# Variational methods for exemplar-based image inpainting and gradient-domain video editing

Pablo Arias Martínez

TESI DOCTORAL UPF / 2013

Director de la tesi
Prof. Dr. Vicent Caselles,
Department of Information and Communication Technologies

The court's PhD was appointed by the recto of the Universitat Pompeu Fabra on ....., 2013.

Chairman

Member

Member

Member

Secretary

The doctoral defense was held on ....., 2013,
at the Universitat Pompeu Fabra and scored as .....

PRESIDENT MEMBERS

SECRETARY

*Para Mar*

# Agradecimientos

El trabajo presentado en esta tesis es resultado del esfuerzo compartido y la contribución de varias personas desde que comencé en setiembre del 2007. Quisiera expresar mi gratitud hacia ellas.

Primero que nada, quiero agradecer de manera especial a mi supervisor Vicent Caselles. Durante estos años, no sólo me ha guíado a través de los distintos temas que conforman esta tesis, sino que ha logrado canalizar mi motivación fluctuante y atención dispersa en un esfuerzo coherente. Todo esto con una gran paciencia a la que he puesto a prueba en varias oportunidades. He aprendido mucho durante el proceso. Por todo esto y más, siento una gran deuda para con Vicent.

También estoy muy agradecido a las personas participaron en el desarrollo de los trabajos expuestos en la tesis: Gabriele Facciolo, Rida Sadek y Guillermo Sapiro. Gabriele y Rida, con quienes además de colaborar compartí despacho, merecen un reconocimiento especial. No sólo que muchas ideas importantes para el desarrollo de esta tesis son suyas o fruto del diálogo con ellos, sino que también participaron en gran medida en llevar a cabo esas ideas. Trabajar con ellos fue muy gratificante y son para mí ejemplos a seguir por su entusiasmo, tenacidad y dedicación.

Un doctorado implica, además del desafío científico, un desafío burocrático comparable, y que no hubiera podido salvar sin la ayuda de las secretarias del DTIC, brindada siempre con gran paciencia, pedagogía, sentido del humor y un sentido excepcional del *timing* para cumplir todos los plazos. A todas y en particular a Lydia, muchas gracias.

Quisiera agradecer también a los colegas y amigos del departamento: Coloma Ballester, Felipe Calderero, Juan Calvo, Margarita Camacho, Juan Cardelino, Oscar Civit, Vadim Fedorov, Sira Ferradans, Lluís Garrido, Vanel Lazcano, Enric Meinhardt, Roberto Pablo Pérez Palomares, Edoardo Provenzi. To-

dos ellos contribuyen a hacer más ameno el día a día. Con Edoardo en particular tengo gran una deuda gastronómica y cultural.

Mudarme a Barcelona implicó alejarme de muchos amigos de Uruguay. A pesar de la distancia, siempre han estado cerca, dispuestos a escuchar y a brindarme su apoyo. A mis amigos y amigas del liceo, a los de La Pedrera, a los la facultad y del IIE y a los la Neumonía Atípica: muchas gracias.

Quiero agradecer también a Gregory Randall; su entusiasmo contagioso es en gran parte culpable de que me haya dedicado al procesamiento de imágenes.

Muchas gracias también a mi familia por su apoyo y compañía durante estos años a la distancia: a mis padres Ángel y María del Pilar, a mi hermana Sofía y a la Nona. También quisiera agradecer a mi abuelo Ignacio. Su memoria siempre ha sido para mí un ejemplo a seguir.

Finalmente quiero agradecer muy especialmente a mi compañera Gloria. Nunca podré devolver toda la comprensión, el cariño y la ayuda que me brindó durante todos estos años.

# Abstract

In this thesis we tackle two problems which deal with filling-in the information in a region of an image or a video, where the data is either missing or has to be replaced. These problems have applications in the context of image and video editing. The first is *image inpainting*, and aims at obtaining a visually plausible completion in a region in which data is missing due to damage or occlusion. The second problem concerns the propagation of an editing performed by a user in one or two reference frames of a video, throughout the rest of the video. Both problems are of theoretical interest since their analysis involves an understanding of the self-similarity in natural images and videos. At a high level, the common theme in both problems, is the exploitation and imposition of a model of redundancy (or self-similarity) to fill-in missing parts of a signal.

