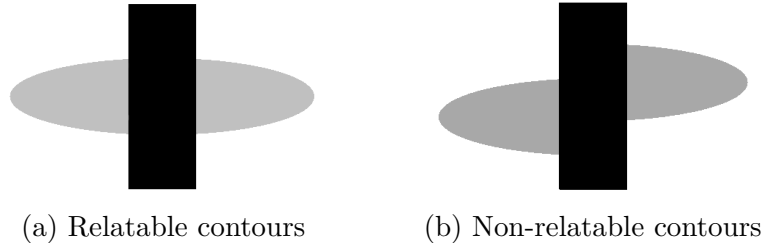


Figure 6.5: How many perceptual objects?

finding the occluded boundaries. Their energy functional is defined in terms of region simplification and completion of occluded contours, which is achieved by linking signatures of occlusion, such as T-junctions, with the Euler's elastica. In this way, the completion tends to respect the principle of good continuation. Despite its theoretical importance, the complexity of minimizing this energy makes the approach far from practical applications.

We are also interested in computationally modeling this perceptual phenomenon, recovering what the brain infers about the structure and the relative depth of the objects composing the scene from a planar image. To simplify the analysis of our approach, we focus on scenes where objects appear at two different depths, ones occluding the others. The current approach can handle scenes with both partially occluded and fully visible objects. Our contribution is twofold:

1. We propose a computational method relying on perceptual findings related to amodal visual completion to compute the disoccluded objects that form the possible 3D interpretations or configurations that arise from a planar image.
2. We propose a Bayesian probabilistic model which chooses, between the possible interpretations of a planar image, the most plausible one, justifying the visual completion human experience.

6.1 Disocclusion

In computer vision, the computational translation of the visual completion phenomenon is commonly referred to as disocclusion or inpainting. More precisely, inpainting refers to the recovery of the image in a hole or region where the data is missing or corrupted, so that the reconstructed image looks natural. The corrupted region is usually referred as the *inpainting mask*.

Most available methods for inpainting can be divided into two groups: texture-oriented methods, which use the self-similarity of the images to look for potential similarities to fill the hole (Demanez et al. (2003); Criminisi et al. (2004); Wexler et al. (2007); Kawai et al. (2009); Aujol et al. (2010); Arias et al. (2011); Mansfield et al. (2011)); and geometry-oriented, which interpolate the inpainting domain by continuing the geometric structure of the image. In these methods the images are modeled as functions with some degree of smoothness, expressed, for instance, in terms of the curvature of the level lines (Masnou and Morel (1998); Ballester et al. (2001); Chan and Shen (2001a); Masnou (2002); Citti and Sarti (2006); Cao et al. (2011)) or the total variation of the image (Chan and Shen (2001b)).

In this work we are interested in a particular type of geometry-oriented methods: binary inpainting, which is usually used to disocclude shapes. Shape inpainting can be achieved by implementing Euler's elastica, which is defined as the curve with minimal length that joints two points x_1 and x_2 and is also tangent to the straight lines, t_{x_1} and t_{x_2} , associated to these points (Bernoulli (1692); Euler (1744); Levien (2008)). As it is not lower semicontinuous some relaxed versions have been proposed by Bellettini et al. (1993), Masnou and Morel (1998) and Ballester et al. (2001), which are compatible with the amodal completion theory of Kanizsa (1991). In a work of Kang et al. (2014) that proposes a computational method for modal completion, the elastica is a key ingredient to obtain illusory contours. It is also used in a method, proposed by Citti and Sarti (2006), for both modal and amodal completion which uses geodesics

in the group of rotations and translations.

Binary inpainting models can be implemented by means of threshold dynamics, firstly proposed by Merriman et al. (1992), which is based on diffusion processes followed by thresholding. Threshold dynamics interpolations usually minimize geometric functionals, based either on the length (Esedoglu et al. (2005)), area (Grzhibovskis and Heintz (2008)), or curvature (Merriman et al. (1992)) of the shape contours.

We propose to disocclude the objects using the principle of good continuation modelled by minimizing the Euler's elastica and using the threshold dynamics algorithm of Esedoglu et al. (2005). As the elastica is not convex we propose to introduce global cues, such as relatability and convexity to provide an initial completion close to the one expected by the perception theory.

6.2 Depth ordering: a Bayesian approach

Our scene interpretation model is inspired by the proposal of van Lier et al. (1994), who suggest to choose the preferred scene interpretation based on the minimum complexity or description code, by taking into account local and global aspects of occlusion. Their work assumes that the most likely interpretation is the one that minimizes the sum of the complexity of three components of the visual pattern:

1. The internal structure, related to each of the visible shapes separately.
2. The external structure, related to the positional relation between these shapes.
3. The virtual structure, related to the occluded parts of the shapes.

The perceptual complexity of each of these three components is expressed in terms of structural information theory (SIT) (Leeuwenberg and Van der Helm (2013)), an extension of the Gestalt theory that

provides a formal and manual calculus to decide the plausibility of the perceptual interpretations. However, van Lier et al. (1994) do not automatically complete the occluded objects and the complexities are manually estimated from line drawings, thus their approach can not be directly applied to images in a computer vision task.

Knill et al. (1996) and van der Helm (2011) also noticed that the global minimum principle, i. e. the criterion of selecting the most preferred interpretation, can be settled in a Bayesian framework by properly defining prior and conditional probabilities.

We formalize the proposal of van Lier et al. (1994) and van der Helm (2011) by proposing a fully automatic method that can be applied to any image decomposed in shapes. Once the objects conforming the scene are disoccluded we follow a Bayesian approach to decide the structure of the scene. We give definitions for the prior probability and the likelihood, measured, respectively, by the object complexities and an elastica-based quantity that measures the length of the occluded and the disoccluded boundaries. As a consequence, our probability model takes into account the shape of the objects in the hypothesized scenes as well as the effort of bringing these objects in their relative positions in the visual image.

7

An Elastica Based Model for Scene Structure

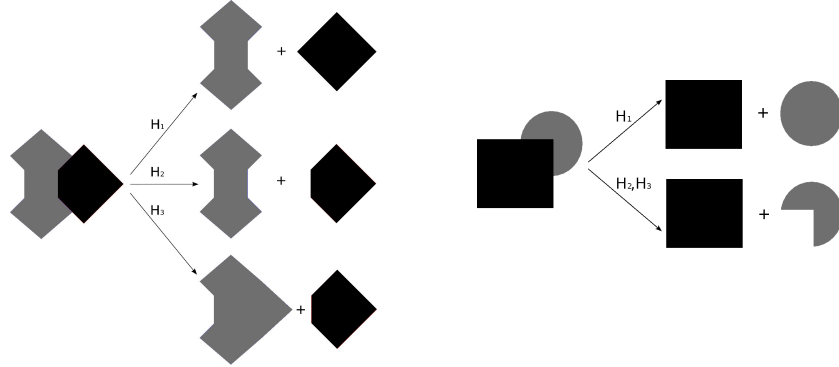
In this Chapter we present a computational model to recover the most likely interpretation of the 3D scene from a planar image, where some objects may occlude the others. In particular, our model estimates the depth order of the objects and gives a plausible completion of the partly occluded objects.

We propose a model to recover the scene structure of a planar image that is grounded in two elements:

1. A binary inpainting method that disoccludes the hidden regions of the objects. It gives us the complete objects that conform the different scene configurations compatible with the planar image.
2. A probabilistic model that quantitatively justifies which scene configuration is the preferred one.

As we are considering three-depth images, i.e. objects at two depths forming a scene in front of a background, there are three possible interpretations or hypothesis of the real 3D scene:

- H_1 : Objects at depth A occluding objects at depth B.
- H_2 : Objects at depth B occluding objects at depth A.
- H_3 : All objects fitting together forming a mosaic.



(a) After disoccluding we get different shapes for each hypothesis. (b) The shapes coincide in two of the three hypothesis.

Figure 7.1: Two examples with its corresponding hypothesis, H_i , $i = 1, 2, 3$, for describing the 3D scene structure that gives rise to the observed image.

We exemplify these possible hypothesis in Figure 7.1, where two images of different scenes showing its different hypothesis are presented. Let us observe that sometimes the objects in interpretation H_3 , i.e., A and B fitting together, might coincide with the ones in H_1 or H_2 , as can be seen in Figure 7.1b; or even, as the image of Table 9.3 shows, the shapes may be the same in all three hypothesis. This phenomenon is related to the optical illusion of relative depth between the objects. For instance, in Figure 7.2 we present two images where the real objects are not occluded, therefore the objects coincide in all three hypothesis, $H_1 = H_2 = H_3$, but we still have to decide which is the correct configuration. Consequently, in our model, even when the objects forming the scene coincide we consider the three different hypothesis with its respective depth ordering.

In the Perception community, the observed image is often called the *proximal stimulus* (e.g., the left image in Figure 7.1a and 7.1b), and each of the hypothesized interpretations H_i is called the *distal stimulus*.



(a) Author: Edoardo Accenti



(b) Author: Laurent Laveder

Figure 7.2: Two optical illusions where the actual depth order of the scene objects is ambiguous or undetermined unless we know the objects.

To decide which is the correct distal stimulus we compute the several distal interpretations of the scene which are compatible with the proximal planar image. In this sense we follow the ideas proposed by Rubin (2001), who states that T-junctions are used to launch the completion process when contours are relatable (Kellman and Shipley (1991)). Then, the Gestalt law of good continuation plays an important role. This motivates us to use the Euler's elastica in order to smoothly continue the contours behind the occluder. Euler's elastica is the minimum curve that joints two T-junctions at points x_1 and x_2 , with tangents τ_{x_1} and τ_{x_2} to the respective terminating stems, with a smooth continuation curve. It is defined as the solution of minimizing the following energy:

$$\int_{\gamma} (\kappa^2(s) + \beta) ds, \quad (7.1)$$

where $\beta > 0$ and the minimum is taken among all the curves γ joining x_1 and x_2 with tangents τ_{x_1} and τ_{x_2} , respectively, $\kappa(s)$ denotes the curvature of γ and ds its arc length. The parameter β plays a geometric role by settling the expected underlying a priori regularity: with a larger β , the energy favours the completion with straight lines (minimal length); while for a small β favours smooth curves of low curvature, even if their length is increased¹. Figure 7.3 shows

¹In the limit case ($\beta = 0$) the energy to be minimized is the Willmore energy,

an example illustrating how the parameter β affects the disoccluded shape. When β is big, more weight is given to the length of the curve and then straight lines are favoured. When β decreases the disoccluded objects tend to smooth their shape no matter if it produces a bigger length.

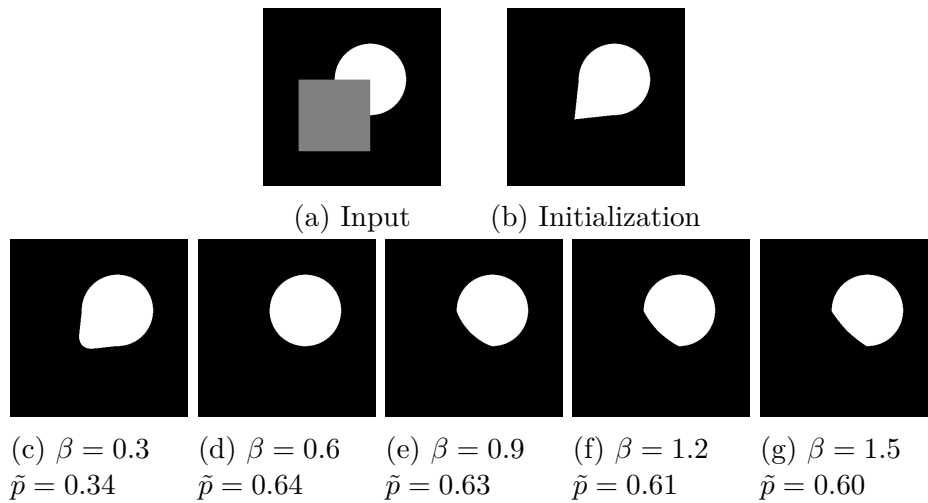


Figure 7.3: Disocclusion results depending on β and its certainty \tilde{p} .

On the other hand, depending on the resolution of the proximal stimulus the parameter β needs to be adapted to obtain the same underlying shape regularity. This property can be explained by considering curvy boundaries with smaller or bigger curvature. Indeed, an example is shown in Figure 7.4: circles with larger radius need a larger value of β in order to obtain the same regularity of the disoccluded shape. The reason is the following: the curvature of smooth plane curves is defined as the inverse of the radius of the osculating circle (the unique circle which most closely approximates the curve near the point). Therefore, there is a relationship between the numerical curvature of the disoccluded objects and the a priori regularity

$$\int_{\gamma} \kappa^2(s) ds.$$

imposed through the parameter β : the larger the β , the larger the expected radius of the osculating circle.

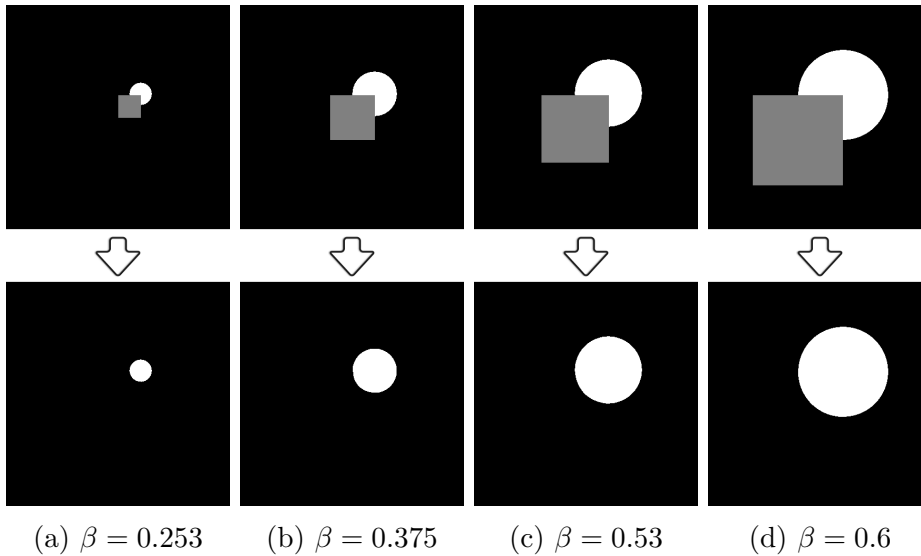


Figure 7.4: To obtain the same shape we need to adapt parameter β according to the resolution.

In this work, the elastica is used in two ways. We propose in Section 7.1 an elastica-based object disocclusion method which incorporates the reliability of partially occluded contours and the convexity of the disoccluded objects. On the other hand, the elastica is also used in Section 7.2 to select the most probable disoccluded scene.

7.1 Elastica-based object disocclusion

As we are solely concerned by the shape of the objects, we work with segmented images and we perform a geometric inpainting of the binary shapes that represent these objects. More precisely, we disocclude each object in each hypothesis by separately considering the hypothesized occluding object as the inpainting mask. The object

is automatically completed in such a way that its boundary minimizes a relaxed version of the elastica (7.1). For that, the object to be completed is represented in a binary image (given by the object segmentation) and its completion is performed through a threshold dynamics algorithm which consists in a diffusion process followed by a thresholding. In our case, the minimization algorithm, which is proposed in Esedoglu et al. (2005), iteratively alternates the following steps:

- One step of the scheme presented in Grzhibovskis and Heintz (2008) that decreases $\int_{\gamma} \kappa^2 ds$.
- One step of the standard Merriman et al. (1992) scheme that decreases $\beta \int_{\gamma} ds$.
- A thresholding step.

We present the pseudo-code and more details in Chapter 8, Algorithm 3.

7.1.1 Initialization of the inpainting mask

Since the elastica energy (7.1) is not convex, the inpainting result depends on the initial condition inside the inpainting mask. We illustrate it in Figure 7.5, which shows the inpainting results (shown in the second row) obtained by minimizing the elastica with different initializations (shown in the first row): white, black, random (black and white chosen randomly from a uniform distribution) or with our proposal, which is explained in the remainder of this section. Notice how the proposed initialization gives a better result (according to the Gestalt laws of perception) and produces a completion that maintains the tangents at the endpoints of the disoccluded boundary.

In order to automatically compute an initialization of the inpainting problem sufficiently close to what humans perceive as disoccluded objects by amodal completion, we incorporate perceptual cues such

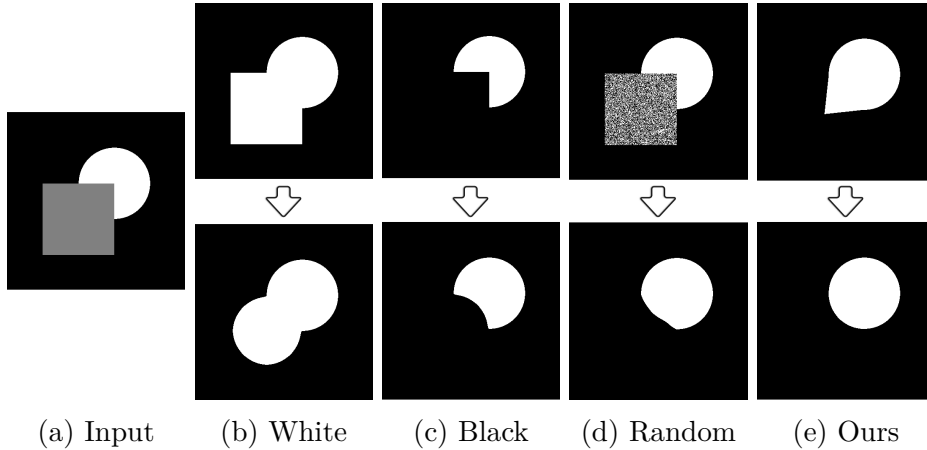


Figure 7.5: Inpainting results starting with several initializations.

as relatability of object contours (Kellman and Shipley (1991)) and convexity of the disoccluded objects.

The notion of relatability (see Figure 6.5) was introduced by Kellman and Shipley (1991) in the attempt of defining under which conditions visual completion occurs. Let us recall the definition of relatability.

Definition 3 (Relatability (Kellman and Shipley (1991); Singh and Hoffman (1999))). Let x_1 and x_2 be two points with tangent vector τ_{x_1} and τ_{x_2} , respectively. Consider the semilines:

$$\begin{aligned} s_1 &= \{x_1 + \lambda\tau_{x_1}, \lambda \geq 0\} \\ s_2 &= \{x_2 + \lambda\tau_{x_2}, \lambda \geq 0\} \end{aligned} \quad (7.2)$$

Then, s_1 and s_2 are relatable if:

- a) Semilines s_1 and s_2 intersect.
- b) The directed angle from τ_{x_1} to $-\tau_{x_2}$ is acute or 90° .

In fact, Singh and Hoffman (1999) also proved that this definition is equivalent to the existence of a smooth contour without inflection

points connecting x_1 and x_2 , and that the interpolating curve doesn't turn through a total angle of more than $\frac{\pi}{2}$.

As shown by Burge et al. (2010) non-occluded objects in the world tend to be convex, therefore we favour the convexity of the disoccluded object by taking advantage of the following well-known property of convex sets.

Lemma 2. *Every closed convex set in R^n is the intersection of the closed half-spaces that contain it.*

The automatic initialization of the binary image inside the inpainting mask is illustrated in Figure 7.6. In practice, we follow these steps:

1. Consider all the T-junctions of the object contours arriving to the inpainting mask together with their tangents (illustrated in Figure 7.6b). In order to compute these tangents we use the Line Segment Detector proposed by von Gioi et al. (2012).
2. Calculate all the possible pairs of relatable contours (shown in Figure 7.6c).
3. Then, for each pair of relatable contours, for the T-junction x_i and tangent τ_{x_i} we consider the half-space.

$$\{x \in \mathcal{R}^2 : \langle \tau_{x_i}^\perp, x \rangle - \langle \tau_{x_i}^\perp, x_i \rangle \geq 0\} \quad (7.3)$$

(or ≤ 0 , depending on which half-space the object is), and we assign a vote to the half-space on which the known object is. Figure 7.6d displays the image gathering of these votes in the inpainting mask, where brighter colors mean more votes. Let us remark that, in order to better illustrate our initialization, in Figure 7.6d and Figure 7.6e we only show the computed values inside the inpainting mask.

4. Finally, we binarize the image containing the votes with a threshold based on a rank order filter of these votes. We order the

votes in increasing order and start with a threshold with the value ranked at percentile 75th. If no new connected component appears in the initialization with this threshold we keep it. Otherwise we decrease the threshold by 5 percentiles and repeat the process until no new connected components appear. Two different examples are shown in Figure 7.6e, they are the initialization of the binary inpainting algorithm. First row shows an example where the threshold on the votes corresponds to the 75th percentile while in the second row the threshold was automatically decreased to the 65th percentile in order to obtain an initialization with a single connected component.

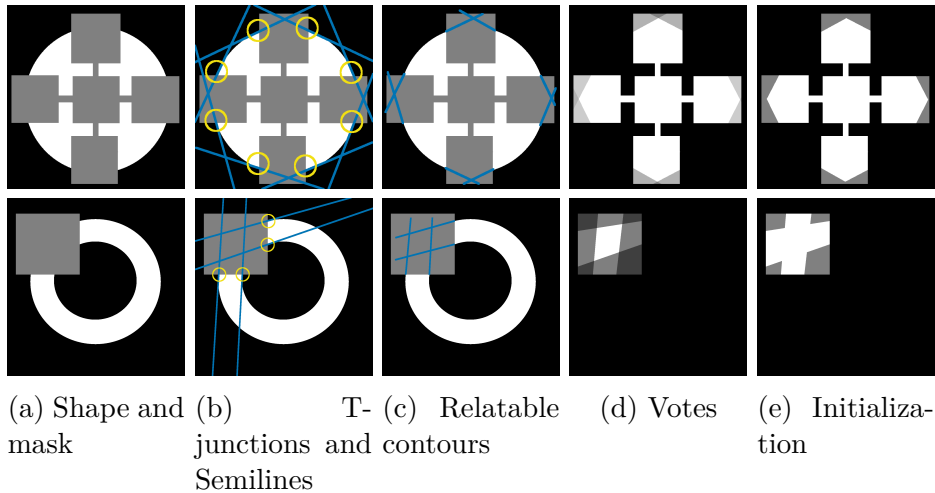


Figure 7.6: Two examples of reliability- and convexity-based initialization of the inpainting mask.

7.2 Elastica-based probabilistic model

Following the idea of Knill et al. (1996) we introduce a Bayesian approach to choose the most plausible scene interpretation, among all

the possible interpretations. We propose definitions for the prior and the conditional probabilities which take into account the global complexity of the objects in the hypothesized scenes as well as the effort of bringing the objects in their relative positions in the visual image. As a consequence, the result of this probability model indicates that the most simple interpretation is the one that more likely results from the amodal completion process, which was also suggested by van der Helm (2011).

Inspired by the work of van Lier et al. (1994), to define the prior probability of the hypothesized scene our probabilistic model takes into account the global complexity of both objects, already being disoccluded. The likelihood, i.e. the conditional probability of the given image (proximal stimulus) given a certain hypothesis (distal stimulus) is defined through an Euler's elastica-based quantity that measures two attributes: the effort of bringing these objects in their relative positions given in the image and the smoothness of the disoccluded boundaries.

We justify the preferred interpretation by maximizing the responsibility or a posterior probability, given by the Bayes' rule as

$$\arg \max_i p(H_i/I) = \arg \max_i \frac{p(I/H_i)p(H_i)}{p(I)} \quad (7.4)$$

over the hypothesized interpretations H_i , where I is the proximal stimulus or given image. As the quotient $p(I)$ remains the same for all hypothesis H_i in the maximization process, Equation (7.4) is equivalent to

$$H_P = \arg \max_i p(H_i/I) = \arg \max_i p(I/H_i) p(H_i). \quad (7.5)$$

7.2.1 Conditional Probability: Relative Position Complexity

Given the underlying hypothesis H_i and the proximal image I , we define the conditional probability $p(I/H_i)$ as

$$p(I/H_i) \propto \tilde{p}(I/H_i) = e^{-\omega_1 \int_{\mathcal{B}_i^c} (\kappa^2 + \beta) ds} e^{-\omega_1 \int_{\mathcal{B}_i^d} (\kappa^2 + \beta) ds}, \quad (7.6)$$

where \mathcal{B}_i^c and \mathcal{B}_i^d stand for common and disoccluded boundaries, respectively and ω_1 is a normalization constant, defined as the inverse of the maximum $\tilde{p}(I/H_i)$, $i = 1, 2, 3$.