# Resumen

En esta tesis consideramos dos problemas que tratan el completado de una imagen o un video en una región en la que los datos o bien faltan (por ejemplo porque han sido dañados) o bien tienen que ser sustituídos. Estos problemas tienen aplicación en el contexto de edición de imagen y video. El primero de estos problemas es el de *inpainting* de imágenes, cuyo objetivo es dar una completación plausible en la region en la que faltan datos. El segundo problema consiste en la propagación de una edición proporcionada por un usuario en un cuadro de un video, a los demás cuadros. Además de su aplicación práctica, ambos problemas son de interés teórico, ya que su análisis requiere una compresión de la estructura de auto-similitud de las imagenes y videos naturales. De hecho, a nivel conceptual ambos temas tienen en común el cómo aprovechar e imponer la redundancia de una señal para completar partes faltantes.

# Preface

In this thesis we tackle two problems of data interpolation: image inpainting and keyframe-based video editing. Both problems deal with filling-in the information in a region of the image or video, where the data is either missing or has to be replaced.

Although both problems are related, we have addressed them using somewhat independent approaches, for reasons that will be explained soon. As a consequence, the thesis is divided into two parts which can be read independently.

The first part is devoted to the problem of *image inpainting*. The aim here is to obtain a visually plausible image interpolation in a region in which data is missing due to damage or occlusion. Usually, to solve this problem, the only available data is the image outside the region to be inpainted. It has become a standard tool for digital image retouching tasks, such as the removal of scratches in scanned old photographies, or dust spots in digital photography. In these cases the edited region is rather small. Recent advances in the field with the development of the so-called exemplar-based methods, allow major modifications of an image, such as removing entire objects and completing the part of the image that was occluded by the removed object. These image inpainting methods are sometimes referred to as *image completion*, *disocclusion* or *object removal*. We should stress that the aim of inpainting is not to recover the true occluded background, but to create a background which looks natural to a human observer.

Exemplar-based methods for inpainting exploit the self-similarity of images. Natural images tend to be highly redundant formed by repetitive geometric and texture patterns. Exemplar-based methods mimic this repetitive behaviour, filling the inpainting domain replicating the same patterns present in the rest of the image. Roughly speaking, they work by copying small pieces of the image and pasting them in the inpainting domain in a coher-


**Figure 1: Inpainting problem.** On the left: the original image. In the center: the result of one of the exemplar-based inpainting scheme derived from the variational framework presented in this thesis. On the right: the red curves show the boundaries between regions that have been copied from a corresponding region in $O^c$.

ent manner. The work we present in the first part of thesis is a variational framework for these type of inpainting methods.

In the second part of the thesis, we propose a variational method for video editing. The term video editing applies to a variety of tasks ranging from basic operations such as trimming, cutting, splitting and resizing video segments to more sophisticated ones such as modifying the appearance of the objects in a scene, or more generally, removing or adding objects in a video segment.

In this thesis, we focus on following video editing problem. We are given an image sequence where an object's surface has been edited in one or two reference frames, and a certain editing domain. The objective is then to propagate the edited information in the reference frame (or frames) throughout the editing domain, following the motion of the objects' surface.

The problem that we tackle is a rather specific one, but in fact it is a direct application of a fundamental problem in video editing, namely the propagation of color/intensity information through motion trajectories, in a temporally and spatially consistent manner.

Temporal consistency refers to a smooth transition between successive frames, coherent with the motion in the sequence. Objects in a video are expected to move following smooth trajectories. This is a very strong a priori of the human visual system, and as a consequence any artifact causing sudden changes or temporal jitter in a video is immediately perceived and highly distracting. Therefore, the editing of a video cannot be reduced to a series of independent image editing problems. The temporal interdependence between frames, imposed by the motion in the sequence, has to be taken into

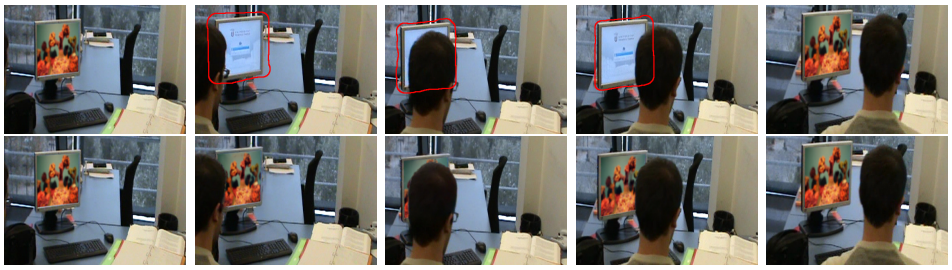


**Figure 2:** An example of an input and output of the video editing application. The sequence has 20 frames in total. In the first row a user has edited the first and the last frames of the sequence. In the remaining frames, we would like to propagate the edited information inside the area delimited by the red curve (the editing domain). In the second row, we show the output obtained using the proposed model.

account. This implies that of the motion in the editing domain has to be known (or rather estimated).

Spatial consistency refers to a seamless integration of the editing with its spatial surrounding in each frame. In the image editing literature, spatial consistency is usually addressed using gradient-domain methods. These are based on the manipulation of the image gradients instead of its grey levels. The modified gradients are then integrated to recover the resulting image, typically by solving a Poisson equation. This procedure prevents the appearance of seams at the boundaries of the edited region.

In order to achieve both temporal and spatial consistency, we propose an energy functional to propagate gradient-domain information along motion trajectories.

This problem appears as a building block of other more complex video editing tasks, such as video inpainting for object removal. The added difficulty in video inpainting, is that the motion of the dis-occluded background is unknown, and due to the temporal consistency constraint, it has to be somehow interpolated. The problem of motion inpainting and interpolation is also a very interesting problem, but is not in the scope of this thesis.

Besides their practical applications, the problems addressed in this thesis are of theoretical interest since their analysis involves an understanding of the sources of self-similarity present in natural images and videos. Indeed at a high level, the common theme in both problems, is the exploitation and imposition of a model of redundancy (or self-similarity) to fill-in missing parts of a signal. Both approaches are different due to the different nature

of the type of redundancy considered in each case.

In the case of video the main source of redundancy is the temporal consistency. It is a notion of redundancy related to the constancy of an object's appearance with respect to the evolution of time, which is a continuous parameter. Therefore we can express temporal consistency with a local formulation: two snapshots of a pixel at successive time stamps of its trajectory have to be similar. These local formulations of self-similarity are naturally modelled by local differential operators, which lead to PDEs that propagate or interpolate the data at the key-frames along trajectories.

On the other hand, in still images, the main source of self-similarity is given by the repetition of features and textures at different locations of an image. This locations might be far apart in the image domain, as in a periodic signal the repetitions are separated by integer multiples of the period. Therefore, modelling this non-local self-similarity requires different techniques, which search the available part of the image for pieces that can be coherently assembled to fill-in the inpainting domain. Once found, these pieces are directly copied from their source locations without any propagation happening.

# Contents

**Abstract** ix

**Resumen** ix

**Preface** xi

**I Exemplar-based image inpainting** 1

**1 Introduction** 3
1.1 Previous work . . . 3
1.2 Contributions . . . 9
1.3 Notation . . . 12

**2 Variational framework** 13
2.1 Preliminaries . . . 13
2.2 Proposed formulation: similarity weights . . . 14
2.3 Proposed formulation: getting a correspondence . . . 19

**3 Minimization of the energies** 23
3.1 Alternating optimization scheme . . . 23
3.2 Equivalence with EM . . . 30
3.3 Convergence of the alternating optimization scheme . . . 31

**4 Observations on the energies and their minima** 35
4.1 Effect of the selectivity parameter . . . 35
4.2 Regularity of optimal correspondence maps . . . 38
4.3 Local minima . . . 42
4.4 Decoupling image and weights update steps . . . 44
4.5 Extensions . . . 48

**5 Discussion and connections** **51**
5.1 Interpretation in the space of patches . . . . . 51
5.2 Connections with statistical mechanics . . . . . 55
5.3 Deterministic annealing . . . . . 58
5.4 Revisiting related work . . . . . 62

**6 Implementation** **69**
6.1 Discrete setting and notation . . . . . 69
6.2 Image update step . . . . . 71
6.3 Computation of the Nearest Neighbor Field . . . . . 76
6.4 Multiscale scheme . . . . . 85

**7 Experimental results** **89**
7.1 Experimental setting . . . . . 89
7.2 Observations and comments . . . . . 91

**8 Conclusions and future work** **99**

**Appendices of Part I** **101**

**A Existence and regularity of minima** **103**

**B Optimal correspondence maps** **119**

**C Convergence of the alternating optimization scheme** **129**

**II Gradient-based video editing** **133**

**9 Introduction** **135**
9.1 Problem definition . . . . . 135
9.2 Previous work . . . . . 136
9.3 Our contribution . . . . . 139

**10 Deriving the model** **141**
10.1 Models for temporal consistency . . . . . 141
10.2 Proposed energy in the continuous setting . . . . . 144
10.3 Boundary conditions . . . . . 147
10.4 Euler-Lagrange equation . . . . . 151