In Figure 7.7 we show two examples of how these boundaries are considered: Figure 7.7a considers the case of a square in front of circle, when the circle is disoccluded the common boundary \mathcal{B}_1^c among both objects is formed by the 2 T-junction points (in blue), while the disoccluded boundary \mathcal{B}_1^d is made of all the boundary of the circle that was behind the square (in black); on the other hand, Figure 7.7b shows the hypothesis of a square at the same depth than the circle-part; the common boundary \mathcal{B}_2^c is shown in blue while the disoccluded boundary coincides with it, $\mathcal{B}_1^d = \mathcal{B}_1^c$, due to the fact that we consider closed objects.



(a) H_1 : square in front of circle. (b) H_3 : objects at same depth.

Figure 7.7: Boundaries \mathcal{B}_i^c and \mathcal{B}_i^d ($i = 1, 2$) for hypothesis H_1 and H_3 from Figure 7.1b.

Formula (7.6) measures the responsibility that hypothesis H_i takes for explaining the proximal stimulus I as well as the deviation of I

from H_i . With the first integral of (7.6) we compute the difficulty of bringing the two objects together in order to get the perceived image taking into account only the known boundary of the objects; for example, it is easier to obtain configuration 7.7a than 7.7b as in the first case only two points need to coincide, independently of the two coinciding points we will perceive the same image, and in the other case, H_2 , a larger boundary needs to coincide in order to perceive exactly that configuration. The second integral takes into account the regularity of the occluded boundary of the shape to define the probability of obtaining a particular stimulus; for example in Figure 7.8a we can move the disc at many different positions behind the square to obtain the same kind of proximal stimulus we are observing, while in Figure 7.8b the movements we can do are more limited, as the perceived image will change drastically.



(a) The square can be moved and we get the same proximal stimulus.

(b) If the square is moved the proximal stimulus changes.

Figure 7.8: Example of two different disocclusions.

Let us remark that due to the way we disocclude the objects the resulting disoccluded boundaries are always smooth; if we had different models of disocclusion this term would help to distinguish among them (in addition to the prior term). For instance, with our disocclusion model based on the elastica we are not able to recover the occluded object in Figure 7.8(b) or objects A in Figure 6.3a. The probability distribution in (7.6) also appeared in Mumford (1994)

and Williams and Jacobs (1997), who characterized the probability distribution of the shape of boundary completions based on the paths followed by a particle undergoing a stochastic motion, a directional random walk. It turns out that the elastica has the interpretation of being the mode of the probability distribution underlying this stochastic process restricted to curves with prescribed boundary behaviour, i. e. the maximum likelihood curve with which to reconstruct hidden contours.

Let us comment that in our definition (7.6), when visual completion occurs while propagating the stem, (e.g., hypothesis H_1 in Figure 7.1b; also, hypothesis H_1 in Figure 7.7a), the common boundaries \mathcal{B}^c between the objects are reduced to the T-junctions. In this case: $\int_{\mathcal{B}^c} (\kappa^2 + \beta) ds = 0$ and thus $e^{-\omega_1 \int_{\mathcal{B}^c} (\kappa^2 + \beta) ds} = 1$. Let us notice that in the distal stimulus, since we are considering closed objects, \mathcal{B}^c belongs to both objects. Therefore, in the hypothesis where the objects are interpreted as being fit-together (e.g., hypothesis H_2 in Figure 7.1b; also, hypothesis H_2 in Figure 7.7b), a disoccluded boundary \mathcal{B}^d appears which coincides with \mathcal{B}^c (i.e., $\mathcal{B}^d = \mathcal{B}^c$). Let us also comment on the effect of the regularity of \mathcal{B}^c . Figure 7.9 presents three different proximal stimuli or images. The numerical computation of the term $e^{-\omega_1 \int_{\mathcal{B}^c} (\kappa^2 + \beta) ds}$ associated to each of the three images will decrease from left to right in the fit-together (or mosaic) interpretation.

7.2.2 Prior probability: Objects Complexity

Prior probabilities are defined as

$$p(H_i) \propto \tilde{p}(H_i) = e^{-\omega_2 \mathcal{C}(O_i^1)} e^{-\omega_2 \mathcal{C}(O_i^2)}, \quad (7.7)$$

where O_i^1 and O_i^2 are the (disoccluded) objects in the hypothesized interpretation H_i and ω_2 is a normalizing constant defined as the inverse of the maximum value of $p(H_i)$, $i = 1, 2, 3$. The factor $\mathcal{C}(O_i^j)$ denotes the complexity of the object or shape O_i^j at depth j . In the case that the object at one depth is formed by more than one



Figure 7.9: Three different proximal stimulus or images. From left to right, visual completion will become more and more evident than the interpretation of two pieces fitting together, both perceptually and quantitatively, with probability (7.6).

connected component the complexity is computed separately for each connected component and their sum constitutes the complexity of O_i^j .

We use the definition of complexity of a shape defined by Chen and Sundaram (2005),

$$C(O) = (1 + R) \left(0.6 \cdot \min(C_{\text{dist}}, C_{\text{angle}}) + 0.07 \max(C_{\text{dist}}, C_{\text{angle}}) + 0.33P \right), \quad (7.8)$$

which takes into account global properties of the shape such as contour symmetries and repetitions. In particular, it computes:

- The *global distance entropy* (C_{dist}), which is defined as the distance of boundary points to the centroid of the shape.
- The *local angle entropy* (C_{angle}) is the angle formed by the two segments joining three consecutive boundary points.
- The *perceptual smoothness* (P) is computed using the local angle: as closer to π the angle, the smoother the shape.
- The *measure of shape randomness* (R) is the maximum difference between two random traces obtained from the two more distant points of the boundary.

Therefore, the proposed prior probability implicitly considers global properties such as shape contour symmetries and repetitions.

Let us notice that with these definitions our whole model for amodal completion is able to choose, not only between the different hypothesis for a fixed disocclusion parameter β but also between several disocclusions associated to different parameters β , and therefore to take into account global completion properties such as symmetry or repetitions. In Figure 7.3 there is an example illustrating this computational ability, where we provide the probability of each disoccluded result.

Let us finally observe that in Figure 7.9 the complexity-related terms $e^{-\omega_2 \mathcal{C}(O_1^i)}$ and $e^{-\omega_2 \mathcal{C}(O_i^2)}$ will decrease from left to right, as happened to the conditional probability, and the visual completion will become more and more evident and the interpretation of two complex pieces fitting together will become perceptually less favourable.

Let us finally remark that we are not considering *all* possibles configurations as proposed by von Gioi (2009) but only the ones favoured by relatability, convexity, and good continuation. On the other hand, even if global cues such as symmetry or repetitions are taken into account in our probability model, we do not incorporate them in the disocclusion algorithm.

8

Algorithm and implementation details

In this Chapter we detail the three algorithms that describe our model: the complete model, the inpainting method and the one that computes the probabilities.

8.1 Complete Algorithm

Algorithm 2 shows the steps of the whole numerical algorithm. Our algorithm needs a decomposition of the given image into objects and object parts which are interpreted as projections of real 3D objects on the image plane. This decomposition can be given either from the classical decomposition in level sets, in bi-level sets or segmenting the image from a criterion. In this thesis, for the synthetic images, we use the decomposition in bi-level sets, which are defined as $X^{(\lambda_n, \lambda_{n+1})}I = \{x \in \Omega : \lambda_n \leq I(x) < \lambda_{n+1}\}$, where Ω is the image domain and $\{\lambda_n\} \subset \mathcal{R}$ is a finite strictly increasing sequence. In our experiments $\{\lambda_n\} = \{\lambda_1, \lambda_2, \lambda_3\}$ with $\lambda_1 = 0$, $\lambda_2 = 128$ and $\lambda_3 = 255$. For the real images, we use the segmented shapes from the Berkeley segmentation dataset created by Martin et al. (2001).

Objects appearing at the image are denoted by X_1 and X_2 . From X_1 and X_2 the three hypothesis will be considered by the algorithm: X_1 occluding the distal object D_2 (corresponding to the inpainted

result of proximal X_2), X_2 occluding the distal object D_1 (corresponding to the inpainted result of proximal X_1), and X_1 and X_2 fitting together. Now, by applying the disocclusion method of Section 7.1 where X_1 and X_2 are the inpainting mask of hypothesis H_1 and H_2 , we compute the *complete* hypothesis $H_1 = X_1 \cup D_2$ and $H_2 = X_2 \cup D_1$. Then, to this two hypothesis, we always add the additional hypothesis $H_3 = X_1 \cup X_2$ of the *mosaic* interpretation (which is obtained when we do not apply the disocclusion algorithm). For each H_i we compute the probabilities $\tilde{p}(I/H_i)$ and $\tilde{p}(H_i)$ from the definitions in Section 7.2. Finally, we compute the perceptually preferred hypothesis H_P using (7.5).

Algorithm 2: Pseudo-code summarizing the proposal.

Input : An image I with objects X_1 and X_2 .
Output: The set of distal hypothesis H_1, H_2, H_3 , each one made of complete objects at two depths, and the preferred one H_P (with $P \in \{1, 2, 3\}$).

```

for  $i \in \{1, 2, 3\}$  do
  if  $i \neq 3$  then
    • Consider  $X_i$  as inpainting mask and initialize the inpainting mask using the perceptual method described in Sect. 7.1.1
    • Disocclude object  $X_j$ , with  $j \neq i$ , using Algorithm 3, that is implementing the elastica-based method of Sect. 7.1. From it, we obtain the disoccluded object  $D_j$  and the completed hypothesis  $H_i = X_i \cup D_j$ .
  else
    | • Set  $H_3 = X_1 \cup X_2$ 
  end
  • Compute the probabilities  $\tilde{p}(I/H_i)$  with Algorithm 4 and  $\tilde{p}(H_i)$  (equation (7.8)) from the definitions given in Sect. 7.2.
end
Set  $H_P = \arg \max_i \tilde{p}(I/H_i) \tilde{p}(H_i)$ .

```

8.2 Inpainting Algorithm

In Algorithm 3 we describe the threshold dynamics method that we use for disocclusion.

Algorithm 3: Pseudo-code of the discocclusion algorithm.

Input : A binary image I containing a region without information, the inpainting region $\tilde{M} \subset \Omega$, and the elastica parameter $\beta > 0$.

Output: Disoccluded object D (given by an inpainted binary image \bar{I})

- Set $\alpha = 0.99$ and $\delta t = 12$.
- Set the initial shape $\Sigma_0 = \{x : I(x) = 1\}$.
- Set $n = 0$ and $\Sigma_1 = \Omega$.

while $\|\Sigma_{n+1} - \Sigma_n\| > 10^{-3}$ **do**

1. A step of Grzibovskis-Heintz algorithm. Set:

$$\Gamma_1 = \left\{ x : 2\alpha G_{\sqrt{\delta t}} * \mathbb{1}_{\Sigma_n}(x) - 2G_{\alpha^2 \sqrt{\delta t}} * \mathbb{1}_{\Sigma_n}(x) \leq \alpha - 1 \right\}.$$

2. A step of standard Merriman-Bence-Osher algorithm. Set:

$$\Gamma_2 = \left\{ x : G_{\beta \delta t} * \mathbb{1}_{\Gamma_1}(x) \geq \frac{1}{2} \right\}.$$

3. Fidelity step. Set

$$\Sigma_{n+1} = (\Gamma_2 \cap \tilde{M}) \cup (\Omega \setminus \tilde{M}).$$

4. $\Sigma_n = \Sigma_{n+1}$.

end

- $\bar{I} = \mathbb{1}_{\Sigma_n}$ and $D = \Sigma_n$.
-

Observe that the Grzibovskis-Heintz step depends on $\alpha \in (0, 1)$ (Esedoglu et al. (2005, 2008)).

The Gaussian convolution has been computed using the Lindeberg's discrete scale-space method and the implementation described in Otero and Delbracio (2016), that is, we use that the Gaussian convolution $v(x, \delta t) = (G_{\sqrt{2\delta t}} * u)(x)$ is the solution of the heat equation $\frac{\partial v}{\partial t} = \Delta v$ for a diffusion time δt (set to 12 in our experiments, to guarantee the prescribed upper and lower bounds depending on the curvature of the visible shape (Merriman et al. (1992))). To solve the heat equation we need to discretize partial derivatives. The discretization is done using the discretization of Otero and Delbracio (2016):

$$\Delta_\gamma v = (1 - \gamma)\Delta_+ v + \gamma\Delta_\times v \quad (8.1)$$

where,

$$\Delta_+ v_{k,l} = v_{k+1,l} + v_{k-1,l} + v_{k,l+1} + v_{k,l-1} - 4v_{k,l} \quad (8.2)$$

$$\Delta_\times v_{k,l} = \frac{1}{2}(v_{k+1,l+1} + v_{k+1,l-1} + v_{k-1,l+1} + v_{k-1,l-1}) - 2v_{k,l} \quad (8.3)$$

and $0 < \gamma \leq 0.5$, we use $\gamma = 0.5$ in our experiments. We refer to Otero and Delbracio (2016) for more details on the discretization.

8.3 Likelihood Estimation

In Algorithm 4 we present the algorithm for computing the conditional probability $\tilde{p}(I/H_i)$, $i = 1, 2, 3$. The discrete boundaries of each shape are computed as external boundaries and using 4-connectivity. On the other hand, in order to compute the curvature of a discrete curve (or boundary) γ , we use the method of Sethian (1985) and compute

$$\kappa(x) = \operatorname{div} \left(\frac{\nabla \phi(x)}{|\nabla \phi(x)|} \right), \quad (8.4)$$

where ϕ is the signed distance function to the boundary γ . We use forward derivatives to compute the gradient and backward deriva-

tives for the divergence. The discrete signed distance function u is computed using the algorithm explained in Meijster et al. (2002).

Finally, the prior probability is computed using (7.7) with the complexity measure given by (7.8). We consider as boundary points for computing (7.8) all the pixels that form the boundary of an object. In case of an object formed by more than one connected component we compute the complexity (7.8) of every connected component and the final complexity measure is the addition of the individual complexities. For details about how to compute R , C_{dist} , C_{angle} and P we refer to Section 7.2 and to Chen and Sundaram (2005).

Algorithm 4: Pseudo-code of the algorithm computing $\tilde{p}(I/H_i)$ for $i = 1, 2, 3$.

Input : Inpainting masks X_1, X_2 , disoccluded objects D_1, D_2 , elastica parameter β .

Output: Conditional probability of each hypothesis H_1, H_2, H_3 .

- Compute the boundaries $\partial X_1, \partial X_2, \partial D_1, \partial D_2$ of X_1, X_2, D_1, D_2 , respectively.
- Set $\mathcal{B}_i^c = \partial X_1 \cap \partial X_2, \quad i = 1, 2, 3$.

for $i = 1, 2$ **do**
 | $\mathcal{B}_i^d = \partial D_i \setminus \partial X_i$
end

- Set $\mathcal{B}_3^d = \mathcal{B}_3^c$

for $\{i = 1, 2, 3\}$ **do**
 | $E_{B_i} = \sum_{x \in \mathcal{B}_i^c} (\kappa^2(x) + \beta) + \sum_{x \in \mathcal{B}_i^d} (\kappa^2(x) + \beta)$
end

- Set $\omega_1 = \max\{E_{B_1}, E_{B_2}, E_{B_3}\}$

for $i = 1, 2, 3$ **do**
 | $\tilde{p}(I/H_i) = \exp\{-\omega_1 E_{B_i}\}$
end

9

Experimental results

In this Chapter we present the behavior of our method on real and synthetic images. We also present some results of foreground-background on synthetic images. Moreover, for some real images we have the ground-truth and we compare our result with it.

The proposed method has been tested with synthetic and real images. Let us recall that our method assumes the proximal stimulus to be decomposed into objects and object parts (which can be interpreted as projections of real 3D objects on the image plane). As in the synthetic experiments the images are formed by objects with a single and unique color, this already gives a segmentation and we apply our algorithm directly. For the real experiments, we use a segmentation of the image. In particular, we have taken segmented images from the Berkeley segmentation dataset proposed by Martin et al. (2001) and from the dataset introduced in Li et al. (2013).

Parameter β , which sets the underlying a priori regularity (see comments on its role in Section 7.1), has been fixed to 0.6 for all the experiments in order to have an algorithm as general as possible. There are two exceptions: Proximal 2 of Table 9.1 and Example 4 of Table 9.11, where β is fixed to 1.2 and 1.7, respectively, due to the biggest size of the circular shapes. As explained in Section 7.1, there is a relationship between the numerical curvature of the disoccluded objects and the a priori regularity imposed through the parameter

β : the larger the β , the larger the expected radius of the osculating circle locally approximating the curve.

The experiments are organized in Sections and Tables as follows:

- Section 9.1 presents the synthetic experiments that agree with our perception. They are shown in Tables 9.1, 9.2 and 9.3.
- Section 9.2 introduces results over real images that agree with our perception. They are shown in Tables 9.4, 9.5, 9.6 and 9.7. Table 9.5 shows our results on images of the Berkeley dataset with figure-ground ground-truth provided by Fowlkes et al. (2007).
- Section 9.3 presents Table 9.9, that shows the ability of our method to also decide on (perceptually) fully visible objects over a background.
- Section 9.4 shows and discuss the experiments that failed.

For each row in each table we show a complete experiment: We first present the proximal piecewise-constant image I on the left (in the case of real images it includes both the original input and the segmented version), followed by the three hypothesis H_i (each one separated by a gray box), together with the values $\tilde{p}(I/H_i)$ and $\tilde{p}(H_i)$ proportional to the conditional probability and the prior probability, respectively, and the normalized probability value $p(H_i|I)$. We have normalized the probabilities in such a way that $p(H_1/I) + p(H_2/I) + p(H_3/I) = 1$. The probability value of the preferred hypothesis H_P is highlighted in boldface.

For the first two hypothesis, H_1 and H_2 , we display the objects at depth 1 on the left, and the disoccluded objects (at depth 2) on the right. Finally the last column is the hypothesis H_3 where the two objects are fitting together at the same depth.

9.1 Synthetic images

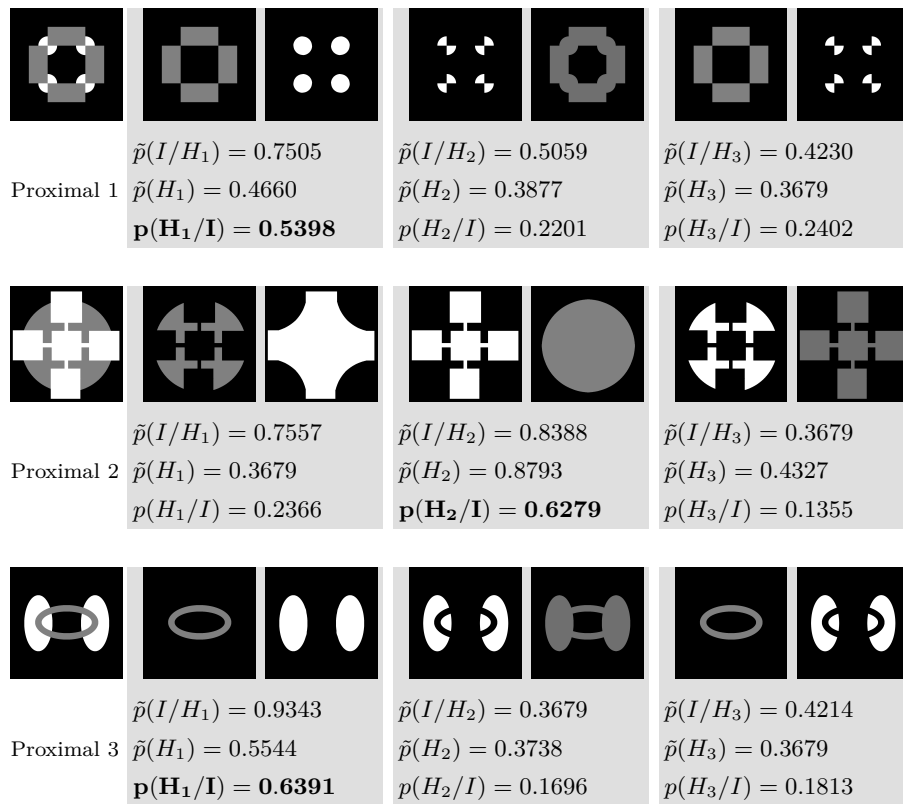
Tables 9.1, 9.2 and 9.3 show some experiments on synthetic images. In Table 9.2, the third hypothesis is not shown because it coincides with H_2 due to the fact that the disocclusion algorithm does not change the objects being disoccluded. In Table 9.3, a synthetic experiment where the three hypothesis coincide is shown: The disocclusion algorithm applied in the first two hypothesis does not change the objects and thus $H_1 = H_2 = H_3$, and the posterior probability is the same for all three hypothesis. As for depth order, H_3 is interpreted as two objects at the same depth (and having the real relative size which is observed in the proximal image) while H_1 can be interpreted as a gray square which is closer to the observer, plus a white rectangle which can be of bigger size but farther away from the square and whose boundary partially coincides with part of the boundary of the square. At last, H_2 can be interpreted as a white rectangle which is closer to the observer, plus a gray square which can be of bigger size but farther away from the rectangle and whose boundary partially coincides with part of the boundary of the rectangle. Notice that this situation is related to the ambiguity in depth of some proximal stimulus, sometimes causing optical illusion of relative depth perception as those in the images displayed in Figure 7.2.

Let us comment on the results corresponding to Proximal 6 and 7 of Table 9.1 and Proximal 9 of Table 9.2, which include quite similar shapes with equal occlusion signatures but different common boundaries among the shapes. In all of them the local perception cue at the T-junctions indicates that there is an occluded disc which continues behind an incomplete square (the occluder)¹. Our method is able to choose the corresponding preferred hypothesis, according to the T-junctions, as is shown by the probability values $p(H_1/I)$.


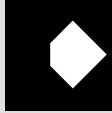

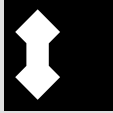


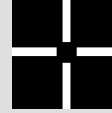
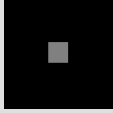


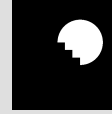


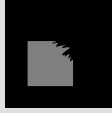
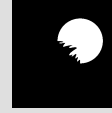
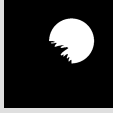
Finally, let us comment on Proximal 4 of Table 9.1 and on Proxi-

¹The perception literature acknowledges that, in a T-junction, the occluder is the surface on the T-head side while the surfaces on the T-stem side continue behind the occluder.

mal 12 on Table 9.2, which are two controversial examples as there is no agreement within people to decide which is the preferred one². In both cases, our algorithm favors local completion, that is, a completion that agrees with the T-junction cues and produces good continuation, instead of the global one which produces a more symmetric object (notice that the local completion in both cases produces a symmetric object with respect to one axis).



²Proximal 4 was also studied by van Lier et al. (1994) and they reported that the local hypothesis (H_1) is the most preferred by the subjects participating in their psychophysics experiments, although it is very tied.