**11 Discrete model** **153**
11.1 Discretization of operator $\boldsymbol{\nabla}\partial_{\boldsymbol{v}}$ . . . . . 153

11.2 The discrete energy . . . 154
11.3 Definition of the operators . . . 155
11.4 Treatment of occluded pixels . . . 160
11.5 The discrete Euler-Lagrange equation . . . 162

**12 A deblurring scheme for the convective derivative 165**
12.1 A motivating example . . . 165
12.2 The DSCD: a mixed scheme . . . 168
12.3 Scope and limitations of the previous analysis . . . 171
12.4 The discrete energy with the DSCD . . . 176

**13 Experimental results 181**
13.1 One-lid setting . . . 183
13.2 Two-lid setting . . . 187
13.3 Limitations of the proposed method . . . 190

**14 Conclusions and future work 199**

**Appendices of Part II 201**

**D An occlusion detection method 203**

**E The Euler-Lagrange equation 209**

**F Analytic solutions of the Euler-Lagrange equation 215**

**Bibliography 221**

# Part I

# Exemplar-based image inpainting

Chapter 1

# Introduction

## 1.1 Previous work

Most inpainting methods found in the literature can be classified into two groups: *geometry-* and *texture-oriented* methods depending on how they characterize the redundancy of the image . We now briefly review the developments in both types of approaches, with emphasis in texture-oriented methods. This review will be helpful for motivating the proposed formulation.

### Geometry-oriented methods

In this class of methods images are usually modeled as functions with some degree of smoothness, expressed for instance in terms of the curvature of the level lines or the total variation of the image. This smoothness assumption is exploited to interpolate the inpainting domain by continuing the geometric structure of the image (its level lines, or its edges), usually as the solution of a (geometric) variational problem or by means of a partial differential equation (PDE). Such PDE can be derived from variational principles, as for instance in Masnou and Morel (1998); Ballester et al. (2001); Chan et al. (2002); Chan and Shen (2001); Esedoglu and Shen (2002); Masnou (2002), or inspired by phenomenological modeling Bertalmío et al. (2000); Bornemann and März (2007); Tschumperlé and Deriche (2005). These methods show good performance in propagating smooth level lines or gradients, but fail in

the presence of texture. They are often referred to as *structure* or *cartoon* inpainting.

In most cases, geometry-oriented methods are *local* in the sense that they are based on PDEs. An implication of this is that among all the data available from the image, they only use that at the boundary of the inpainting domain.

### Texture-oriented methods

Texture-oriented inpainting was born as an application of texture synthesis, *e.g.* Efros and Leung (1999); Igehy and Pereira (1997). Its recent development was triggered in part by the works of Efros and Leung (1999) and Wei and Levoy (2000) using non-parametric sampling techniques (parametric models have also been considered, *e.g.* Levin et al. (2003)). In these works texture is modeled as a two dimensional probabilistic *graphical model*, in which the value of each pixel is conditioned by its neighborhood. These approaches rely directly on a sample of the desired texture to perform the synthesis. The value of each target pixel $x$ is copied from the center of a (square) patch in the sample image, chosen to match the available portion of the patch centered at $x$. See Levina and Bickel (2006) for a probabilistic theoretical justification.

This strategy (with various modifications) has been extensively used for inpainting Bertalmío et al. (2003); Bornard et al. (2002); Criminisi et al. (2004); Drori et al. (2003); Efros and Leung (1999); Pérez et al. (2004). As opposed to geometry-oriented inpainting, these so-called *exemplar-based* approaches, are *non-local*: to determine the value at $x$, the whole image may be scanned in the search for a matching patch.

As pointed out in Demanet et al. (2003) the problem of exemplar-based inpainting can be stated as that of finding a *correspondence map* $\varphi : O \to O^c$, which assigns to each point $x$ in the inpainting domain $O$ (a subset of the image domain $\Omega$, usually a rectangle in $\mathbb{R}^2$) a corresponding point $\varphi(x) \in O^c := \Omega \setminus O$ where the image is known (see Figure 1.2). The unknown part of the image is then synthesized using the map $\varphi$. The filling-in strategy of Efros and Leung (1999); Wei and Levoy (2000) can be regarded as a *greedy* procedure (each hole pixel is visited only once) for computing a *correspondence map*. The results obtained are very sensitive to the order in which the pixels are processed Criminisi et al. (2004); Pérez et al. (2004); Harrison (2005).