 Proximal 4	 $\tilde{p}(I/H_1) = 0.7466$ $\tilde{p}(H_1) = 0.3910$ $\mathbf{p(H_1/I) = 0.3943}$	 $\tilde{p}(I/H_2) = 0.7445$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.3700$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.4743$ $p(H_3/I) = 0.2357$
 Proximal 5	 $\tilde{p}(I/H_1) = 8823$ $\tilde{p}(H_1) = 0.6020$ $\mathbf{p(H_1/I) = 0.5265}$	 $\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.3879$ $p(H_2/I) = 0.1414$	 $\tilde{p}(I/H_3) = 0.9108$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.3321$
 Proximal 6	 $\tilde{p}(I/H_1) = 0.4409$ $\tilde{p}(H_1) = 0.7056$ $\mathbf{p(H_1/I) = 0.4618}$	 $\tilde{p}(I/H_2) = 0.5562$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.3037$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.4295$ $p(H_3/I) = 0.2345$
 Proximal 7	 $\tilde{p}(I/H_1) = 0.7343$ $\tilde{p}(H_1) = 0.5995$ $\mathbf{p(H_1/I) = 0.4917}$	 $\tilde{p}(I/H_2) = 0.7823$ $\tilde{p}(H_2) = 0.4087$ $p(H_2/I) = 0.3572$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1512$

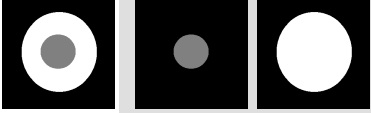











			
Proximal 8	$\tilde{p}(I/H_1) = 1$ $\tilde{p}(H_1) = 0.7087$ $\mathbf{p(H_1/I) = 0.7223}$	$\tilde{p}(I/H_2) = 0.3729$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.1398$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1379$

Table 9.1: Synthetic experiments where the three hypothesis are formed by different shapes.

		
Proximal 9	$\tilde{p}(I/H_1) = 0.6848$ $\tilde{p}(H_1) = 0.6487$ $\mathbf{p(H_1/I) = 0.6214}$	$\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.1893$
		
Proximal 10	$\tilde{p}(I/H_1) = 0.3877$ $\tilde{p}(H_1) = 0.3804$ $\mathbf{p(H_1/I) = 0.3527}$	$\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.3237$
		
Proximal 11	$\tilde{p}(I/H_1) = 0.5436$ $\tilde{p}(H_1) = 0.4051$ $\mathbf{p(H_1/I) = 0.4486}$	$\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.2757$






				
Proximal 12	$\tilde{p}(I/H_1) = 0.6172$ $\tilde{p}(H_1) = 0.3679$ $p(H_1/I) = \mathbf{0.4165}$		$\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.4324$ $p(H_2/I) = 0.2918$	

Table 9.2: Synthetic experiments where the fitting together hypothesis coincide with one of the others.




		
Proximal 13	$\tilde{p}(I/H) = 0.3679$ $\tilde{p}(H) = 0.3679$ $p(H/I) = 0.3333$	

Table 9.3: Synthetic experiment where the three hypothesis coincide.

9.2 Real images

In this section we show some results on real images from the Berkeley dataset created by Martin et al. (2001) and the dataset provided in Li et al. (2013).

We start illustrating that our method is robust to different segmentations of the same image. Table 9.4 shows a real image with a bear holding a branch and two different segmentations (representing the proximal stimuli). Both segmentations are from the ground-truths available in Martin et al. (2001). Segmentation 1 reflects that some flowers are partially occluding the bear and increasing the complexity of the bear shape; the flowers do not appear in segmentation 2 and thus the bear shape has a lower complexity (its complexity is

0.53, while in the previous case, Segmentation 1, was 1.34). Notice that the values of \tilde{p} are not comparable among the two experiments (only among different hypothesis within the same experiment) because they use a different normalizing constant ω_2 (see Section 7.2 for further details). Finally, the most preferred interpretation of the image coincides using the two different segmentations, i.e., it is a branch partially occluding a bear for both stimulus.










				
Segmentation 1				
		$\tilde{p}(I/H_1) = 0.7818$	$\tilde{p}(I/H_2) = 0.7590$	$\tilde{p}(I/H_3) = 0.3679$
		$\tilde{p}(H_1) = 0.6996$	$\tilde{p}(H_2) = 0.3717$	$\tilde{p}(H_3) = 0.3679$
		$\mathbf{p(H_1/I)} = 0.4184$	$p(H_2/I) = 0.3966$	$p(H_3/I) = 0.1850$
Segmentation 2				
		$\tilde{p}(I/H_1) = 0.6739$	$\tilde{p}(I/H_2) = 0.6676$	$\tilde{p}(I/H_3) = 0.3679$
		$\tilde{p}(H_1) = 0.5265$	$\tilde{p}(H_2) = 0.3679$	$\tilde{p}(H_3) = 0.3751$
		$\mathbf{p(H_1/I)} = 0.4805$	$p(H_2/I) = 0.3326$	$p(H_3/I) = 0.1869$

Table 9.4: Real images experiments (Martin et al. (2001)). Comparison of results with the same image, but different segmentations

In Table 9.5 we present results on images of the Berkeley dataset with provided figure-ground ground-truth labeled by humans. Then, Table 9.6 shows experimental results on real images from Li et al. (2013) and Table 9.7 shows results on images from the Berkeley Segmentation database of Martin et al. (2001). Each row shows a different experiment: the two left-most images are, respectively, the

original image and a segmentation of it, they are followed by the three different hypothesis (each one separated by a gray box). For the images in Table 9.5, superimposed on the original image, we display the provided figure-ground ground-truth Fowlkes et al. (2007) as a boundary in two colors, namely, black and white. The black side of the border indicates the object that is behind, while the white region indicates the frontal object.

Image 1		$\tilde{p}(I/H_1) = 0.8665$	$\tilde{p}(I/H_2) = 0.6820$	$\tilde{p}(I/H_3) = 0.3679$				
		$\tilde{p}(H_1) = 0.5212$	$\tilde{p}(H_2) = 0.3701$	$\tilde{p}(H_3) = 0.3679$				
		$\mathbf{p(H_1/I) = 0.5380}$	$p(H_2/I) = 0.3007$	$p(H_3/I) = 0.1612$				
Image 2		$\tilde{p}(I/H_1) = 1$	$\tilde{p}(I/H_2) = 0.8952$	$\tilde{p}(I/H_3) = 0.3679$				
		$\tilde{p}(H_1) = 0.3688$	$\tilde{p}(H_2) = 0.3724$	$\tilde{p}(H_3) = 0.3692$				
		$\mathbf{p(H_1/I) = 0.4410}$	$p(H_2/I) = 0.3972$	$p(H_3/I) = 0.1618$				
Image 3		$\tilde{p}(I/H_1) = 0.9520$	$\tilde{p}(I/H_2) = 0.7729$	$\tilde{p}(I/H_3) = 0.3679$				
		$\tilde{p}(H_1) = 0.3828$	$\tilde{p}(H_2) = 0.3864$	$\tilde{p}(H_3) = 0.3679$				
		$\mathbf{p(H_1/I) = 0.4564}$	$p(H_2/I) = 0.3741$	$p(H_3/I) = 0.1695$				
Image 4		$\tilde{p}(I/H_1) = 0.9899$	$\tilde{p}(I/H_2) = 0.7811$	$\tilde{p}(I/H_3) = 0.3679$				
		$\tilde{p}(H_1) = 0.4554$	$\tilde{p}(H_2) = 0.4124$	$\tilde{p}(H_3) = 0.3924$				
		$\mathbf{p(H_1/I) = 0.5064}$	$p(H_2/I) = 0.3416$	$p(H_3/I) = 0.1520$				

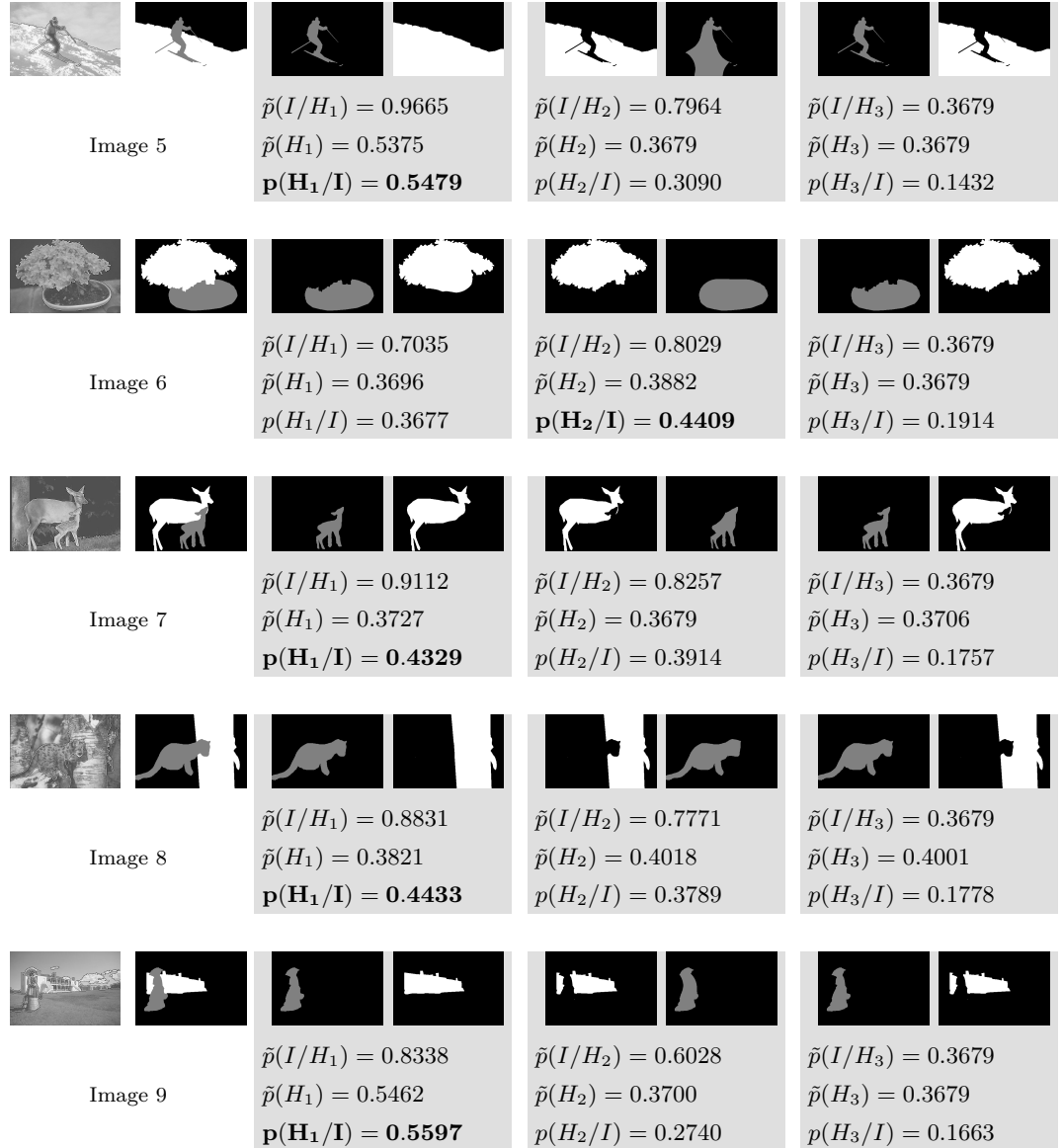


Image 10	$\tilde{p}(I/H_1) = 0.8935$	$\tilde{p}(H_1) = 0.3679$	$p(H_1/I) = 0.3390$	$\tilde{p}(I/H_2) = 0.9504$	$\tilde{p}(H_2) = 0.5218$	$\mathbf{p(H_2/I) = 0.5115}$	$\tilde{p}(I/H_3) = 0.3679$	$\tilde{p}(H_3) = 0.5219$	$p(H_3/I) = 0.1495$
Image 11	$\tilde{p}(I/H_1) = 0.8059$	$\tilde{p}(H_1) = 0.5367$	$\mathbf{p(H_1/I) = 0.5002}$	$\tilde{p}(I/H_2) = 0.8029$	$\tilde{p}(H_2) = 0.3696$	$p(H_2/I) = 0.3432$	$\tilde{p}(I/H_3) = 0.3679$	$\tilde{p}(H_3) = 0.3679$	$p(H_3/I) = 0.1565$
Image 12	$\tilde{p}(I/H_1) = 0.8940$	$\tilde{p}(H_1) = 0.3764$	$p(H_1/I) = 0.3620$	$\tilde{p}(I/H_2) = 0.9192$	$\tilde{p}(H_2) = 0.4979$	$\mathbf{p(H_2/I) = 0.4924}$	$\tilde{p}(I/H_3) = 0.3679$	$\tilde{p}(H_3) = 0.3679$	$p(H_3/I) = 0.1456$
Image 13	$\tilde{p}(I/H_1) = 0.7809$	$\tilde{p}(H_1) = 0.3712$	$p(H_1/I) = 0.3293$	$\tilde{p}(I/H_2) = 0.9201$	$\tilde{p}(H_2) = 0.4947$	$\mathbf{p(H_2/I) = 0.5170}$	$\tilde{p}(I/H_3) = 0.3679$	$\tilde{p}(H_3) = 0.3679$	$p(H_3/I) = 0.1537$

Table 9.5: Experiments with the depth ground-truth (Fowlkes et al. (2007)).

Image 14	$\tilde{p}(I/H_1) = 0.8526$ $\tilde{p}(H_1) = 0.3701$ $p(\mathbf{H}_1/I) = \mathbf{0.4356}$	$\tilde{p}(I/H_2) = 0.7416$ $\tilde{p}(H_2) = 0.3688$ $p(H_2/I) = 0.3776$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.4472$ $p(H_3/I) = 0.1868$
Image 15	$\tilde{p}(I/H_1) = 0.8796$ $\tilde{p}(H_1) = 0.3750$ $p(\mathbf{H}_1/I) = \mathbf{0.4539}$	$\tilde{p}(I/H_2) = 0.7074$ $\tilde{p}(H_2) = 0.3680$ $p(H_2/I) = 0.3582$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3729$ $p(H_3/I) = 0.1879$
Image 16	$\tilde{p}(I/H_1) = 0.9112$ $\tilde{p}(H_1) = 0.4034$ $p(\mathbf{H}_1/I) = \mathbf{0.5058}$	$\tilde{p}(I/H_2) = 0.6035$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.3055$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1887$
Image 17	$\tilde{p}(I/H_1) = 0.8833$ $\tilde{p}(H_1) = 0.3718$ $p(\mathbf{H}_1/I) = \mathbf{0.4263}$	$\tilde{p}(I/H_2) = 0.8279$ $\tilde{p}(H_2) = 0.6354$ $p(H_2/I) = 0.3980$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.6334$ $p(H_3/I) = 0.1757$

Table 9.6: Experiments with real images from Li et al. (2013).

Image 18	$\tilde{p}(I/H_1) = 0.7913$ $\tilde{p}(H_1) = 0.3685$ $p(H_1/I) = 0.2990$	$\tilde{p}(I/H_2) = 0.8845$ $\tilde{p}(H_2) = 0.6163$ $p(\mathbf{H}_2/I) = \mathbf{0.5608}$	$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1392$















			
Image 23	$\tilde{p}(I/H_1) = 0.8382$	$\tilde{p}(I/H_2) = 0.9034$	$\tilde{p}(I/H_3) = 0.3679$
	$\tilde{p}(H_1) = 0.3722$	$\tilde{p}(H_2) = 0.3877$	$\tilde{p}(H_3) = 0.3679$
	$p(H_1/I) = 0.3911$	$p(\mathbf{H}_2/I) = \mathbf{0.4392}$	$p(H_3/I) = 0.1697$
			
Image 24	$\tilde{p}(I/H_1) = 0.8392$	$\tilde{p}(I/H_2) = 0.7695$	$\tilde{p}(I/H_3) = 0.3679$
	$\tilde{p}(H_1) = 0.4028$	$\tilde{p}(H_2) = 0.7417$	$\tilde{p}(H_3) = 0.7414$
	$\mathbf{p}(\mathbf{H}_1/I) = \mathbf{0.4467}$	$p(H_2/I) = 0.3742$	$p(H_3/I) = 0.1788$

Table 9.7: Experiments with real images from Martin et al. (2001).

We present in Table 9.8 two experiments with real images from Martin et al. (2001) dataset where there is an ambiguity in the depth ordering (there are conflicting local depth cues). This situation can appear when the proximal image is made of objects that are not fronto-parallel to the camera or when their relative order changes due to, for example, mutual occlusions as in these examples. In other words, an object does not appear at a single depth layer. In this situation our algorithm chooses the object that is more occluded as being behind but let us remark how the posterior probabilities of the two first hypothesis are very close; in fact, these two hypothesis correspond to the two different depth orderings indicated by the local depth cues and figure/ground ground-truth labels superimposed on the original image.

			
Image 25	$\tilde{p}(I/H_1) = 0.7757$	$\tilde{p}(I/H_2) = 0.7942$	$\tilde{p}(I/H_3) = 0.3679$
	$\tilde{p}(H_1) = 0.3709$	$\tilde{p}(H_2) = 0.3763$	$\tilde{p}(H_3) = 0.3679$
	$p(H_1/I) = 0.3986$	$p(H_2/I) = 0.5140$	$p(H_3/I) = 0.1875$









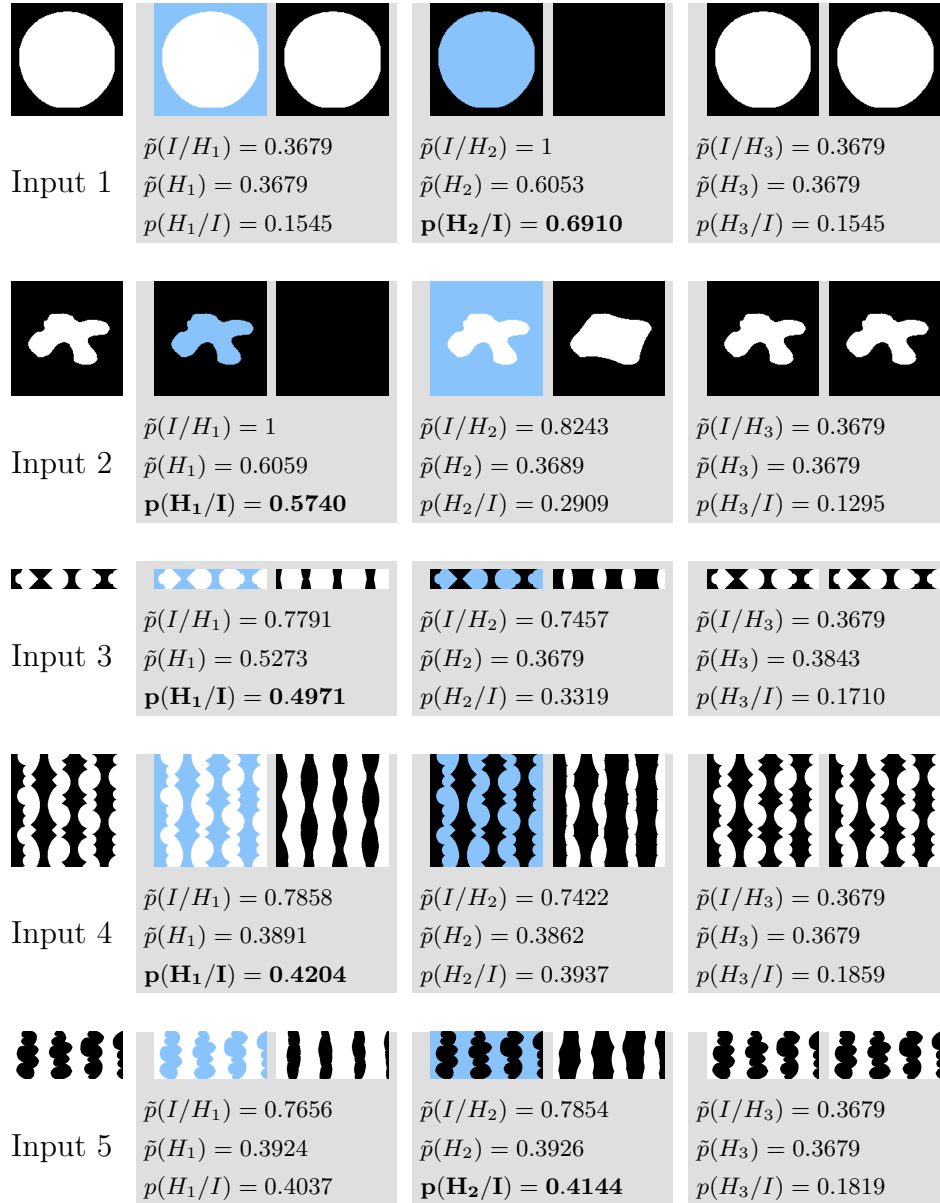
							
Image 26	$\tilde{p}(I/H_1) = 0.9787$ $\tilde{p}(H_1) = 0.3689$ $p(H_1/I) = 0.4401$		$\tilde{p}(I/H_2) = 0.8768$ $\tilde{p}(H_2) = 0.3695$ $p(H_2/I) = 0.3950$		$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1650$		

Table 9.8: Two experiments with real images from Martin et al. (2001) where there is an ambiguity in the depth ordering.

9.3 Shapes in front of a background

Finally, we present some results showing the ability of our method to also decide on (perceptually) fully visible objects over a background. Table 9.9 displays several synthetic images of this type, where there are no T-junctions present and, according to human perception, the depth ordering is established by convexity cues as observed by Kanizsa (1991). In these experiments, our method fails in Results 3, 4 and 5; in all of them the convexity cue is a stronger depth cue than symmetry, and the algorithm we are using for computing shape complexity favours symmetries. Let us also remark that Result 9 allows both interpretations: black in front of white and white in front of black, as they form the same shape but with different orientation. In this case our algorithm prefers H_1 but with a small difference with respect to H_2 .












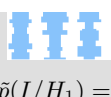
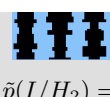
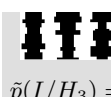

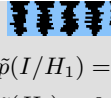
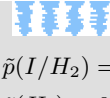
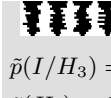
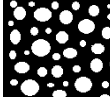
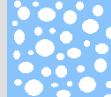
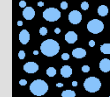
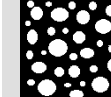
 Input 6	 $\tilde{p}(I/H_1) = 1$ $\tilde{p}(H_1) = 0.5870$ $\mathbf{p}(H_1/I) = \mathbf{0.5211}$	 $\tilde{p}(I/H_2) = 0.9508$ $\tilde{p}(H_2) = 0.4250$ $p(H_2/I) = 0.3588$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1201$
 Input 7	 $\tilde{p}(I/H_1) = 0.9093$ $\tilde{p}(H_1) = 0.3746$ $p(H_1/I) = 0.4023$	 $\tilde{p}(I/H_2) = 0.7799$ $\tilde{p}(H_2) = 0.4755$ $\mathbf{p}(H_2/I) = \mathbf{0.4379}$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.1598$
 Input 8	 $\tilde{p}(I/H_1) = 1$ $\tilde{p}(H_1) = 0.6961$ $\mathbf{p}(H_1/I) = \mathbf{0.6164}$	 $\tilde{p}(I/H_2) = 0.7191$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.2343$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.4584$ $p(H_3/I) = 0.1493$
 Input 9	 $\tilde{p}(I/H_1) = 0.7790$ $\tilde{p}(H_1) = 0.3812$ $\mathbf{p}(H_1/I) = \mathbf{0.3812}$	 $\tilde{p}(I/H_2) = 0.7438$ $\tilde{p}(H_2) = 0.3679$ $p(H_2/I) = 0.3679$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.4757$ $p(H_3/I) = 0.2347$
 Input 10	 $\tilde{p}(I/H_1) = 0.9966$ $\tilde{p}(H_1) = 0.3679$ $p(H_1/I) = 0.4055$	 $\tilde{p}(I/H_2) = 1$ $\tilde{p}(H_2) = 0.3977$ $\mathbf{p}(H_2/I) = \mathbf{0.4398}$	 $\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3803$ $p(H_3/I) = 0.1547$

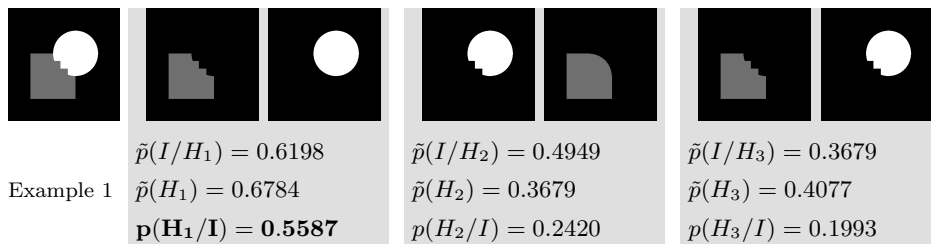
Table 9.9: Synthetic experiments with a shape in front of a background.

9.4 Discussion of failure cases

This section is devoted to present and discuss situations where our method can fail.

Let us first comment on the experiments on synthetic images. Example 1 and Example 2 of Table 9.10 do not agree with human perception: in a T-junction, the occluder is the surface on the T-head side while the surfaces on the T-stem side continue behind the occluder. However, in these two results, the local occlusion signatures given by the T-junctions indicate that there is an occluded square which continues behind an incomplete disc (the occluder). Taking this into account, our method fails to give the hypothesis that agrees with the T-junction cues (which should be H_2). In Example 1, the likelihood of the hypothesis H_1 and H_2 are similar but the global complexity of the shapes in H_1 is smaller (thus higher prior) than the global complexity of the shape in H_2 . In particular, the two shapes present in H_1 are jointly simpler than those in H_2 or H_3 . Regarding Example 2, the highly irregular contour of the shapes makes difficult a straightforward analysis and the final chosen hypothesis is due to a balance among the corresponding complexities and likelihoods.

In the examples of Table 9.11, according to the local cues given by the T-junctions, the preferred option should always be H_1 . However, our method obtains H_2 . In Example 3, although a higher prior due to a smaller complexity of the objects in H_1 , the likelihood of H_2 is higher due to smaller elastica values in H_2 . Example 4 is the opposite: H_1 presents a higher complexity (thus smaller prior) and a higher likelihood.



Example 2	$\tilde{p}(I/H_1) = 0.6948$ $\tilde{p}(H_1) = 0.3876$ $\mathbf{p(H_1/I) = 0.4026}$		$\tilde{p}(I/H_2) = 0.6535$ $\tilde{p}(H_2) = 0.4043$ $p(H_2/I) = 0.3951$		$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.2023$	

Table 9.10: Synthetic experiments that fail.

Example 3	$\tilde{p}(I/H_1) = 0.3679$ $\tilde{p}(H_1) = 0.3770$ $p(H_1/I) = 0.2951$		$\tilde{p}(I/H_2) = 0.4501$ $\tilde{p}(H_2) = 0.3679$ $\mathbf{p(H_2/I) = 0.3524}$		
Example 4	$\tilde{p}(I/H_1) = 0.4062$ $\tilde{p}(H_1) = 0.3679$ $p(H_1/I) = 0.2853$		$\tilde{p}(I/H_2) = 0.3679$ $\tilde{p}(H_2) = 0.5089$ $p(H_2/I) = \mathbf{0.3574}$		

Table 9.11: Synthetic experiments that fail.

Example 5	$\tilde{p}(I/H_1) = 0.6858$ $\tilde{p}(H_1) = 0.3726$ $\mathbf{p(H_1/I) = 0.3962}$		$\tilde{p}(I/H_2) = 0.03726$ $\tilde{p}(H_2) = 0.3704$ $p(H_2/I) = 0.3939$		$\tilde{p}(I/H_3) = 0.3679$ $\tilde{p}(H_3) = 0.3679$ $p(H_3/I) = 0.2099$		

Example 6	$\tilde{p}(I/H_1) = 0.7469$		$\tilde{p}(I/H_2) = 0.6515$		$\tilde{p}(I/H_3) = 0.3679$		
	$\tilde{p}(H_1) = 0.3686$		$\tilde{p}(H_2) = 0.3695$		$\tilde{p}(H_3) = 0.3679$		
	$\mathbf{p}(H_1/I) = \mathbf{0.4227}$		$p(H_2/I) = 0.3696$		$p(H_3/I) = 0.2078$		
Example 7	$\tilde{p}(I/H_1) = 0.8575$		$\tilde{p}(I/H_2) = 0.7297$		$\tilde{p}(I/H_3) = 0.3679$		
	$\tilde{p}(H_1) = 0.3709$		$\tilde{p}(H_2) = 0.5332$		$\tilde{p}(H_3) = 0.3679$		
	$p(H_1/I) = 0.3709$		$\mathbf{p}(H_2/I) = \mathbf{0.5332}$		$p(H_3/I) = 0.3679$		
Example 8	$\tilde{p}(I/H_1) = 0.8732$		$\tilde{p}(I/H_2) = 0.8325$		$\tilde{p}(I/H_3) = 0.3679$		
	$\tilde{p}(H_1) = 0.3679$		$\tilde{p}(H_2) = 0.3792$		$\tilde{p}(H_3) = 0.3679$		
	$p(H_1/I) = 0.4115$		$\mathbf{p}(H_2/I) = \mathbf{0.4118}$		$p(H_3/I) = 0.1767$		
Example 9	$\tilde{p}(I/H_1) = 0.8744$		$\tilde{p}(I/H_2) = 0.9118$		$\tilde{p}(I/H_3) = 0.3679$		
	$\tilde{p}(H_1) = 0.3763$		$\tilde{p}(H_2) = 0.3747$		$\tilde{p}(H_3) = 0.3679$		
	$p(H_1/I) = 0.4082$		$\mathbf{p}(H_2/I) = \mathbf{0.4239}$		$p(H_3/I) = 0.1679$		
Example 10	$\tilde{p}(I/H_1) = 1$		$\tilde{p}(I/H_2) = 0.8920$		$\tilde{p}(I/H_3) = 0.3679$		
	$\tilde{p}(H_1) = 0.3679$		$\tilde{p}(H_2) = 0.4117$		$\tilde{p}(H_3) = 0.3758$		
	$\mathbf{p}(H_1/I) = \mathbf{0.4213}$		$p(H_2/I) = 0.4205$		$p(H_3/I) = 0.1583$		









								
Example 11								

Table 9.12: Experiments with real images that fail.

Let us comment on the images of Table 9.12. Examples 5 and 6 reflect the same situation; the inpainting method is unable to recover the leg of the older horse or sheep. In any case, the difference among the posterior probabilities of the first two hypothesis is very small. On the other hand, in Examples 7 and 8, although according to the likelihood the preferred hypothesis is the correct one (e.g., two ladybugs in front of two flowers in Example 7), the complexity of the objects in the second hypothesis (flowers in front of ladybugs) is smaller (higher prior) because of the simplified completed object and this second hypothesis wins. In Example 9, the prior probabilities of H_1 and H_2 are similar but the likelihood of H_2 is slightly higher. Finally, Examples 10 and 11 show the same situation, where there appears a window showing the sky, which is behind. Our method fails in these cases, which are interpreted as small convex shapes over a biggest shape which is behind. Another example would be the arches of a bridge, which are further away compared to the bridge itself, which would be interpreted as closest by our method.

10

Conclusions and Future Work

We have proposed a computational model of amodal completion that allows to compute the most preferred scene structure given a still image of it. As we are considering scenes where objects appear at two different depths, we take into account the three possible hypothesis. Our main contribution is a Bayesian probabilistic model based on the Euler's elastica and the global complexity of the hypothesized objects in order to choose the most preferred explanation of the image. This explanation includes both the disoccluded objects that form the scene and their ordering according to depth. Furthermore, we have proposed a disocclusion method, to compute the hypothesized objects, based on human visual completion, which is modeled by a binary inpainting method based again on the Euler's elastica and that takes into account perceptual findings related to amodal completion, such as relatability, convexity, and good continuation. Finally, we have shown the capability of our method with numerical experiments, both with real and synthetic images.

As future work, we plan to extend the approach to scenes with more than two depth layers. Furthermore, we plan to incorporate other disocclusion strategies (such as, e.g., exemplar-based methods (Aujol et al. (2010); Arias et al. (2011) or Hayashi and Sasaki (2014)) allowing to model global completions taking into account properties such as symmetries or repetitions. Last but not least, we are also interested in the extension of the model to video sequences.

PART III:
VIDEO INPAINTING

Imagination is more important
than knowledge.

ALBERT EINSTEIN

11

Introduction

This Chapter explains the video and optical flow inpainting problems. We also present a review of the work done on these topics and give a summary of our model.

Video inpainting stands for the completion of missing, damaged or occluded information in a video sequence or a still image in such a way that this restoration is as unnoticeable (visually plausible) as possible and the result looks natural. The applications include tools for cinema post-production to remove, e.g., unwanted or private items, or tools for the recovering of occluded areas in new-view generation for 3D television or broadcasting of sport events, to mention a few.

Shih et al. (2009) show that we can not apply image inpainting techniques to each frame separately due to the fact that the temporal incoherence from frame to frame is very noticeable for the human vision system, producing an undesirable flickering effect. Consequently, video inpainting brings additional challenges to the ones of image inpainting not only in order to obtain temporally coherent results but also due to the occlusions and disocclusions among objects that move along time. Object occlusions and disocclusions generate artifacts which are specially visible at moving occlusion boundaries. Moreover, the estimation of the vector field that recovers the apparent movement of pixels between two consecutive frames, i. e. the optical flow, may fail in occlusion areas due to the impossibility of point matching. In general, points visible at time t that are occluded at

time $t + 1$ do not have a corresponding point at frame $t + 1$. While points that appear at time $t + 1$ have no corresponding point at time t . Thus video completion algorithms have to detect such occlusions in order to correctly decide how to treat them.

Most of the existing video inpainting algorithms are exemplar-based methods, algorithms that exploit the non-local self-similarities present in natural images and videos and are based on the assumption that the information necessary to complete the missing part is available elsewhere in the image or video. Pioneer works are attributed to Patwardhan et al. (2007) and Wexler et al. (2007). Patwardhan et al. (2007) propose an extension to the video case of the exemplar-based image inpainting method proposed by Criminisi et al. (2004). The method proposed by Patwardhan et al. (2007) separates the inpainting of moving foreground and static background, and uses a priority-based scheme for copying patches. Wexler et al. (2007) extended the texture synthesis approach of Efros and Leung (1999) by introducing an objective function, based on the coherence of the completed video. Their approach is based on the assumption that all space-time patches intersecting the missing region are presented somewhere in the unoccluded region. Both proposals are rather limited to static background and to restricted foreground and camera motion (e.g., static camera and cyclic motion without changing in size). The method of Wexler et al. (2007) was generalized to dynamic background by Newson et al. (2014), who extended the PatchMatch search scheme (Barnes et al. (2009)) to the spatio-temporal domain which reduces the time complexity of the algorithm. The method proposed in Bugeau et al. (2010) applies image inpainting independently to each frame and then temporal consistency is imposed by Kalman filtering along the estimated trajectories. Granados et al. (2012b) proposed an energy-based method with a graph-cut-based optimization to deal with dynamic background and non-periodical moving objects. However, it is limited to the static camera case. The video completion method proposed by Ebdelli et al. (2015) starts by aligning a set of neighbouring frames via a region-based homography

(see also Granados et al. (2012a)); then an energy function defined on the registered frames and based on both spatial and temporal regularity is globally minimized. Instead of aligning the video frames, as in most of the preceding cases, two recent works impose the temporal coherence thanks to the previously inpainted optical flow (Strobel et al. (2014); Xu et al. (2016b)). In particular, Xu et al. (2016b) pose the completion process as an MRF-based optimization problem where candidates for the occluded pixels are given by the motion field correspondences in neighbouring frames. On the other hand, Strobel et al. (2014) propose an exemplar-based method where the patch distance function introduced in Criminisi et al. (2004) is modified in such a way that takes temporal consistency into account thanks to the completed motion field.

We propose a variational method for binary video inpainting, with the goal of recovering the dynamic shape with a smooth surface, that works directly in the spatio-temporal dimension (3D). Binary inpainting tools fall into the category of geometry-oriented methods that aim at recovering shapes, stated as binary objects. They might be combined, in a two-step algorithm or jointly, in a model for the joint estimation of shape geometry and texture inpainting. The completed shape may help to guide the correspondence map or copy of the patches: inside the shape of interest only patches from the same object are allowed to be copied and similarly for the background as observed in Cao et al. (2011). On the other hand, binary inpainting represents a tool for the understanding of a moving scene through decomposition of it in complete and isolated moving objects interacting among them.

To evolve shapes according to the minimization of a geometric functional, based either on the length area or the curvature of the shape contours, it has been used the threshold Dynamics strategy, which was introduced by Merriman et al. (1992) as a method to move shapes by mean curvature motion. By changing the geometric functional it can be used to solve image problems such as shape recovery (Jawerth and Lin (2002)), shape disocclusion (Esedoglu et al.

(2005); Bertozzi et al. (2007)), moving shapes with another motion (Grzhibovskis and Heintz (2008); Esedoglu et al. (2008)), segmentation (Esedoglu et al. (2006); Calatroni et al. (2017); Bertozzi and Flenner (2012)) or Point Clouds (Thorpe and Theil (2016)). Afterwards, Rubinstein et al. (1989) proved that it solves the Allen-Cahn equation, which is the gradient descent equation of the Grinzburg-Landau functional (a geometric functional that contains a double well potential).

To recover the video shapes we propose the minimization of an energy functional that imposes not only spatial regularity but also temporal continuity along the visible trajectory of the object, which is defined by the optical flow. As previously done by Bhat et al. (2007, 2010); Facciolo et al. (2011) or Sadek et al. (2012), we impose the temporal continuity by using the convective derivative, a motion-compensated temporal derivative. The motion field can be estimated outside the inpainting mask (or hole with missing information) with any of the existing optical flow methods. On the other hand, the optical flow is unknown inside the hole and it is geodesically interpolated (with a motion inpainting method) in order to guide the inpainting process.

Optical flow or motion inpainting is a pervasive problem in many areas of computer vision which range from semantic video analysis to video editing. One of the capital difficulties of the optical flow estimation are the occlusions where its estimation becomes extremely difficult due to the lack of correspondence between the two consecutive frames. Therefore, the optical flow is partially missing in areas more or less large of the video whereas for most of the applications its completion is essential. For instance, in cinema post-production a completed optical flow is often needed for the elimination of unwanted (moving or static) objects. Other applications are automatic assistance of sensor-based optical flow estimation where the sensor acquisition usually produces large regions without optical flow data, as in the Kitti Vision benchmark proposed by Menze and Geiger (2015)).

The motion inpainting problem has been addressed previously in the literature for different purposes. In order to inpaint the flow in the occlusion areas, Matsushita et al. (2006) and Strobel et al. (2014) extend the Telea (2004) inpainting idea to optical flow, i.e., they assume that the motion variation is locally small and propagate the optical flow according to a weighting function which depends on the Euclidean distance and the color difference among the interpolated pixel and its neighbours. Kondermann et al. (2008) propose a postprocess of the optical flow in order to improve it: they retain the optical flow at points where it is reliable and then they densify it by minimizing the L^2 norm of the spatio-temporal gradient of the flow. Berkels et al. (2009) propose to recover the optical flow in non-reliable regions by regularization, in particular they use a TV-type anisotropic functional and Palomares et al. (2014) propose a rotation-invariant regularizer. On the other hand, Ince and Konrad (2008) propose a variational method for the joint estimation of optical flow and occlusions while extrapolating the optical flow in occlusion areas by means of anisotropic diffusion based on the image gradients. Leordeanu et al. (2013) and Revaud et al. (2015) propose a sparse-to-dense optical flow estimation method that takes as input an initial set of sparse matches. In a first stage, the flow is densified (completed) by fitting a local affine model, that uses edge-aware distances in the case of EpicFlow. Then, the densified flow is refined by minimizing an optical flow energy functional.

We propose to use the Absolutely Minimizing Lipschitz Extension (AMLE) model in a Riemannian manifold in order to take advantage of the geometric information given by the video frames. Given a video and an incomplete motion field, we endow each 2D frame domain with a Riemannian metric based on the video values and propose to recover the missing optical flow by solving the AMLE partial differential equation on the 2D Riemannian manifold from the known values on the boundary of the interpolation domain, which may contain isolated points. Each of the coordinates of the optical flow is thus reconstructed with this metric-based anisotropic interpolation. The

AMLE equation, $\Delta_\infty u = 0$, where $\Delta_\infty u$ is called both, AMLE operator and infinity Laplacian, was introduced by Aronsson (1967, 1968) and uniqueness of viscosity solutions was proved in Jensen (1993) (see also Aronsson et al. (2004) for a review). The AMLE allows to interpolate the value on isolated points and curves. It appeared as one of the interpolating operators satisfying a set of suitable axioms in Caselles et al. (1998). This axiomatic approach was extended by Sander et al. (2003) and Caselles et al. (2006) to interpolate data given on a set of curves on a surface in \mathbb{R}^3 . The AMLE on a manifold was applied in Lazcano (2016) for interpolating depth data in images or videos where large regions of incomplete depth information often appear. On the other hand, the classical AMLE has been used in Almansa et al. for the interpolation of digital elevation models.

Our method is applied in three different scenarios: optical flow inpainting in large regions, the completion of the motion in occlusion areas and densification of an optical flow from a sparse set of initial matches. The experiments show that in general our results outperform those of EpicFlow proposed by Revaud et al. (2015), which has become a reference method for optical flow estimation and a standard technique for post-processing an estimated and filtered optical flow.

12 Video Shape Inpainting

In this Chapter we present a variational model for video shape inpainting, which uses a differential operator based on a generalized 3D gradient with the convective derivative in time and the usual gradient in space. Its optimization grows on a threshold dynamics strategy. Finally we provide some results.

We consider a spatio-temporal domain $\mathcal{V} = \{(\mathbf{x}, t) : \mathbf{x} = (x, y) \in \Omega, t \in [0, T], T > 0\}$ where in some big regions $\mathcal{M} = \{(\mathbf{x}, t) : \mathbf{x} = (x, y) \in \Omega, t \in (0, T), T > 0\}$, called holes, the information is missing. We have defined the image domain (i.e., the spatial domain of any image frame at time t) as $\Omega \subset \mathbb{R}^2$, which is a rectangle in \mathbb{R}^2 . Let $u_0(\mathbf{x}, t)$ be a binary video sequence defined on $\mathcal{V} \setminus \mathcal{M}$ and \mathbf{v} the optical flow associated to u_0 defined on \mathcal{V} . Observe that, in this Chapter, the optical flow is considered in the whole domain, therefore we assume that we have at our disposal an optical flow estimation method as well as an optical flow inpainting method.

In order to inpaint the binary video inside the inpainting mask $\mathcal{M} \subset \mathcal{V}$ we propose to solve the following optimization problem

$$\min_{u: \mathcal{V} \rightarrow \{0,1\}} \int_{\mathcal{M}} \|\mathcal{L}(u)\|^2, \quad \text{s.t. } u = u_0 \text{ in } \mathcal{V} \setminus \mathcal{M} \quad (12.1)$$

where $\mathcal{L}(u)$ is defined taking into account both spatial and temporal regularity as well as the occlusion areas produced by the motion of

objects in the scene:

$$\mathcal{L}(u) = (u_x, u_y, \gamma\chi\partial_{\mathbf{v}}u), \quad (12.2)$$

where $\gamma > 0$ is a parameter and $\chi : \mathcal{V} \rightarrow [0, 1]$ is a function modeling the occlusion areas so that $\chi(\mathbf{x}, t) = 0$ identifies the occluded pixels, i.e. pixels that are visible at time t but not at time $t + 1$. Thus, $\chi(\mathbf{x}, t) = 1$ identifies the non occluded pixels and the functional only imposes temporal regularity along the pixel trajectories that are not occluded. The convective derivative is defined as

$$\partial_{\mathbf{v}}u(\mathbf{x}, t) = \nabla u(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) + \frac{\partial u}{\partial t}(\mathbf{x}, t). \quad (12.3)$$

Let us recall how it naturally appears: From the assumption that for a Lambertian object¹ under uniform constant illumination the brightness of an object's particle does not change in time, one deduces that $u(\mathbf{x}(t), t)$ is constant along trajectories of the points in the scene. This implies that

$$0 = \frac{du}{dt}(\mathbf{x}(t), t) = \nabla u(\mathbf{x}, t) \cdot \frac{d\mathbf{x}(t)}{dt} + \frac{\partial u}{\partial t}(\mathbf{x}, t) \approx \partial_{\mathbf{v}}u(\mathbf{x}, t) \quad (12.4)$$

since $\mathbf{v} \approx d\mathbf{x}(t)/dt$. It leads to the brightness constancy assumption, introduced in Horn and Schunck (1981). By minimizing the convective derivative in (12.1) we are imposing shape regularity along the trajectories. Moreover, since we do not consider the convective derivative for occluded pixels we are imposing the regularity only along the visible trajectories. Finally, by minimizing the L^2 -norm of the first two terms of the operator (12.2) we are imposing spatial smoothness in the recovered shape.

In our proposal (12.1)-(12.2), the parameter γ accounts for the different units in the spatial and temporal domains and also balances the effect of the temporal diffusion in the resulting gradient descent

¹A Lambertian object is a surface where the apparent brightness to an observer is the same regardless of the observer's angle of view (Koppal (2014)).

equation. Observe that, when γ is big enough, the minimization of (12.1) can be approximated by using, instead of \mathcal{L} , the operator

$$\tilde{\mathcal{L}}(u) = \chi \partial_{\mathbf{v}} u \quad (12.5)$$

as the spatial derivatives have almost no impact. However, as the experiments in Section 12.2 show, it is necessary to consider the dynamic shape evolution of the 3D shape (space and time) to correctly complete the moving objects.

Optical Flow estimation

To fully specify this method, one needs to provide an estimation of the optical flow. We propose to use, in the original sequence, a variational optical flow estimation method and the optimization strategy for variational optical flow methods proposed in Palomares et al. (2017), which can be applied to any energy. In particular, we apply it with both the well-known TV-L1 energy functional of Zach et al. (2007) and the NLTV-CSAD. The NLTV-CSAD energy functional uses Non Local Total Variation as regularization term as suggested by Werlberger et al. (2010) and a smooth variant of the Census Transform proposed by Vogel et al. (2013) as data term. To show the robustness of our binary inpainting method with respect to the optical flow, we present in Fig. 12.1 two sequences of frames showing that similar results are obtained using as input either the ground-truth optical flow or the estimated ones. In fact, on the first row we show the sequences to be inpainted (with the mask in gray), on the second row we present the result using the ground-truth optical flow, on the third row we show the inpainted result using the NLTV-CSAD optical flow and, finally, on the last row we present the results using the TV-L1 optical flow.

As a consequence of removing objects from the sequence, we need to modify the optical flow in the area of the removed object (which constitutes the inpainting mask or hole). So, we need to do motion inpainting inside the spatio-temporal 3D hole. We propose to use the

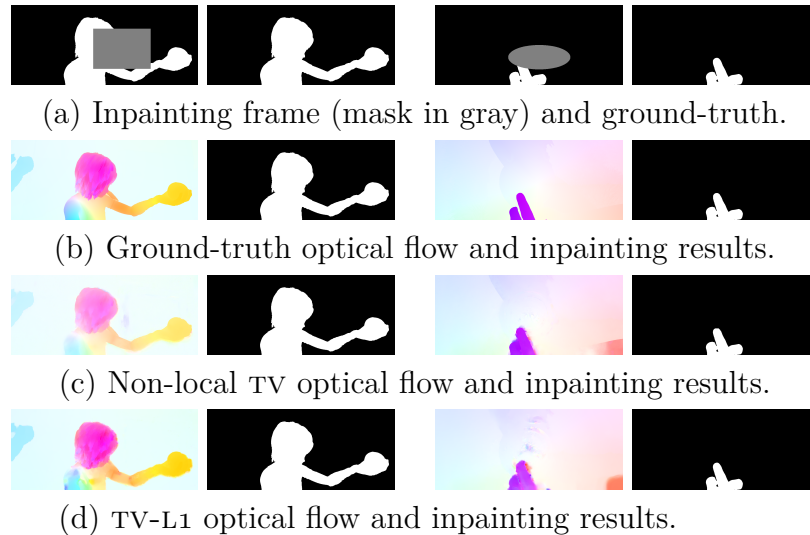


Figure 12.1: Comparison of the inpainted result obtained with optical flow ground-truth, NLTV-CSAD or TV-L1.

optical flow inpainting method proposed in Palomares et al. (2014), although other methods exist in the literature, such as Kondermann et al. (2008) and Matsushita et al. (2006).

In order to correctly fill-in the optical flow we dilate the hole to avoid the irregularities of the optical flow close to its boundary. Indeed, the optical flow estimated by variational methods is not accurate at motion boundaries.

Occlusion estimation

In order to estimate motion occlusions we stem from the assumption that the occluded region, given in our context by $\chi(x, y) = 0$, can be correlated with the region where the divergence of the optical flow is negative. This was pointed out by Sand and Teller (2008), who noticed that the divergence of the motion field may be used to distinguish between different types of motion areas. Schematically,

the divergence of a flow field is negative for occluded areas, positive for disoccluded, and near zero for the matched areas (an example is presented in Figure 12.2). In our method, we relax this criteria and use the estimation of occluded ($\chi = 0$) and visible ($\chi = 1$) regions as

$$\chi(\mathbf{x}, t) = \begin{cases} 1, & \text{div}(\mathbf{v}) \geq -0.5 \\ 0, & \text{else.} \end{cases} \quad (12.6)$$



(a) Frame 13.



(b) Frame 14.

(c) $\chi(\mathbf{x}, t)$.

Figure 12.2: Visible areas (in white) among the two frames.

12.1 A Threshold Dynamics Strategy

Our problem has been defined for binary functions. To overcome the nondifferentiability and the nonconvexity, a common relaxation strategy is given by not restricting the solution to be binary and using instead a double-well potential² in the functional. To this goal, we propose to rewrite our energy (12.1) in terms of a Ginzburg-Landau

²A double-well potential is an energy with two, degenerate or not, minima separated by a maximum.

type functional. Let us recall that it consists of two terms, the first is a regularization of the derivative and the second is a penalization that forces the functional to be close to binary:

$$\varepsilon \int_{\Omega} |\nabla u(x)|^2 dx + \frac{1}{\varepsilon} \int_{\Omega} W(u(x)) dx, \quad (12.7)$$

where $\varepsilon > 0$ and $W : \mathbb{R} \rightarrow \mathbb{R}$ is a double-well potential. The penalization term forces u to be in one of the minima, while the regularization term forces u to have some smoothness. It has been proved to Γ -converge to the total variation functional on binary functions by Modica and Mortola (1977).

The L^2 gradient descent equation of the Grinzburg-Landau functional is the Allen-Cahn equation (Allen and Cahn (1979)):

$$u_s = 2\varepsilon \Delta u - \frac{1}{\varepsilon} W'(u). \quad (12.8)$$

Rubinstein et al. (1989) proved that when $\varepsilon \rightarrow 0$ the rescaled solutions of the Allen-Cahn equation, $u_\varepsilon(x, \frac{t}{\varepsilon})$, evolve according to mean curvature flow of the interface. Equation (12.8) can be solved using the threshold dynamics algorithm (MBO algorithm), proposed by Merriman et al. (1992) which, as proved by Evans (1993) and Barles and Georgelin (1995), simulates the motion of a 2D binary shape by mean curvature motion.

MBO algorithm starts with an initial shape $\mathcal{S}^0 \subset \mathbb{R}^2$ and, by considering its indicator or characteristic function $u^0 = \mathbb{1}_{\mathcal{S}^0}$, iterates the following two steps until convergence:

1. Diffusion step. Compute $\bar{u}(\tau)$, the solution of the heat equation, $u_s = \Delta u$, for a certain small diffusion time τ , with initial condition $u(0) = \mathbb{1}_{\mathcal{S}^n}$.
2. Thresholding step. Binarize by defining the shape $\mathcal{S}^{n+1} = \{\mathbf{x} : \bar{u}(\tau)(\mathbf{x}) \geq \frac{1}{2}\}$

As a consequence, the MBO scheme solves the Allen-Cahn equation by time splitting: step 1 solves $u_s = 2\varepsilon \Delta u$ and step 2 solves $u_s =$

$\frac{1}{\varepsilon}W'(u)$ for $\varepsilon \rightarrow 0^+$, both for a fixed time $\tau > 0$. Observe that the wells of the potential will define the thresholding step of the scheme, for instance, in the case of the MBO algorithm it is used a double-well potential with $x_0 = 0$ and $x_1 = 1$.

Our functional (12.1) can be thought as a generalization of (12.7), as it involves the gradient in the spatial dimensions and the convective derivative (a derivative taken with respect to a moving coordinate system) in the temporal one. Therefore, we can rewrite the energy (12.1) as:

$$\begin{aligned} & \varepsilon \int_{\mathcal{M}} \|(u_x, u_y, \gamma\chi\partial_{\mathbf{v}}u)\|^2 dx + \frac{1}{\varepsilon} \int_{\Omega} W(u) dx \\ & \text{s.t. } u = u_0 \text{ in } \mathcal{V} \setminus \mathcal{M} \end{aligned} \quad (12.9)$$

where $W(u) = u^2(1 - u)^2$. Its gradient descent equation is:

$$u_s = 2\varepsilon \left(\Delta_{xy}u - \gamma^2(\chi\partial_{\mathbf{v}})^*\chi\partial_{\mathbf{v}}u \right) - \frac{1}{\varepsilon}W'(u), \quad (12.10)$$

where Δ_{xy} denotes the Laplacian on the spatial dimension and $(\chi\partial_{\mathbf{v}})^*$ the adjoint operator of $\chi\partial_{\mathbf{v}}$.

We propose to solve the boundary value problem associated to the PDE (12.10) by time splitting in such a way that one of the resulting equations, $u_s = -\frac{1}{\varepsilon}W'(u)$, is an ordinary differential equation that is solved by a thresholding step, as in the MBO scheme (Merriman et al. (1992)). Then, starting by an initial spatio-temporal shape \mathcal{T}^0 and, considering its (binary) characteristic function $u^0 = \mathbb{1}_{\mathcal{T}^0}$, the core of the threshold dynamics scheme that we propose consists on the iteration of the following steps until convergence:

1. Diffusion step. Compute $\bar{u}(\tau)$, the solution of the following PDE for a certain small diffusion time τ :

$$\begin{aligned} u_s &= \Delta_{xy}u + \gamma^2(\chi\partial_{\mathbf{v}})^*\chi\partial_{\mathbf{v}}u \\ u(0) &= \mathbb{1}_{\mathcal{T}^n}. \end{aligned} \quad (12.11)$$

2. Thresholding step. Binarize by defining the following shape:

$$\mathcal{T} = \left\{ \mathbf{x} : \bar{u}(\tau)(\mathbf{x}) \geq \frac{1}{2} \right\}. \quad (12.12)$$

3. Fidelity step. Imposes that the binary video coincides with the original video outside of the inpainting mask.

$$\mathcal{T}^{n+1} = (\mathcal{T} \cap \mathcal{M}) \cup (\mathcal{T}^0 \cap (\mathcal{V} \setminus \mathcal{M})). \quad (12.13)$$

12.1.1 Numerical Details on the Diffusion Step

We consider a discrete video obtained by regularly sampling the continuous one with a spatial step h and a temporal step k . Then, we use an explicit method with a forward Euler discretization of the temporal derivative to solve the PDE (12.11) involved in the first step of the Threshold Dynamics algorithm:

$$\frac{u^{n+1} - u^n}{k} = \frac{\Delta_{xy}u^n - \gamma^2(\chi\partial_{\mathbf{v}})^*\chi\partial_{\mathbf{v}}u^n}{h}. \quad (12.14)$$

The equation involves a spatial diffusion term, namely $\Delta_{xy}u$, and a temporal diffusion step, given by $(\chi\partial_{\mathbf{v}})^*\chi\partial_{\mathbf{v}}u$. In the temporal step there are also considered the occlusions among frames.

The spatial term is discretized using a finite differences scheme, while the temporal term is performed with a sparse matrix, in this way we only need to compute $A = \chi\partial_{\mathbf{v}}u$ and the conjugate is obtained by the transposed of it. The scheme turns into:

$$u^{n+1} - u^n = \frac{k}{h} \left(\Delta_{xy}u - \gamma^2 A^t A u \right). \quad (12.15)$$

Discretization of Δ

The spatial Laplacian is implemented using the forward and backward finite differences scheme proposed by Chambolle (2004), where it is proposed to solve it using the following equality:

$$\Delta u = \text{div}(\nabla u). \quad (12.16)$$

The gradient is computed using a forward scheme:

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \quad (12.17)$$

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \quad (12.18)$$

and the divergence is discretized with a backwards scheme:

$$\begin{aligned} (\operatorname{div} \mathbf{p})_{i,j} = & \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{if } 1 < i < N \\ p_{i,j}^1 & \text{if } i = 1 \\ -p_{i-1,j}^1 & \text{if } i = N \end{cases} \\ & + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{if } 1 < j < N \\ p_{i,j}^2 & \text{if } j = 1 \\ -p_{i,j-1}^2 & \text{if } j = N \end{cases} \end{aligned} \quad (12.19)$$

where $\mathbf{p} = ((\nabla u)^1, (\nabla u)^2)$.

Discretization of $\chi \partial_{\mathbf{v}}$

The convective derivative is discretized using a forward difference scheme, which takes into account the value of the optical flow:

$$\partial_{\mathbf{v}} u(x_0, y_0, t_0) = \hat{u}(\mathbf{x}_0 + k\mathbf{v}, t_0 + k) - u(\mathbf{x}_0, t_0) \quad (12.20)$$

where \hat{u} is an interpolated value and k is the temporal step. Let us observe that, because of the values of the optical flow, $\mathbf{x}_0 + k\mathbf{v}$ may fall outside the sampling grid. As proposed by Zhou et al. (1998) and Arias (2013), we obtain $\hat{u}(\cdot)$ using the following interpolation:

$$\begin{aligned} \hat{u}(\mathbf{x}_0 + k\mathbf{v}) = & k(\tilde{v}_1 u_{i,j} + (1 - \tilde{v}_1) u_{i+1,j}) \tilde{v}_2 + \\ & k(\tilde{v}_1 u_{i,j+1} + (1 - \tilde{v}_1) u_{i+1,j+1})(1 - \tilde{v}_2), \end{aligned} \quad (12.21)$$

where \tilde{v}_1 and \tilde{v}_2 denote the fractional part of the x and y components of the optical flow, respectively. Given the interpolation values

we create a matrix A of size $(ST)^2$, where S stands for the area of each frame and T is the number of frames of our video. We fill this matrix using the coefficients of (12.20), therefore the main diagonal will contain coefficients -1 and the other values of the matrix will depend on \mathbf{v} . Matrix A is filled only where there are no occlusions, otherwise it is left to 0.

12.2 Applications

In this section we provide some results of the proposed method used on some image sequences from the Sintel database created by Butler et al. (2012) and from de Monkaa dataset designed by Mayer et al. (2016). We present two types of experiments:

- **Damaged objects recovering.** With the aim of recovering a moving object which is occluded in the video we consider an inpainting mask that covers part of an object. We apply our proposed method to fill-in the object of interest. These experiments also help us to evaluate how sensitive our method is to the given optical flow, to set the parameters, and also to compare with the 3D MBO suggested in Merriman et al. (1992) that evolves a surface by mean curvature motion. In Section 12.2.1 we present both, qualitative and quantitative results
- **Objects removal.** The goal is to remove an object from the input video which is occluding another one and apply the inpainting to complete the occluded object. Qualitative results are presented in Section 12.2.2.

Our model depends on two parameters: γ , the balance between the spatial and temporal derivatives, and τ , the diffusion time. We performed a thorough experimental analysis and finally observed that γ needs to be at least 1 in order to obtain good results. So we set it to $\gamma = 1.5$ for all the experiments. Regarding the diffusion time,

τ , it is well acknowledged from the threshold dynamics (Merriman et al. (1992); Barles and Georgelin (1995); Esedoglu et al. (2008)) methods that τ has to be big enough to allow the curve to evolve, but small enough so that the MBO scheme approximates motion by mean curvature as the solution of the Allen-Cahn equation. In all our experiments we set $\tau = 1$.

12.2.1 Damaged objects recovering

In this section we show the performance of our method on experiments where we know the ground-truth and we have a complete optical flow, computed using the NLTV-CSAD optical flow method. We also use these experiments to illustrate the behaviour of our video inpainting method with the proposed operator \mathcal{L} , the operator $\tilde{\mathcal{L}}$, and a comparison with the 3D MBO method. Also, in all these experiments we consider that the first and last frames are completely known, i.e., the object to be completed is fully visible in these two frames.

We show six experiments from the Sintel Dataset (from Fig. 12.3 to Fig. 12.8), where we have added a big mask, that is, where we do not know the value of the video. In order to provide quantitative results, we take advantage of the fact that we have the video ground-truth for all the experiments and we are able to compute the root mean square error, which is shown in Table 12.1. All the experiments are organized as follows: In the first row we present the original color frames from Sintel database (Butler et al. (2012)). On the second row we show data frames with the inpainting mask in gray and the object to be inpainted in white. The third row displays the occlusions from the same frames. Finally, in the last four rows we present output results: on the fourth row we show the ground-truth inpainted results. In the fifth row we present the results performed using the 2D+time MBO algorithm of Merriman et al. (1992). The last two rows show the performance of our operators, in the sixth row we present the inpainted results using the operator $\tilde{\mathcal{L}}$ and in the last row, the inpainted results using the proposed \mathcal{L} .

From all experiments we can see how crucial is to use the \mathcal{L} operator instead of $\tilde{\mathcal{L}}$ or the 3D-MBO scheme. For example, in Fig. 12.3 the hair of the girl is not completely recovered if we only consider the convective term, and in Fig. 12.6 the top of the finger is incomplete. The reason is that the pixels that need to be inpainted have an occluded trajectory – as it can be seen in the respective occlusion maps (third row) – and no temporal diffusion is applied on them. The spatial diffusion helps to complete these occluded areas. As it can be also observed, the results provided by the MBO method do not follow the trajectory of the moving objects but are completed according to the smoothness of the shapes. For example, in the last frame of Fig. 12.3 the face is not well recovered and in Fig. 12.6 the fingers are cut (completed by a plane in 2D+time), while in both cases our operator \mathcal{L} correctly completes the shapes.

For a quantitative evaluation, we present in Table 12.1 the root mean square error of three methods: our proposal (12.1) – (12.2), using $\tilde{\mathcal{L}}$ instead of \mathcal{L} , and MBO. Our operator \mathcal{L} always provides the smaller error.

	MBO	$\tilde{\mathcal{L}}$	\mathcal{L}
Alley 1	0.18	0.55	0.06
Ambush 4	0.46	0.54	0.26
Market 5	0.34	0.23	0.07
Shaman 3 (exp.1)	0.25	0.10	0.05
Shaman 3 (exp.2)	0.63	0.63	0.48
Temple 3	0.23	0.36	0.15

Table 12.1: Root mean square error of some sequences from Sintel Dataset (Butler et al. (2012)), obtained completing the moving shapes using either 3D-MBO scheme, $\tilde{\mathcal{L}}$ or \mathcal{L} operators.

12.2.2 Objects removal

As we are removing an object from the scene (with its own movement) we also need to interpolate the optical flow in the inpainting domain. For that, we use the method proposed in Palomares et al. (2014). The occlusion map is estimated from this completed flow using the criterion (12.6).

Before applying the proposed method we inpaint, independently, the first and last frame with a 2D binary inpainting method (in particular, the perceptual-based 2D-inpainting method described in Chapter 7). Then we apply the proposed video inpainting method that takes into account the estimated optical flow and occlusion map, and the completed shape is shown in the last row.

We show the performance of our algorithm in eight situations where we remove an object from the scene. We show three examples from the Sintel Dataset (from Fig. 12.9 to Fig. 12.11) and four more examples from the Monka Dataset (from Fig. 12.12 to Fig. 12.15). For each experiment we show in the first row the original image from the sequence; in the second row we show the object to be inpainted (in white) and the inpainting mask in gray); the third row is devoted to the optical flow, computed with the NLTV-CSAD method together with the mask inpainted. In the fourth row we show the occlusions and, finally, in the last row we show the inpainted result.

Let us notice how, in Fig. 12.10, the hole corresponding to the spear is correctly filled and the dragon claw is partially recovered in frames 5 and 6 although it was completely occluded in the corresponding frames of the input sequence.

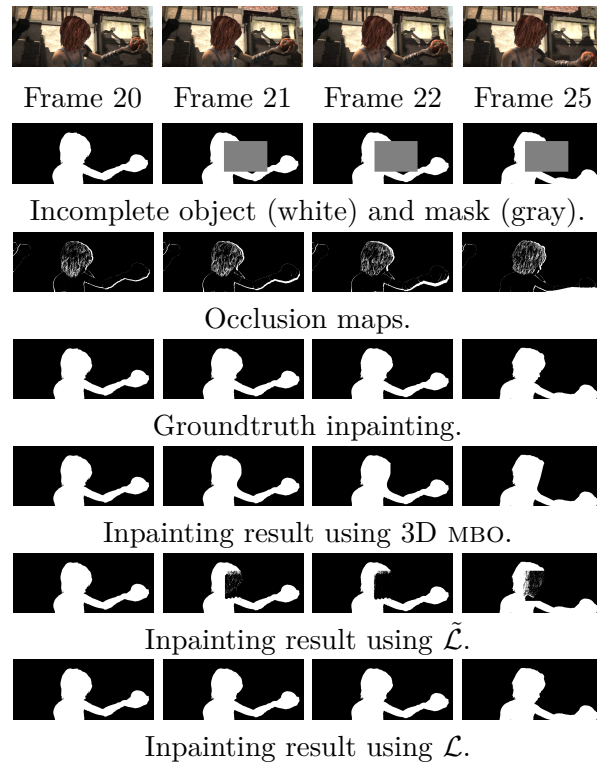


Figure 12.3: Damaged object recovering experiment. Inpainting results of some frames from *Alley 1* sequence of MPI Sintel. The sequence is formed by the frames 20 to 27. The inpainting mask is 6 frames long: from frame 21 to frame 27.

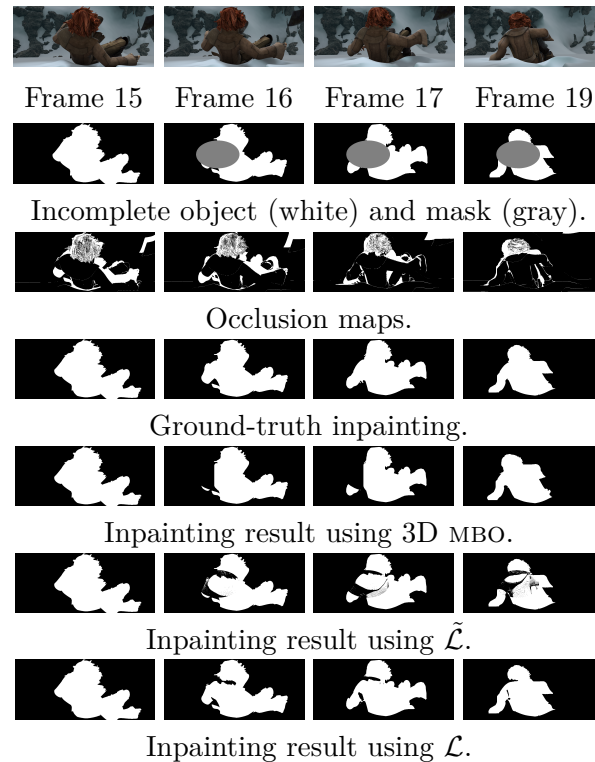


Figure 12.4: Damaged object recovering experiment. Inpainting results of some frames from *Ambush 4* sequence of MPI Sintel. The sequence is formed by the frames 15 to 22. The inpainting mask is 6 frames long: from frame 16 to frame 21.

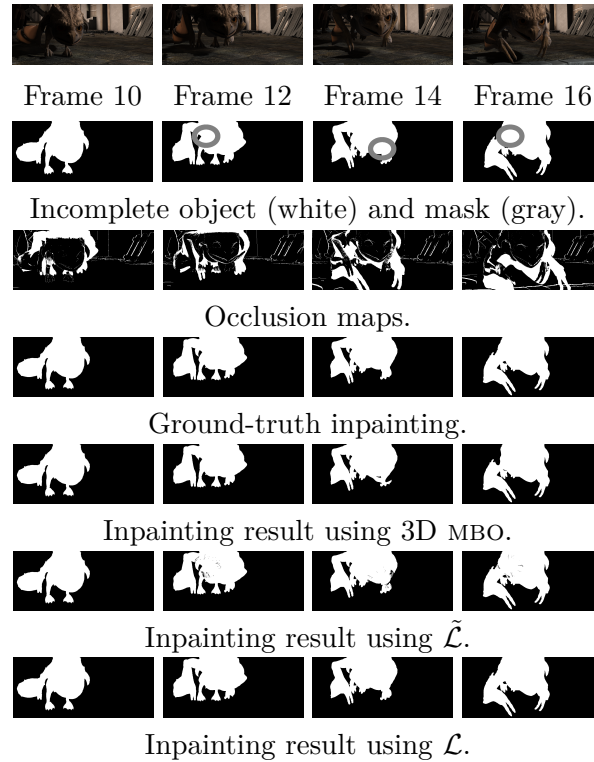


Figure 12.5: Damaged object recovering experiment. Inpainting results of some frames from *Market 5* of MPI Sintel. The sequence is formed by the frames 1 to 6. The inpainting mask is 4 frames long: from frame 2 to frame 5.

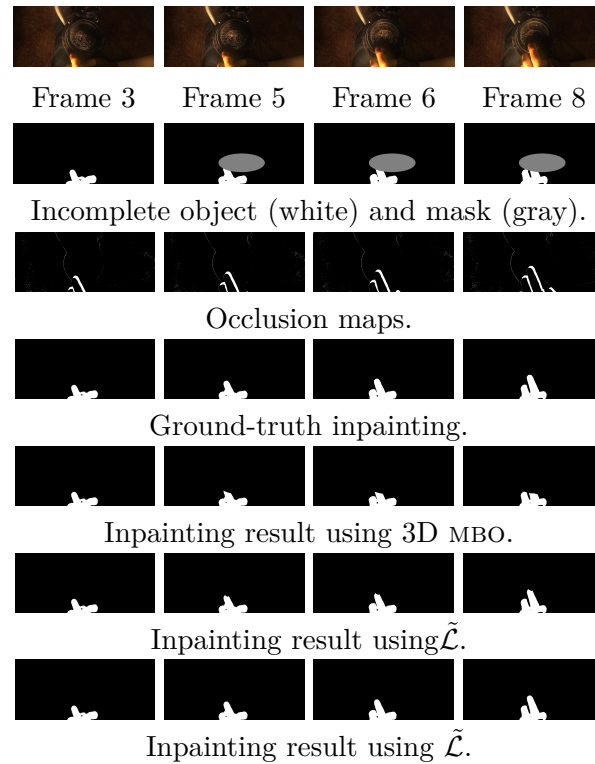


Figure 12.6: Damaged object recovering experiment. Inpainting results of some frames from *Shaman 3* sequence of MPI Sintel. The sequence is formed by the frames 3 to 9. The inpainting mask is 5 frames long: from frame 4 to frame 8.

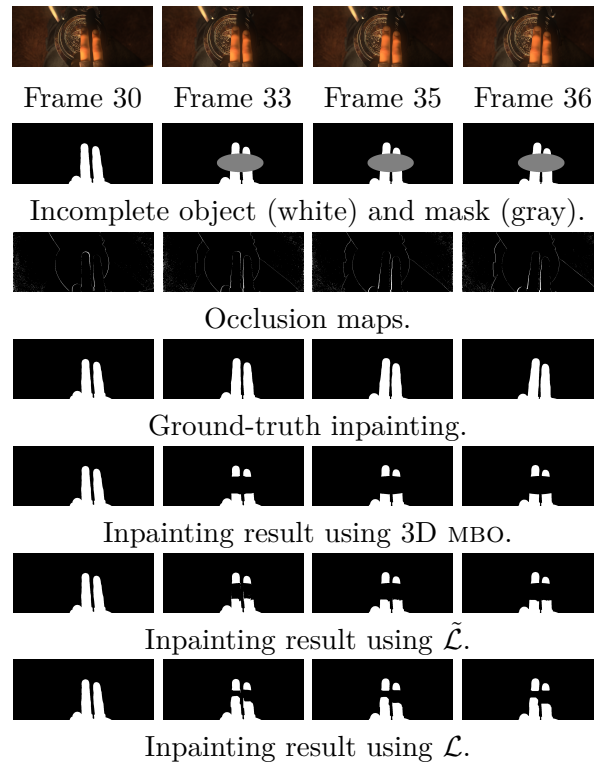


Figure 12.7: Damaged object recovering experiment. Inpainting results of some frames from *Shaman 3* of MPI Sintel. The sequence is formed by the frames 30 to 40. The inpainting mask is 9 frames long: from frame 31 to frame 39.

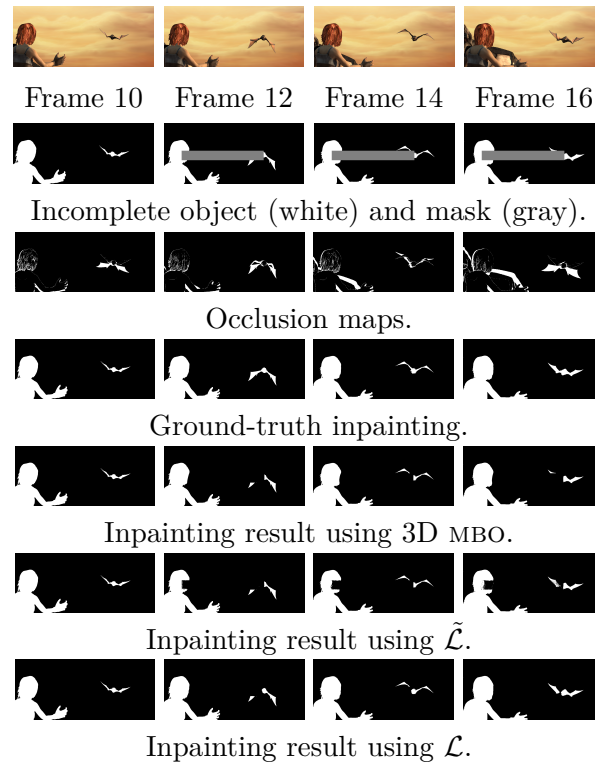


Figure 12.8: Damaged object recovering experiment. Inpainting results of some frames from *Temple 3* of MPI Sintel. The sequence is formed by the frames 10 to 17. The inpainting mask is 5 frames long: from frame 11 to frame 16.

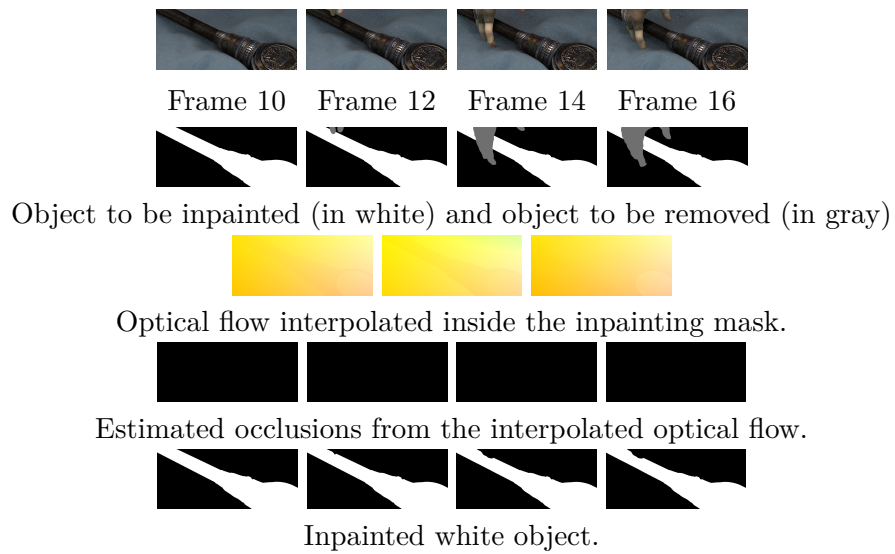


Figure 12.9: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 4 to 13 of the *Ambush 7* sequence of MPI Sintel. The inpainting mask is 9 frames long: from frame 2 to frame 10. The last frame is inpainted using the method proposed in Section 7.1.

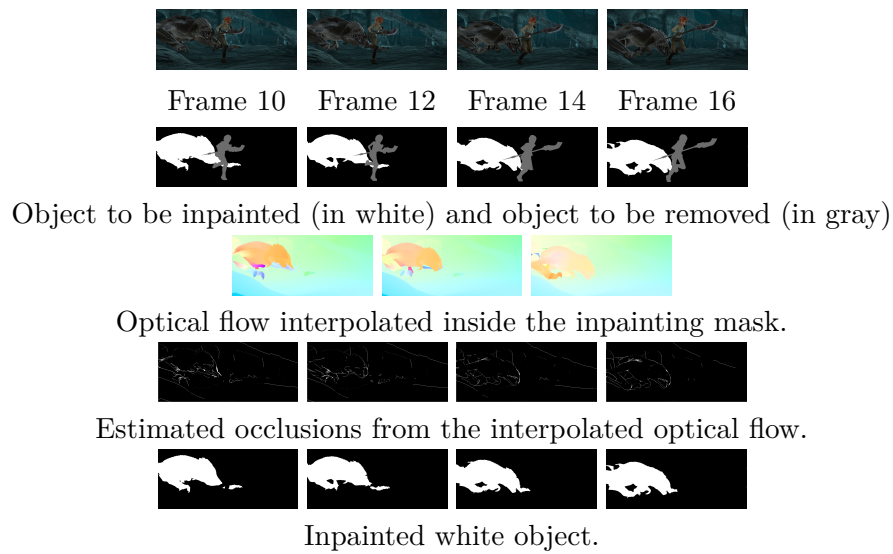


Figure 12.10: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 1 to 8 of the *Cave 2* sequence of MPI Sintel. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

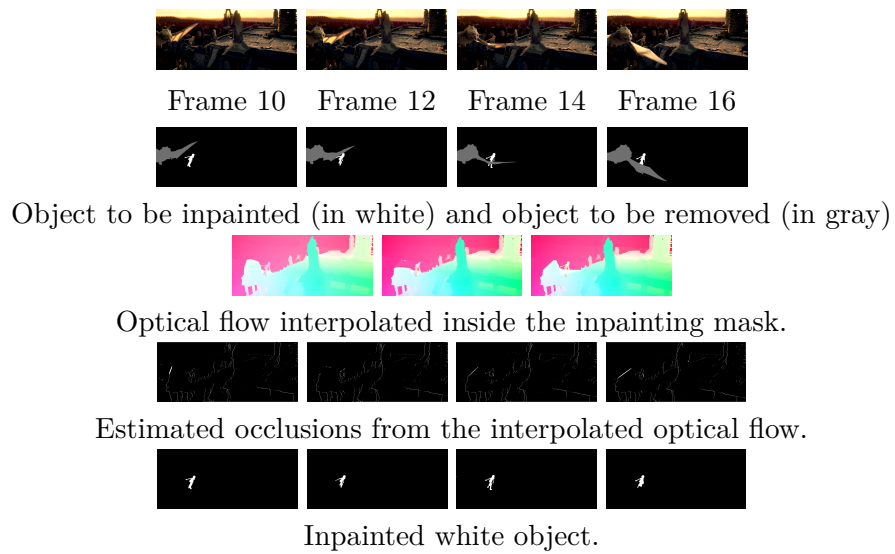


Figure 12.11: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 12 to 16 of the *Temple 2* sequence of MPI Sintel. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

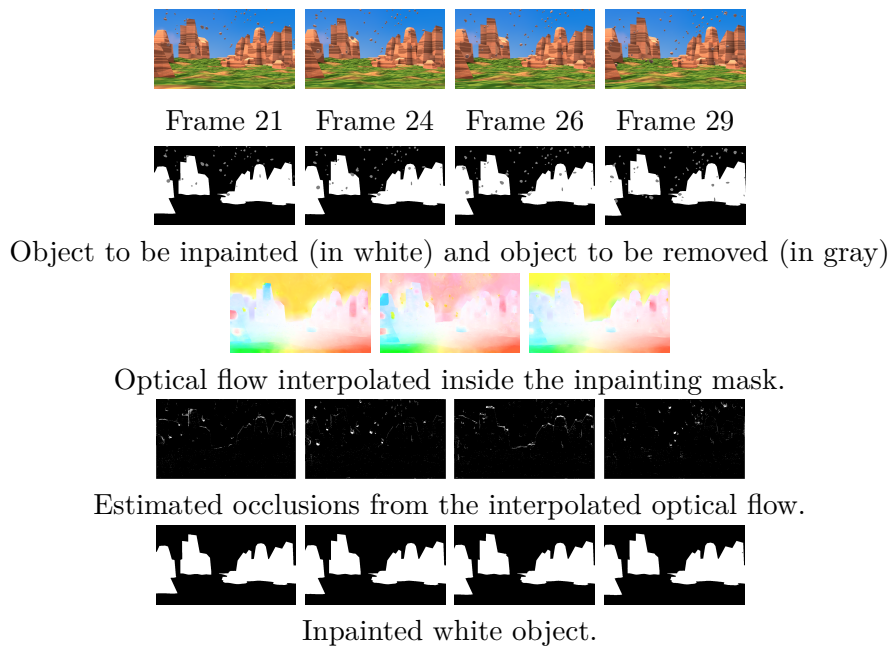


Figure 12.12: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 21 to 30 of the *A rain of Stones* sequence of Monkaa dataset. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

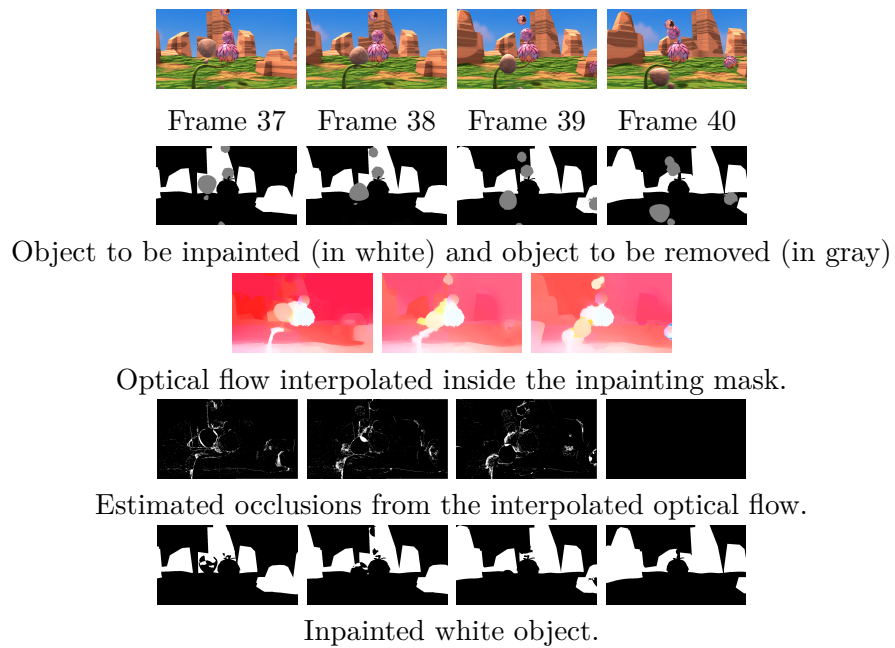


Figure 12.13: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 37 to 40 of the *Flower Storm* sequence of Monkaa dataset. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

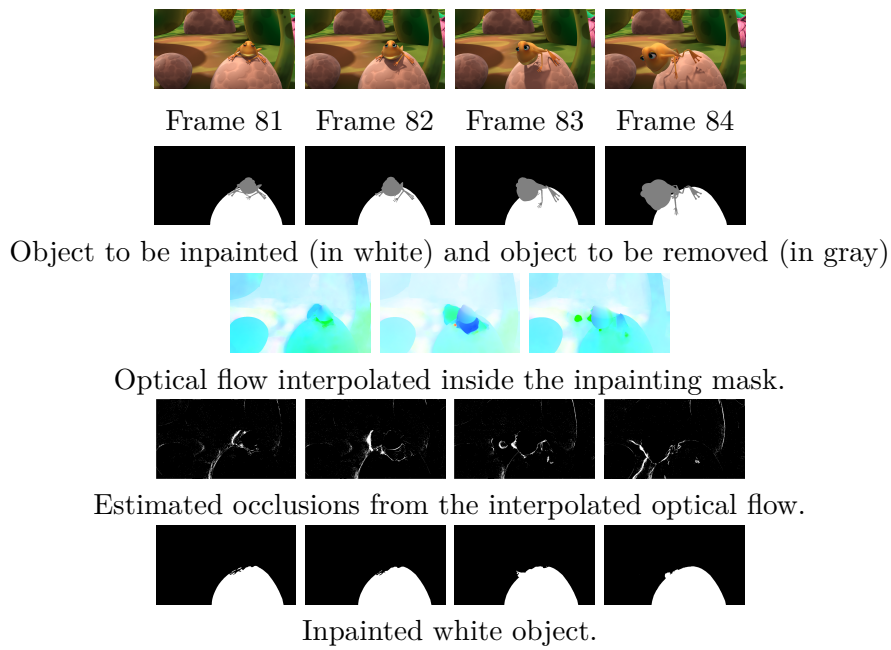


Figure 12.14: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 81 to 86 of the *Funny world* sequence of Monkaa dataset. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

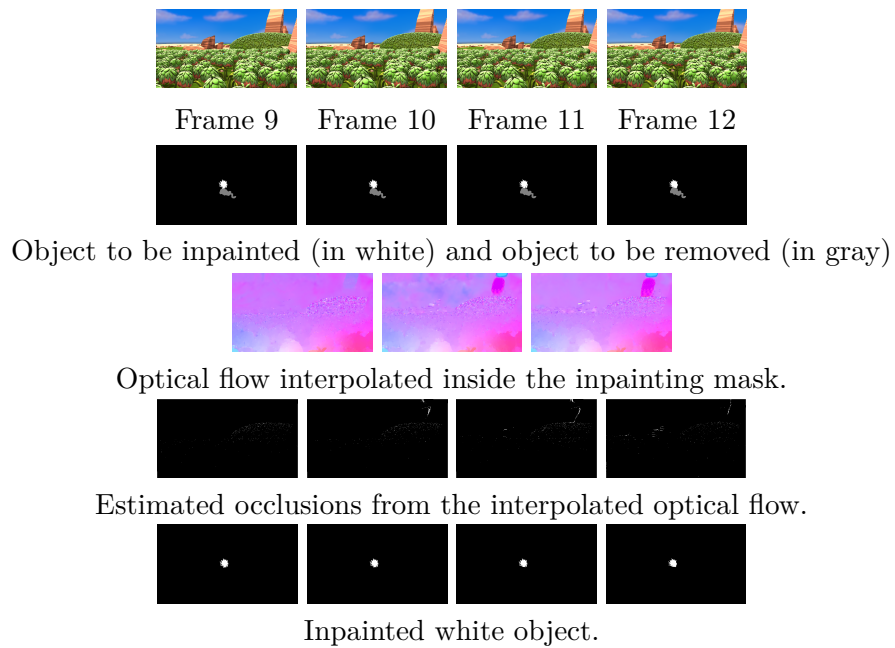


Figure 12.15: Application of the proposed inpainting method to the removal of an object in a video sequence. The sequence is formed by the frames from 8 to 13 of the *Top view* sequence of Monkaa dataset. The inpainting mask is present in all the frames. The first and last frames are inpainted using the method proposed in Section 7.1.

13 Optical Flow Inpainting

In this Chapter we propose to recover the missing optical flow of a given sequence, by solving the geodesic AMLE partial differential equation. An efficient numerical algorithm which uses the eikonal operators on finite graphs is also proposed. Finally, we present some results.

Given a video $u(\mathbf{x}, t)$ defined on $\Omega \times \{1, \dots, T\}$, where $\Omega \subset \mathbb{R}^2$ denotes the image frame domain and $\{1, \dots, T\}$ is the set of discrete times, let $\tilde{\mathbf{v}} = (\tilde{v}_1, \tilde{v}_2)$ be the optical flow of the video u representing the apparent motion between a pixel $\mathbf{x} \in \Omega \setminus \Omega_0(t)$ at time t and the corresponding at time $t + 1$.

We assume that, at time t , $\tilde{\mathbf{v}}(\mathbf{x}, t)$ is unknown in a region $\Omega_0(t) \subset \Omega$ whose boundary, denoted by $\partial\Omega_0$, consists of a finite union of smooth curves and possibly isolated points. In order to complete $\tilde{\mathbf{v}}$ in $\Omega_0(t)$ by an appropriate interpolation taking into account the objects in the video, we endow the whole domain Ω , at each time t , with a metric $g(t)$. Let $\mathcal{M}(t) = (\Omega, g(t))$ be the corresponding Riemannian manifold. We propose to complete $\tilde{\mathbf{v}}$ in $\Omega_0(t)$ with the motion field $\mathbf{v} = (v_1, v_2)$ such that v_1 and v_2 are solutions, respectively, of the geodesic Absolutely Minimizing Lipschitz Extension (AMLE), given by the PDE

$$\begin{aligned} \Delta_{\infty, g} v_i &= 0 \quad \text{in } \Omega_0(t) \\ \text{s.t. } v_i|_{\partial\Omega_0(t)} &= \tilde{v}_i, \end{aligned} \tag{13.1}$$

for $i = 1, 2$, respectively. We also use Neumann boundary conditions on $\partial\Omega$. Here we have denoted

$$\Delta_{\infty, g} v_i := D_{\mathcal{M}}^2 v_i \left(\frac{\nabla_{\mathcal{M}} v_i}{|\nabla_{\mathcal{M}} v_i|}, \frac{\nabla_{\mathcal{M}} v_i}{|\nabla_{\mathcal{M}} v_i|} \right), \quad (13.2)$$

where $\nabla_{\mathcal{M}} v_i$ and $D_{\mathcal{M}}^2 v_i$ denote, respectively, the gradient and the Hessian on the manifold. To simplify, we have omitted the dependence on t of g and \mathcal{M} ; we will also drop the subindex i in what follows. When g is the Euclidean metric, the operator in (13.2) is known as the infinity Laplacian. In our proposal, we define the metric g in terms of the local geometry and texture content of the frame $u(\mathbf{x}, t)$ corresponding to $\mathbf{v}(\mathbf{x}, t)$. For instance, g can be given by affine covariant structure tensors defined in Fedorov et al. (2015) and detailed in Chapter 2 or taking into account spatial distances and photometric similarities as detailed in the following section, where we consider the interpolation problem (13.1) on finite graphs.

13.1 The geodesic AMLE on a finite graph

We solve the AMLE equation on the manifold with a numerical algorithm that is based on the eikonal operators for nonlinear elliptic PDEs on a finite graph, which was proposed by Oberman (2005) and Manfredi et al. (2015). In particular, we consider the discrete grid of Ω as the nodes of a finite weighted graph G . Given a point \mathbf{x} on the grid, let $\mathcal{N}(\mathbf{x})$ be a neighborhood of \mathbf{x} . Following Manfredi et al. (2015), we consider the positive and negative eikonal operators given, respectively, by

$$\|\nabla v(\mathbf{x})\|_{\mathcal{M}}^+ = \max_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} \frac{v(\mathbf{y}) - v(\mathbf{x})}{d^g(\mathbf{x}, \mathbf{y})}, \quad (13.3)$$

$$\|\nabla v(\mathbf{x})\|_{\mathcal{M}}^- = \min_{\mathbf{z} \in \mathcal{N}(\mathbf{x})} \frac{v(\mathbf{z}) - v(\mathbf{x})}{d^g(\mathbf{x}, \mathbf{z})}. \quad (13.4)$$

where d^g is a geodesic distance between the points computed using the metric g . Then, the discrete infinity Laplacian corresponds to

$$\Delta_{\infty,g}v(\mathbf{x}) = \frac{\|\nabla v(\mathbf{x})\|_{\mathcal{M}}^+ + \|\nabla v(\mathbf{x})\|_{\mathcal{M}}^-}{2}. \quad (13.5)$$

We solve (13.1) with the following iterative discrete scheme

$$v^{k+1}(\mathbf{x}) = \frac{d^g(\mathbf{x}, \mathbf{z})v^k(\mathbf{y}) + d^g(\mathbf{x}, \mathbf{y})v^k(\mathbf{z})}{d^g(\mathbf{x}, \mathbf{z}) + d^g(\mathbf{x}, \mathbf{y})} \quad (13.6)$$

where \mathbf{y} and \mathbf{z} are the pixels providing the maximum and minimum in (13.3) and (13.4), respectively. This scheme is applied only for $\mathbf{x} \in \Omega_0$, initializing $u^0(\mathbf{x}) = 0$ in that case and keeping the values of $v_1(\mathbf{x})$ and $v_2(\mathbf{x})$ on the known region $\Omega \setminus \Omega_0$ for all k .

If \mathbf{x} and \mathbf{y} are neighbouring pixels, we define their distance by one of the following simple possibilities

$$d_1(\mathbf{x}, \mathbf{y}) = \sqrt{(1-\lambda)\|u(\mathbf{x}, t) - u(\mathbf{y}, t)\|^2 + \lambda\|\mathbf{x} - \mathbf{y}\|^2} \quad (13.7)$$

$$d_2(\mathbf{x}, \mathbf{y}) = (1-\lambda)\|u(\mathbf{x}, t) - u(\mathbf{y}, t)\| + \lambda\|\mathbf{x} - \mathbf{y}\| \quad (13.8)$$

where $\lambda \in [0, 1]$. We also include

$$d_3(\mathbf{x}, \mathbf{y}) = (1-\lambda)\|u(\mathbf{x}, t) - u(\mathbf{y}, t)\|^2 + \lambda\|\mathbf{x} - \mathbf{y}\|^2 \quad (13.9)$$

which is a semimetric, i.e., it does not satisfy the triangle inequality. We will use $d(\mathbf{x}, \mathbf{y})$ to refer to anyone of them. Given a path $\gamma = \{\gamma(i)\}_{i=0}^m$ joining two points, $\mathbf{x} = \gamma(0)$ and $\mathbf{y} = \gamma(m)$, its length is defined as usual by $L^g(\gamma) = \sum_{i=0}^{m-1} d(\gamma(i), \gamma(i+1))$. Given any two points \mathbf{x} and \mathbf{y} on the grid, then the geodesic distance $d^g(\mathbf{x}, \mathbf{y})$ is

$$d^g(\mathbf{x}, \mathbf{y}) = \inf\{L^g(\gamma) : \gamma \text{ is a curve joining } \mathbf{x} \text{ and } \mathbf{y}\}.$$

This distance can be computed using Dijkstra's algorithm. In practice, for any pair of points \mathbf{x}, \mathbf{y} we approximate $d^g(\mathbf{x}, \mathbf{y})$ by $d_i(\mathbf{x}, \mathbf{y})$, where $d_i(\mathbf{x}, \mathbf{y})$, for $i = 1, 2, 3$, is defined by (13.7)–(13.9). Let us notice again that more sophisticated metrics are possible.

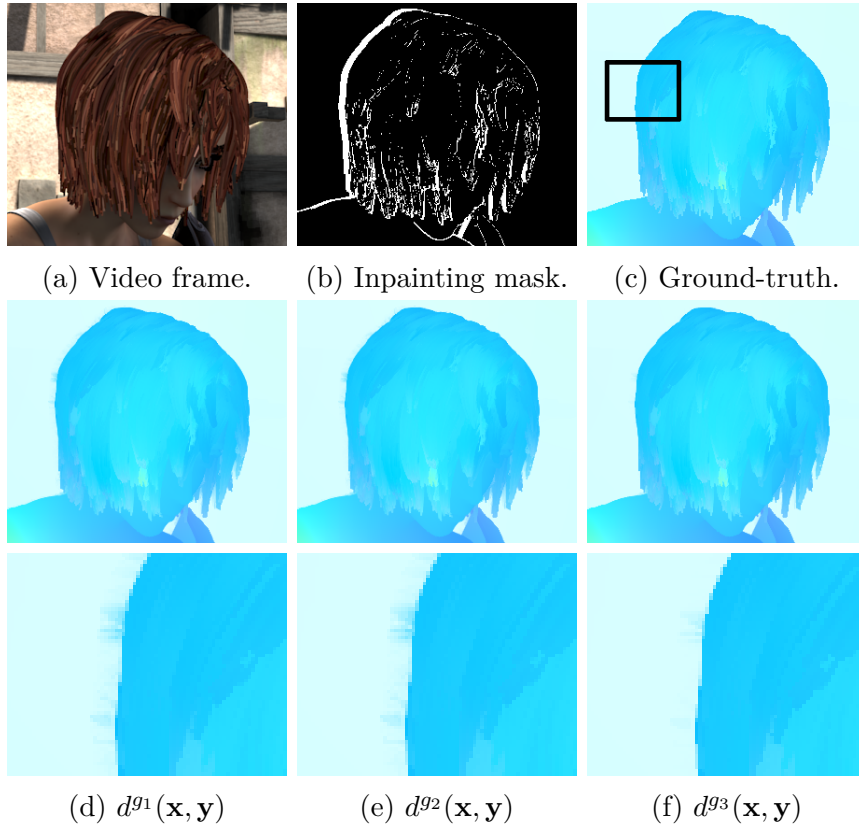


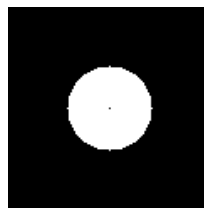
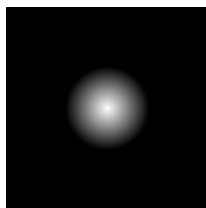
Figure 13.1: Comparison of the three proposed possibilities for the metric illustrated in an experiment of completion of the optical flow in the occlusion areas (white regions in image b). In the last row we present a zoom of the result obtained with each distance.

To experimentally analyze the behavior of our interpolation method depending on the considered metric g we show in Fig. 13.1 some results using d^{g^1} , d^{g^2} and d^{g^3} . We choose to work with $d_3(\mathbf{x}, \mathbf{y})$ given by (13.9), which produces slightly better results. Indeed, when part of an edge is subjective or weak (i.e., the two regions separated by the edge are similar) the proposed metrics do not penalize completely

the propagation of the optical flow from one region to another; (13.9) does a better job discriminating these cases. Fig. 13.1 shows an example, notice how the small leak in the left middle part of the head is reduced in Fig. 13.1f.

13.1.1 Numerical Multiscale Approach

The scheme is embedded in a multiscale approach: the input optical flow and corresponding video frame are downsampled to a set of scales and the solution is computed at each one using (13.6). The inputs are downsampled by a factor of two using 2×2 block averages; bilinear interpolation is used for upsampling. At the coarsest level, the unknowns are initialized to zero; the other scales are initialized by upsampling the solution of the previous scale. The multiscale pyramid provides just an initialization closer to the solution, leading to a faster convergence. In Fig. 13.3 we show this behaviour for a simple situation: we want to interpolate an image of a cone and the cone itself is used to compute the metric. In Figure 13.2 we show the expected result in Fig. 13.2a and the inpainting mask in Fig. 13.2b. As it can be seen in Fig. 13.2b the unknown region is the white disc, which is the support of the unknown cone, except for the single central point (the top of the cone). In each graphic of Fig. 13.3 we present the evolution of the interpolation error for 10^5 iterations. At each iteration,



(a) Ground-truth cone image. (b) Inpainting mask in white.
Used to compute the metrics.

Figure 13.2: Inpainting mask for the cone experiment.

the error is computed as the difference between the current solution and the expected one (that is the cone). We compare three different initializations: zeros inside the mask (first column), starting with the result of the previous scale of the multiscale approach (second column) and the ground-truth (third column). For each row we show the evolution for each considered distance. These graphics allow us to observe that the two first rows converge fast to the cone, but for the last row we never get an error smaller than 0.11. This is because the cone is not the solution of the PDE with that metric. We can also observe as the multi-scale approach gives an initialization closer to the expected solution.

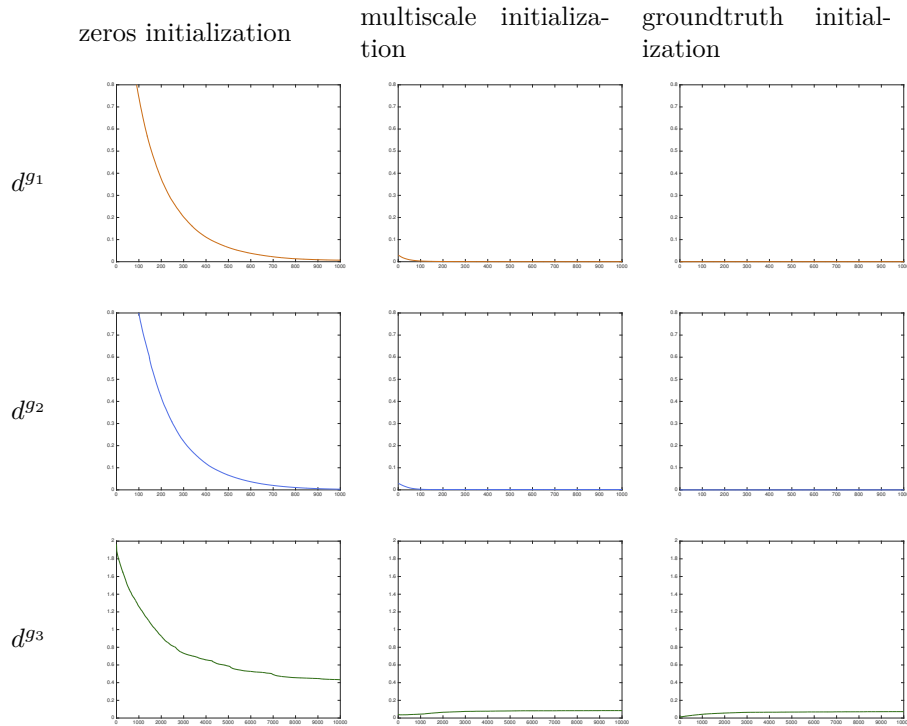


Figure 13.3: Error evolution using different initializations and metrics.

13.1.2 Neighbourhood

As Oberman (2005) and Manfredi et al. (2015) mention, the solution of the numerical scheme (13.6) converges to the solution of (13.1) when the local spatial resolution dx and the local directional resolution $d\theta$ tend to 0. We have evaluated three discrete neighbourhoods, $\mathcal{N}_i(\mathbf{x})$, $i = \{1, 2, 3\}$, to approximate the behaviour of the continuous scheme. These neighbourhoods are constructed by considering the discrete square of semiside dx centered at \mathbf{x} , which is provided by the different directional resolutions $d\theta$. The first neighbourhood approximation $\mathcal{N}_1(\mathbf{x})$, which was already proposed by Oberman (2005), consists on reducing the directional resolution $d\theta$ by increasing dx , that is, we keep only the points of the boundary of the square of semiside dx . In Fig. 13.4 we show, for different values of $d\theta$ and dx , the pixels taken into account when we use this neighbourhood.

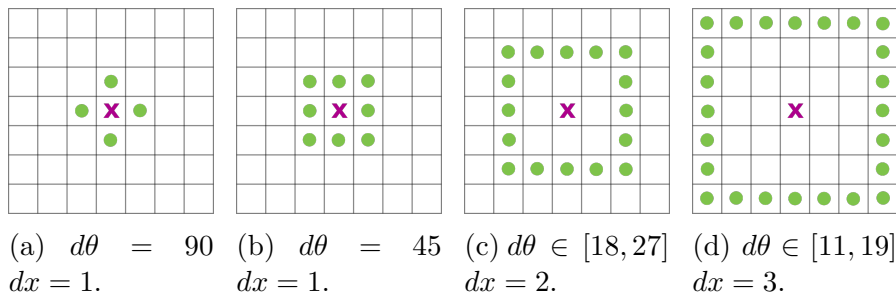


Figure 13.4: $\mathcal{N}_1(\mathbf{x})$ with different values of $d\theta$ and considering the boundary of the squares of the specified dx semiside.

As it can be observed in Fig. 13.4, using $\mathcal{N}_1(\mathbf{x})$ with a small $d\theta$ implies going far from the central pixel \mathbf{x} and, as consequence, the approximation of the derivatives involved in (13.1) is less accurate. That is, with $\mathcal{N}_1(\mathbf{x})$ we can have either a small $d\theta$ or dx , but not both. For this reason we propose to improve it by considering also all the points in between, that is, instead of only considering the boundary of the square of semiside dx , we also take into account all the points in between. By doing so, we are able to reduce dx and considering

pixels closer to the central one and, in some situations, we are also able to improve the angle resolution. In Fig. 13.5 we present some neighbourhoods obtained using $\mathcal{N}_2(\mathbf{x})$.

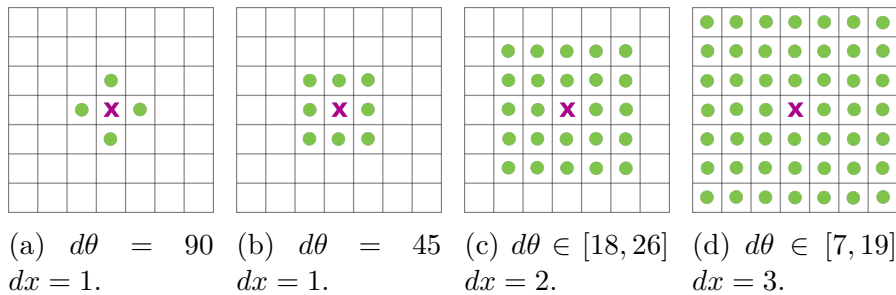


Figure 13.5: $\mathcal{N}_2(\mathbf{x})$ with different values of $d\theta$ and considering all the points of the squares of the specified dx semiside.

Despite being better in resolution terms, the second type of neighbourhood uses some redundant points, that is, for each direction we are considering all the points up to dx . For this reason we consider a third type of neighbourhood, $\mathcal{N}_3(\mathbf{x})$, which considers for each direction $n \cdot d\theta$, $n \in \mathbb{N}$, only the pixel that minimizes the distance to the central pixel. With this last kind of neighbourhood we achieve the minimum values for both, dx and $d\theta$, and also reduce the number of computations. Figure 13.6 presents some neighbourhoods of $\mathcal{N}_3(\mathbf{x})$.

In Figure 13.7 we present an optical flow inpainting result obtained with each neighbourhood and using $dx = 2$ (Figures d, e, f). For all the experiments we have fixed $\lambda = 10^{-3}$, we have used d^{g_3} and $dx = 2$. We can observe as the result obtained with the neighbourhood \mathcal{N}_1 is much worse than the other two because it uses pixels that are far from the pixel to be interpolated. Results produced with \mathcal{N}_2 and \mathcal{N}_3 barely differ.

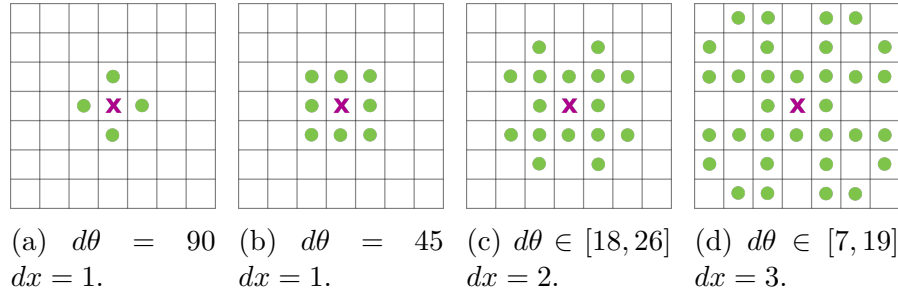


Figure 13.6: $\mathcal{N}_3(\mathbf{x})$ with different values of $d\theta$ and the pixels that minimize its distance.

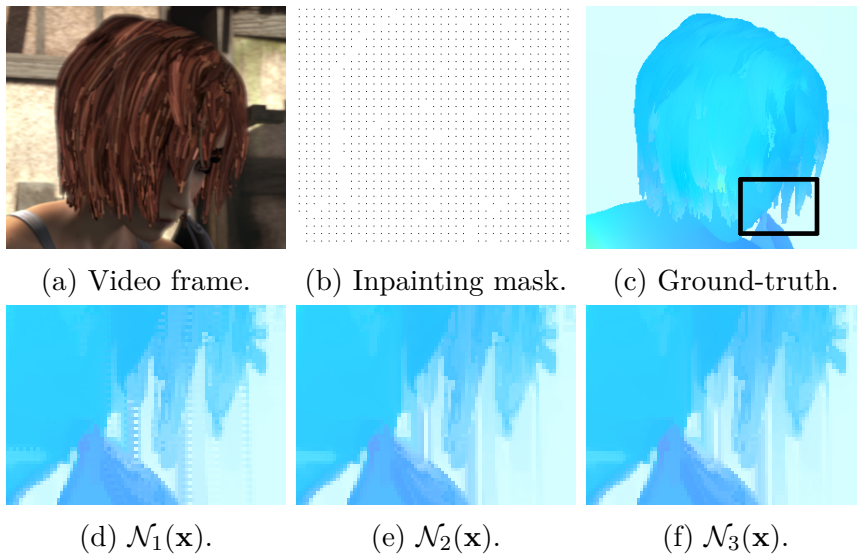


Figure 13.7: Comparison of the results obtained with each neighbourhood: $\mathcal{N}_1(\mathbf{x})$, $\mathcal{N}_2(\mathbf{x})$, $\mathcal{N}_3(\mathbf{x})$.

13.1.3 Influence of the Image to Compute the Metric

In this section we show the influence of the image used to compute the metric in our method. We will denote it by *guide*. To this goal

we present an inpainting result (Fig. 13.8) and a densification result (Fig. 13.9). Both of them are performed using four different guides: a shape guide, where the main object is in white and all the background in black (Figs. 13.8c and 13.9c); a cartoon guide, which has no texture (Figs. 13.8d and 13.9d); a textured guide (Figs. 13.8e and 13.9e); and a realistic guide, this last one may contain texture, blur and shadows among others (Figs. 13.8f and 13.9f).

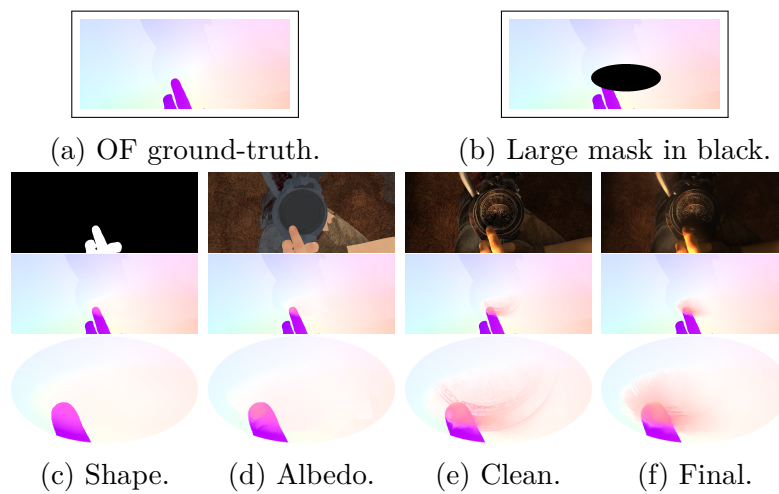


Figure 13.8: Inpainting results with different guides with a zoom on the region of interest (Shaman 3).

We can observe as the results obtained from the shape (Figs. 13.8c and 13.9c) and cartoon (Figs. 13.8d and 13.9d) guides, thanks to the sharper object edges, better preserve the motion boundaries than the other two results, i. e. the optical flow is not diffused outside the object. When we introduce texture we can observe as the small textures of the objects affect the color difference and, therefore, produce larger distances. For instance, we can observe in Fig. 13.8e as the texture from the guide can also be observed in the recovered optical flow, producing undesirable discontinuity in the optical flow. In contrast with the previous behaviour, it may happen that the colors among objects

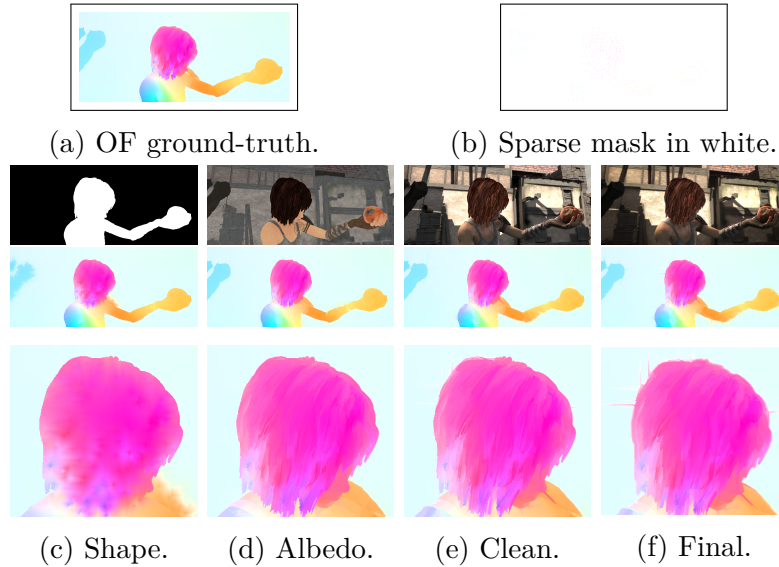


Figure 13.9: Densification results with different guides with a zoom on the region of interest (Alley 1).

are very close, which produces small distances in the geodesic and the optical flow starts leaking outside the object. This behaviour is illustrated in the example of Fig. 13.9 where the color of the building is very close to the color of the hair.

13.2 Applications

In this section we present optical flow inpainting results on the sequences from Sintel database (Butler et al. (2012)), Middlebury (Baker et al. (2011)) and Kitty (Menze and Geiger (2015)) datasets. All the experiments are performed with $\lambda = 10^{-3}$, \mathcal{N}_3 and d^{g3} . We show results for two different applications:

- **Region inpainting.** It is applied to complete the optical flow in occlusion areas and in large holes where different types of

motion have to be recovered. These applications are performed on frames from the Sintel Dataset.

- **Densification.** We consider three situations: densification of groundtruth (gt) flow values at random positions (1%, 5% and 30% of values kept), densification of optical flow values given by the Deepmatching algorithm (DM) proposed by Revaud et al. (2016)), and densification of ground-truth values at the sparse positions given by Deepmatching.

We propose to compare our results to those of EpicFlow proposed by Revaud et al. (2015), which considers an interpolation step. In fact, they proposed to estimate the optical flow in two steps:

- 1) Edge-preserving interpolation of a sparse optical flow generated with the Deepmatching algorithm (Revaud et al. (2016)).
- 2) Variational energy minimization using as initialization the dense flow achieved in the first step.

As we are not estimating the optical flow given two consecutive frames, we compare our results to those in their first step, since both of them are completions of an initial flow that uses only the current frame. In each case, we compute the End-Point-Error (EPE) excluding the data points, so as not to penalize EpicFlow results which changes these values. End-Point-Error is defined as the L_2 -norm of the difference between the estimated motion vector and the ground-truth one. Average End-Point-Error is the average of all the end-point errors in an image and is the standard error measure used in the optical flow benchmarks (usually denoted simply as EPE). We present the EPE values in Table 13.1. The comparison is done over all the training Sintel dataset (Butler et al. (2012)) and on the optical flow corresponding to frames 10 and 11 from the Middlebury dataset, where the ground-truth is available (Baker et al. (2011)).

We can observe that our method achieves lower error for all cases except for the densification of Deepmatching optical flow (sparse DM).

	SINTEL		MIDDLEBURY	
	Ours - EPE	EF - EPE	Ours - EPE	EF - EPE
sparse 1%	0.7061	1.8532	0.1979	0.3105
sparse 5%	0.4340	1.4199	0.1053	0.2426
sparse 30%	0.2241	1.1212	0.0567	0.1801
sparse DM	4.4404	4.1507	0.9216	0.8112
sparse DM (gt)	2.1360	3.5411	0.2049	0.2789
occlusions	5.4198	6.8797	–	–
hole	1.7208	1.9587	–	–

Table 13.1: Comparison of the EPE for EpicFlow and our method in different situations.

This is probably due to the fact that the data provided by the DM are quantized values that contain errors and outliers, which the AMLE propagates along all the pixels of the inpainting hole. By contrast, EpicFlow modifies the provided values which allows to compensate for the quantization and errors, while our method sticks to the given values propagating the error to the recovered optical flow. Depending on the application, one or the other behaviour would be preferred. We include, in Fig. 13.10, an example of completion in large regions without data as in Kitti dataset of Menze and Geiger (2015) and, in Fig. 13.11 we show an example of inpainting of large holes.

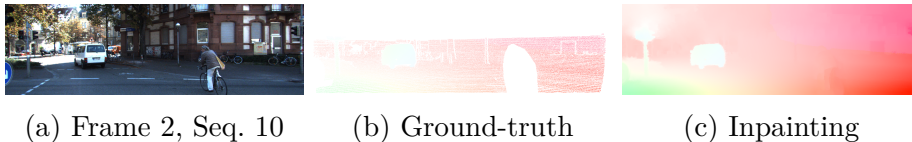


Figure 13.10: A result on Kitti dataset proposed by Menze and Geiger (2015) that contains large holes.

Finally, in Fig. 13.12 we present two different experiments and illustrate the behaviour of different interpolation methods: Total Variation (TV) regularization proposed by Rudin et al. (1992), the

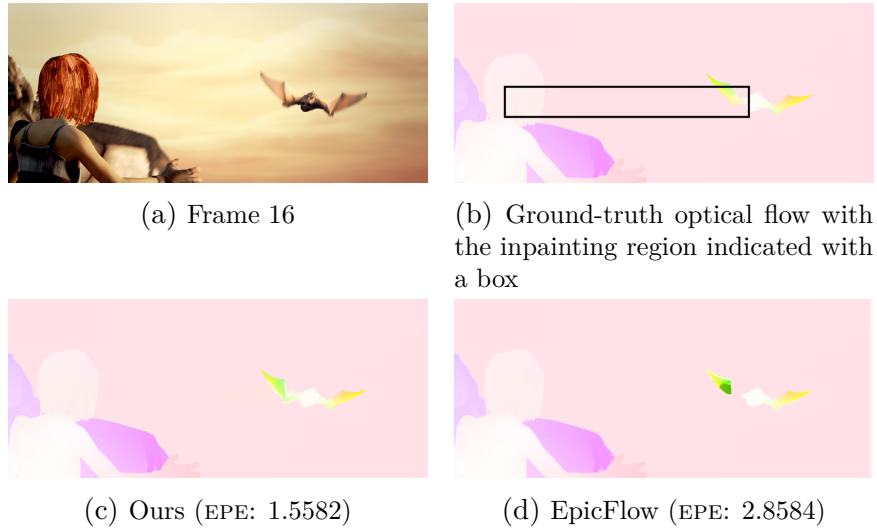


Figure 13.11: Some optical flow inpainting results for a frame of the *Temple 3* sequence of MPI-Sintel benchmark.

rotation invariant regularization defined in Palomares et al. (2014) (Rot-Inv), the interpolation step of EpicFlow (Revaud et al. (2015)), and the proposed method. Both type of experiments are done with the frame 10 of RubberWhale from the Middlebury dataset (Figure 13.12(a)), whose optical flow ground-truth is available (13.12(b)). The first experiment is motion completion in different large holes (shown in white in image 13.12(c)) and the corresponding results for the four evaluated methods are in the 2nd row. The second experiment is flow densification from a sparse set of matches (pixels where the motion is not known are shown in white in 13.12(d)) and the results can be seen in the 3rd row. One can observe that when no guide is used in the interpolation process (TV and Rotation-invariant methods) the discontinuities of the optical flow are not well recovered, since they are not aligned with the objects boundaries. This can be seen for instance, in the semi-circles in the second row of Fig. 13.12 where these methods fail, while the interpolation step of

EpicFlow and our interpolation give satisfactory results. Our interpolation method can not correctly inpaint the optical flow of rotating objects, as the wheel in the bottom-left. In this region the information provided by the guide-based metric is not useful (roughly homogeneous area). In such cases, our method reduces to a local isotropic average which can only solve translations. EpicFlow’s interpolation adjusts instead a local affine transformation, allowing to correctly recover rotations. Although, in general, one can observe that visual results of our interpolation method are better (see Fig. 13.11d).

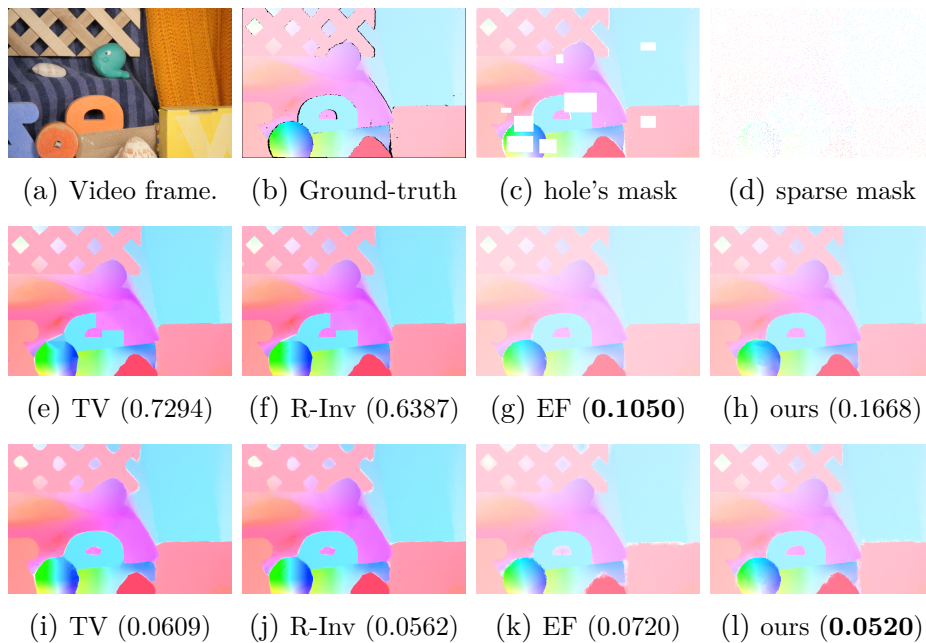


Figure 13.12: Comparison of different motion completion algorithms in two different cases: inpainting of large holes (holes in white in image (c) and results shown in the 2nd row) and flow densification from a sparse set of matches (shown in image (d) and results in 3rd row).

14

Conclusions and Future Work

In this part we have analyzed two important ingredients of a model for video inpainting: the optical flow and the shape inpainting.

We have proposed a variational method for binary video inpainting. In order to minimize it we follow a threshold dynamics strategy where the dynamic shape analysis imposes spatial and temporal smoothness along the visible trajectory of the objects by incorporating the convective derivative in a differential operator based on a generalized 3D gradient. Our proposal allows to keep track of the motion occlusions among the moving binary objects. We present some experimental results containing complex object motion and occlusions. As future work and follow-up of our shape inpainting method, we will use it to guide a texture video inpainting model. For instance, in the context of patch-based models, we can use it to reduce the search domain of similar textures.

The optical flow inpainting completion is based on the Absolutely Minimizing Lipschitz Extension equation (or the infinity Laplace equation) on the 2D Riemannian manifold given by the frame domain endowed with an appropriate metric, defined by the image frame, which acts as a guide for the resulting anisotropic diffusion. The proposed method has been analyzed in three different types of experiments: interpolation of sparse matches (i.e. optical flow densification), and motion completion both in occlusion areas and in large holes. The experimental results show how it outperforms the

EpicFlow interpolation in practically all the situations. We have proposed three different simple metrics and as part of the future work we plan to study more sophisticated ones in order to further improve the results.

As future work we plan to study a variational model for the joint estimation of the shape and optical flow completion built on the two proposed methods.

Bibliography

- S. M. Allen and J. W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, 1979.
- A. Almansa, F. Cao, Y. Gousseau, and B. Rougé. Interpolation of digital elevation models using AMLE and related methods. *IEEE Transactions on Geoscience and Remote Sensing*, 40(2).
- S. Alpert, M. Galun, R. Basri, and A. Brandt. Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- P. Arias. *Variational Methods for exemplar-based image inpainting and gradient-domain video editing*. PhD. Thesis, 2013.
- P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A Variational Framework for Exemplar-Based Image Inpainting. *International Journal of Computer Vision*, 93(3):319–347, 2011.
- G. Aronsson. Extension of functions satisfying Lipschitz conditions. *Arkiv för Matematik*, 6(6):551–561, 1967.
- G. Aronsson. On the partial differential equation $u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy} = 0$. *Arkiv för Matematik*, 7, 1968.

- G. Aronsson, M. Crandall, and P. Juutinen. A tour of the theory of absolutely minimizing functions. *Bulletin of the American mathematical society*, 41(4):439–505, 2004.
- J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4):678–689, 1990.
- J. F. Aujol, S. Ladjal, and S. Masnou. Exemplar-Based Inpainting from a Variational Point of View. *SIAM Journal on Mathematical Analysis*, 42(3):1246–1285, 2010.
- S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera. Filling-In by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.
- C. Ballester, F. Calderero, V. Caselles, and G. Facciolo. Multiscale analysis of similarities between images on riemannian manifolds. *Multiscale Modeling & Simulation*, 12(2):616–649, 2014.
- G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.
- C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patch-match: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.
- M. Barni. A fast algorithm for 1-norm vector median filtering. *IEEE Transactions on Image Processing*, 6(10):1452–1455, 1997.
- M. Barni, V. Cappellini, and A. Mecocci. A fast l1-metric vector median filter. In *14^o Colloque sur le Traitement du Signal et des Images, FRA*, 1993.

- G. Bellettini, G. Dal Maso, and M. Paolini. Semicontinuity and relaxation properties of a curvature depending functional in 2d. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, 20(2):247–297, 1993.
- B. Berkels, C. Kondermann, C. Garbe, and M. Rumpf. Reconstructing optical flow fields by motion inpainting. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 388–400. Springer, 2009.
- J. Bernoulli. Quadratura curvae, e cujus evolutione describitur inflexae laminae curvatura. *Die Werke von Jakob Bernoulli*, pages 223–227, 1692.
- A. L. Bertozzi and A. Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.
- A. L. Bertozzi, S. Esedoglu, and A. Gillette. Inpainting of binary images using the Cahn-Hilliard equation. *IEEE Transactions on Image Processing*, 16(1):285–291, 2007.
- J. C. Bezdek, L. Hall, and L. Clarke. Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4):1033–1048, 1993.
- P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. Cohen, B. Curless, and S. B. Kang. Using photographs to enhance videos of a static scene. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 327–338. Eurographics Association, 2007.
- P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics*, 29(2):1–14, 2010.

- A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65. IEEE, 2005.
- A. Bugeau, P. G. I. Piracés, O. D’Hondt, A. Hervieu, N. Papadakis, and V. Caselles. Coherent background video inpainting through Kalman smoothing along trajectories. In *VMV 2010-15th International Workshop on Vision, Modeling, and Visualization Workshop*, pages 123–130, 2010.
- J. Burge, C. C. Fowlkes, and M. S. Banks. Natural-scene statistics predict how the figure–ground cue of convexity affects human depth perception. *Journal of Neuroscience*, 30(21):7269–7280, 2010.
- D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.
- L. Calatroni, Y. van Gennip, C.-B. Schönlieb, H. M. Rowland, and A. Flenner. Graph clustering, variational image segmentation methods and hough transform scale detection for object measurement in images. *Journal of Mathematical Imaging and Vision*, 57(2):269–291, 2017.
- F. Cao, Y. Gousseau, S. Masnou, and P. Pérez. Geometrically Guided Exemplar-Based Inpainting. *SIAM Journal on Imaging Sciences*, 4(4):1143–1179, 2011.
- S. Carrigan, E. Palmer, and P. Kellman. Differentiating Local and Global Processes in Amodal Completion Through Dot Localization. *Journal of Vision*, 15(12):1123–1123, 2015.
- V. Caselles, J. M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 7(3):376–386, 1998.

- V. Caselles, L. Igual, and O. Sander. An axiomatic approach to scalar data interpolation on surfaces. *Numerische Mathematik*, 102(3):383–411, 2006.
- A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.
- T. F. Chan and J. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001a.
- T. F. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2001b.
- Y. Chen and H. Sundaram. Estimating complexity of 2D Shapes. In *IEEE 7th Workshop on Multimedia Signal Processing*, pages 1–4, 2005.
- G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *Journal of Mathematical Imaging and Vision*, 24(3):307–326, 2006.
- A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.
- C. A. Deledalle, V. Duval, and J. Salmon. Non-local methods with shape-adaptive patches (NLM-SAP). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.
- L. Demanet, B. Song, and T. Chan. Image inpainting by correspondence maps: a deterministic approach. *Applied and Computational Mathematics*, 1100(99):217–250, 2003.

- M. Ebdelli, O. Le Meur, and C. Guillemot. Video inpainting with short-term windows: application to object removal and error concealment. *IEEE Transactions on Image Processing*, 24(10):3034–3047, 2015.
- A. A. Efros and T. K. Leung. Texture Synthesis by Non-parametric Sampling. In *Proceedings of the 7th IEEE International Conference on Computer Vision.*, volume 2, pages 1033–1038, 1999.
- N. Einecke and J. Eggert. A multi-block-matching approach for stereo. In *Intelligent Vehicles Symposium.*, pages 585–592. IEEE, 2015.
- S. Esedoglu, S. Ruuth, and R. Tsai. Threshold dynamics for shape reconstruction and disocclusion. In *IEEE International Conference on Image Processing*, volume 2, pages 502–505, 2005.
- S. Esedoglu, Y.-H. R. Tsai, et al. Threshold dynamics for the piecewise constant Mumford–Shah functional. *Journal of Computational Physics*, 211(1):367–384, 2006.
- S. Esedoglu, S. J. Ruuth, and R. Tsai. Threshold dynamics for high order geometric motions. *Interfaces and Free Boundaries*, 10(3):263–282, 2008.
- L. Euler. *Methodus inveniendi lineas curvas maximi minimive proprietate gaudentes sive solutio problematis isoperimetrici latissimo sensu accepti*, volume 24. Opera Omnia, 1744.
- L. C. Evans. Convergence of an algorithm for mean curvature motion. *Indiana University Mathematics Journal*, 42(2):533–557, 1993.
- G. Facciolo, R. Sadek, A. Bugeau, and V. Caselles. Temporally consistent gradient domain video editing. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 59–73. Springer, 2011.

- V. Fedorov. *Affine Invariant Image Comparison and Its Applications*. PhD. Thesis, 2016.
- V. Fedorov and C. Ballester. Affine non-local means image denoising. *IEEE Transactions on Image Processing*, 26(5):2137–2148, 2017.
- V. Fedorov, P. Arias, R. Sadek, G. Facciolo, and C. Ballester. Linear Multiscale Analysis of Similarities between Images on Riemannian Manifolds: Practical Formula and Affine Covariant Metrics. *SIAM Journal on Imaging Sciences*, 8(3):2021–2069, 2015.
- C. C. Fowlkes, D. R. Martin, and J. Malik. Local Figure/Ground Cues are Valid for Natural Images. *Journal of Vision*, 7(8):1–9, 2007.
- J. F. Garamendi and E. Schiavi. A Multiclass Anisotropic Mumford-Shah Functional for Segmentation of D-dimensional Vectorial Images. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, volume 4, pages 468–475, 2017.
- R. G. von Gioi. Toward a computational theory of perception. In *Proceedings of The Fifth Asia-Pacific Computing and Philosophy Conference*, pages 49–58, 2009.
- R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. LSD: a Line Segment Detector. *Image Processing On Line*, 2:35–55, 2012.
- M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. Background Inpainting for Videos with Dynamic Objects and a Free-Moving Camera. In *European Conference on Computer Vision*, pages 682–695, 2012a.
- M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How not to be seen? object removal from videos of crowded scenes. In *Computer Graphics Forum*, volume 31, pages 219–228, 2012b.

- S. Grewenig, S. Zimmer, and J. Weickert. Rotationally invariant similarity measures for nonlocal image denoising. *Journal of Visual Communication and Image Representation*, 22(2):117–130, 2011.
- R. Grzhibovskis and A. Heintz. A convolution thresholding scheme for the Willmore flow. *Interfaces and Free Boundaries*, 10(2):139–153, 2008.
- Y. Gu, W. Xiong, L. L. Wang, and J. Cheng. Generalizing Mumford-Shah model for multiphase piecewise smooth image segmentation. *IEEE Transactions on Image Processing*, 26(2):942–952, 2017.
- T. Hayashi and M. Sasaki. Contour Completion of Partly Occluded Skew-Symmetry Objects. In *IEEE International Symposium on Multimedia*, pages 90–93, 2014.
- P. A. van der Helm. Bayesian confusions surrounding simplicity and likelihood in perceptual organization. *Acta Psychologica*, 138(3):337–346, 2011.
- B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.
- N. Houhou, J.-P. Thiran, and X. Bresson. Fast texture segmentation based on semi-local region descriptor and active contour. *Numerical Mathematics: Theory, Methods and Applications.*, 2(EPFL-ARTICLE-140431):445–468, 2009.
- S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *IEEE Transactions on Image Processing*, 17(8):1443–1451, 2008.
- B. Jawerth and P. Lin. Shape recovery by diffusion generated motion. *Journal of Visual Communication and Image Representation*, 13(1):94–102, 2002.
- R. Jensen. Uniqueness of Lipschitz extensions: minimizing the sup norm of the gradient. *Archive for Rational Mechanics and Analysis*, 123(1):51–74, 1993.

- M. Jung. Piecewise-Smooth Image Segmentation Models with L1 Data-Fidelity Terms. *Journal of Scientific Computing*, 70(3):1229–1261, 2017.
- S. H. Kang, W. Zhu, and J. Jianhong. Illusory shapes via corner fusion. *SIAM Journal on Imaging Sciences*, 7(4):1907–1936, 2014.
- G. Kanizsa. *Organization in vision: essays on Gestalt perception*, volume 49. Praeger New York, 1979.
- G. Kanizsa. *Vedere e pensare*. Il Mulino, Bologna, 1991.
- N. Kawai, T. Sato, and N. Yokoya. Image inpainting considering brightness change and spatial locality of textures and its evaluation. In *Pacific-Rim Symposium on Image and Video Technology*, pages 271–282. Springer, 2009.
- P. J. Kellman and T. F. Shipley. A theory of visual interpolation in object perception. *Cognitive psychology*, 23(2):141–221, 1991.
- A. Kheradmand and P. Milanfar. A General Framework for Regularized, Similarity-Based Image Restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, 2014.
- D. C. Knill, D. Kersten, and A. Yuille. Introduction: A Bayesian formulation of visual perception. *Perception as Bayesian inference*, pages 1–21, 1996.
- K. Koffka. *Principles of Gestalt psychology*. London: Routledge and Kegan Paul, 1935.
- C. Kondermann, D. Kondermann, and C. Garbe. Postprocessing of optical flows via surface measures and motion inpainting. In *Joint Pattern Recognition Symposium*, pages 355–364. Springer, 2008.
- S. J. Koppal. Lambertian reflectance. In *Computer Vision*, pages 441–443. Springer, 2014.

- V. Lazcano. *Some Problems in Depth Enhanced Video Processing*. PhD. Thesis, 2016.
- E. Leeuwenberg and P. A. Van der Helm. *Structural Information Theory. The Simplicity of Visual Form*. Cambridge University Press, 2013.
- M. Leordeanu, A. Zanfır, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *IEEE International Conference on Computer Vision*, pages 1721–1728, 2013.
- R. Levien. The elastica: a mathematical history. *University of California, Berkeley, Technical Report No. UCB/EECS-2008-103*, 2008.
- F. Li, M. K. Ng, and C. Li. Variational fuzzy Mumford–Shah model for image segmentation. *SIAM Journal on Applied Mathematics*, 70(7):2750–2770, 2010.
- F. Li, S. Osher, J. Qin, and M. Yan. A multiphase image segmentation based on fuzzy membership functions and l1-norm fidelity. *Journal of Scientific Computing*, 69(1):82–106, 2016.
- H. Li, J. Cai, T. N. A. Nguyen, and J. Zheng. A benchmark for semantic image segmentation. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2013.
- R. J. van Lier, P. van Der Helm, and E. Leeuwenberg. Integrating global and local aspects of visual occlusion. *Perception*, 23(8):883–903, 1994.
- R. J. van Lier, P. van Der Helm, and E. Leeuwenberg. Competing global and local completions in visual occlusion. *Journal of Experimental Psychology: Human Perception and Performance*, 21(3): 571–583, 1995a.

- R. J. van Lier, E. L. Leeuwenberg, and P. A. van der Helm. Multiple completions primed by occlusion patterns. *Perception*, 24(7):727–740, 1995b.
- D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- J. J. Manfredi, A. M. Oberman, and A. P. Sviridov. Nonlinear elliptic partial differential equations and p-harmonic functions on graphs. *Differential Integral Equations*, 28(1-2):79–102, 2015.
- A. Mansfield, M. Prasad, C. Rother, T. Sharp, P. Kohli, and L. J. Van Gool. Transforming image completion. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2011.
- D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proceedings of the 8th International Conference on Computer Vision*, volume 2, pages 416–423, 2001.
- S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Transactions on Image Processing*, 11(2):68–76, 2002.
- S. Masnou and J. M. Morel. Level lines based disocclusion. In *Proceedings of the International Conference on Image Processing*, pages 259–263, 1998.
- Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, 28(7):1150–1163, 2006.
- N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

- A. Meijster, J. B. Roerdink, and W. H. Hesselink. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 331–340. Springer, 2002.
- M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.
- B. Merriman, J. K. Bence, and S. Osher. Diffusion generated motion by mean curvature. In *J.E. Taylor, editor, Computational Crystal Growers Workshop*, pages 73–83. American Mathematical Society, Providence, Rhode Island, 1992. Also available as UCLA CAM Report 92-18, April 1992.
- L. Modica and S. Mortola. Un esempio di γ -convergenza. *Bollettino dell'Unione Matematica Italiana*, 14(1):285–299, 1977.
- L. Moravec and J. Beck. Amodal completion: Simplicity is not the explanation. *Bulletin of the Psychonomic Society*, 24(4):269–272, 1986.
- J. M. Morel and S. Solimini. *Variational methods in image processing*. Birkhäuser, 1994.
- B. Mory and R. Ardon. Fuzzy region competition: A convex two-phase segmentation framework. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 214–226. Springer, 2007.
- D. Mumford. Elastica and Computer Vision. In *Algebraic Geometry and its Applications*, pages 491–506. Springer, New York, 1994.
- D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

- A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.
- M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, segmentation, and depth*, volume 662. Lecture notes in computer science, Springer, 1993.
- A. Oberman. A convergent difference scheme for the infinity Laplacian: construction of absolutely minimizing Lipschitz extensions. *Mathematics of Computation*, 74(251):1217–1230, 2005.
- I. R. Otero and M. Delbracio. Computing an Exact Gaussian Scale-Space. *Image Processing On Line*, 6:8–26, 2016.
- R. P. Palomares, G. Haro, and C. Ballester. A Rotation-Invariant Regularization Term for Optical Flow Related Problems. In *Asian Conference on Computer Vision*, pages 304–319. Springer, 2014.
- R. P. Palomares, E. Meinhardt-Llopis, C. Ballester, and G. Haro. Faldoi: A New Minimization Strategy for Large Displacement Variational Optical Flow. *Journal of Mathematical Imaging and Vision*, 58(1):27–46, 2017.
- K. A. Patwardhan, G. Sapiro, and M. Bertalmío. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007.
- J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.
- J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deep-matching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3):300–323, 2016.

- C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.
- M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proceedings of the IEEE Computer Society conference on Computer Vision and Pattern Recognition*, volume 2, pages 699–704, 2003.
- N. Rubin. The role of junctions in surface completion and contour matching. *Perception*, 30(3):339–366, 2001.
- J. Rubinstein, P. Sternberg, and J. B. Keller. Fast reaction, slow diffusion, and curve shortening. *SIAM Journal on Applied Mathematics*, 49(1):116–133, 1989.
- L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- R. Sadek, C. Constantinopoulos, E. Meinhardt, C. Ballester, and V. Caselles. On affine invariant descriptors related to SIFT. *SIAM Journal on Imaging Sciences*, 5(2):652–687, 2012.
- C. Sagiv, N. A. Sochen, and Y. Y. Zeevi. Integrated active contours for texture segmentation. *IEEE Transactions on Image Processing*, 15(6):1633–1646, 2006.
- P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.
- O. Sander, V. Caselles, and M. Bertalmío. Axiomatic scalar data interpolation on manifolds. In *Proceedings of the International Conference on Image Processing*, volume 102, pages 383–411, 2003.
- A. B. Sekuler. Local and global minima in visual completion: Effects of symmetry and orientation. *Perception*, 23(5):529–529, 1994.

- J. A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, 1985.
- T. K. Shih, N. C. Tang, and J. N. Hwang. Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(3):347–360, 2009.
- M. Singh. Modal and amodal completion generate different shapes. *Psychological Science*, 15(7):454–459, 2004.
- M. Singh and D. D. Hoffman. Completing visual contours: The relationship between relatability and minimizing inflections. *Perception & Psychophysics*, 61(5):943–951, 1999.
- E. Strelakovsky and D. Cremers. Real-time minimization of the piecewise smooth Mumford-Shah functional. In *European Conference on Computer Vision*, pages 127–141, 2014.
- M. Strobel, J. Diebold, and D. Cremers. Flow and color inpainting for video completion. In *German Conference on Pattern Recognition*, pages 293–304. Springer, 2014.
- J. H. Syu, S. J. Wang, and L. C. Wang. Hierarchical Image Segmentation based on Iterative Contraction and Merging. *IEEE Transactions on Image Processing*, 26(5):2246–2260, 2017.
- A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.
- M. Thorpe and F. Theil. Asymptotic analysis of the Ginzburg-Landau functional on point clouds. *arXiv:1604.04930 [math.AP]*, 2016.
- C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision*, pages 839–846, 1998.

- L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
- T. Viero, K. Oistamo, and Y. Neuvo. Three-dimensional median-related filters for color image sequence filtering. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(2):129–142, 1994.
- C. Vogel, S. Roth, and K. Schindler. An evaluation of data costs for optical flow. In *German Conference on Pattern Recognition*, pages 343–353. Springer, 2013.
- M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2464–2471, 2010.
- Y. Wexler, E. Shechtman, and M. Irani. Space-Time Completion of Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.
- L. R. Williams and D. W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural computation*, 9(4):837–858, 1997.
- Y. Xu, T. Géraud, and L. Najman. Hierarchical image simplification and segmentation based on Mumford–Shah salient level line selection. *Pattern Recognition Letters*, 83:278–286, 2016a.
- Z. Xu, Q. Zhang, Z. Cao, and C. Xiao. Video Background Completion Using Motion-Guided Pixel Assignment Optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(8):1393–1406, 2016b.
- C. Zach, T. Pock, and H. Bischof. A duality based approach for real-time TV-L1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

- L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3), 1965.
- M. H. Zhou, M. Mascagni, and A. Y. Qiao. Explicit finite difference schemes for the advection equation. *Relation*, 10(1.55):70–98, 1998.