To address this issue, in Demanet et al. (2003) the authors proposed to model the inpainting problem as the minimization of an energy functional in which the unknown is the correspondence map itself:

$$\mathcal{E}(\varphi) = \int_O \int_{\Omega_p} |\hat{u}(\varphi(x+h)) - \hat{u}(\varphi(x)+h)|^2 dh dx, \tag{1.1}$$

where $\hat{u} : O^c \to \mathbb{R}$ is the known part of the image, and $\Omega_p$ is the patch domain (a neighborhood of the origin $0 \in \mathbb{R}^2$). The unknown image is computed as $u(x) = \hat{u}(\varphi(x))$, $x \in O$. Thus $\varphi$ should map a pixel $x$ and its neighborhood in such a way that the resulting patch is close to the one centered at $\varphi(x)$. This model has been the subject of further analysis by Aujol *et al.* Aujol et al. (2010), proposing extensions and proving the existence of a solution in the set of piecewise roto-translation maps, *i.e.* maps of the form

$$\varphi(x) = \sum_{i \in I} R_i(x - c_i) \mathbb{1}_{A_i}(x),$$

where $\{A_i\}_{i \in I}$ is a Caccioppoli partition of $O$ (i.e. all sets of the partition have finite perimeter in $O$ and the sum of the perimeters is finite), and for each $i \in I$ $R_i$ is a rotation matrix and $c_i$ is a translation vector. $\mathbb{1}_{A_i}(x) = 1$ if $x \in A_i$ and zero otherwise.

The energy (1.1) is highly non-convex and no effective way to minimize it is known Aujol et al. (2010). Hence, other authors have addressed the determination of a correspondence map by looking for simpler optimization problems.

For example, Komodakis and Tziritas (2007) compute a coarse correspondence map by formulating the problem as a probabilistic inference on a lattice, in which the nodes correspond to square blocks of pixels covering the inpainting domain. Using *loopy belief propagation* (a message passing inference algorithm, see Pearl (1998)) the authors efficiently compute a coarse correspondence map over the lattice. The size of the blocks and their separation is such that each pixel in the inpainting domain is covered by four blocks. Thus, strictly speaking, this method computes, for each pixel, four correspondences. These have to be somehow fused to get the inpainted image. In this sense, the method departs from model (1.1). However, as we show later, a minor modification of the energy minimized in Komodakis and Tziritas (2007) falls as a coarse case of model (1.1) in which the correspondence map is constrained to be block-wise a translation.

Another optimization strategy is followed in Kawai et al. (2009); Wexler et al. (2007). In both works the variable to be optimized is the unknown

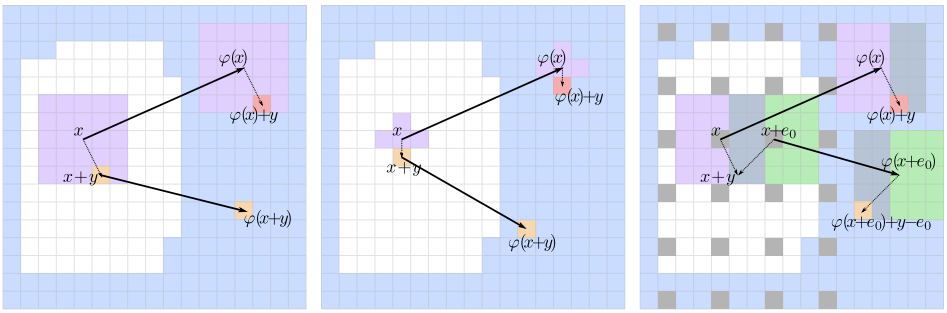
image whereas the correspondence map appears as an auxiliary variable. The resulting energy functional can be regarded as a relaxation of (1.1):

$$\mathcal{E}(u,\varphi) = \int_{\widetilde{O}} \int_{\Omega_p} |u(x+h) - \hat{u}(\varphi(x)+h)|^2 dh dx, \tag{1.2}$$

where $\widetilde{O} := O + \Omega_p$ denotes the set of centers of patches that intersect the inpainting domain $O$ (see Figure 1.2). The energy is usually optimized using an alternating scheme with respect to the variables $u$ and $\varphi$, and the unknown image is determined as part of the optimization process, and is not constrained to be $u(x) = \hat{u}(\varphi(x))$. Although this relaxation is still non-convex, the alternating minimization scheme converges to a critical point of the energy. This approach was also used in the context of texture synthesis Kwatra et al. (2005).

In Pritch et al. (2009) the authors present a framework for several image editing tasks, with inpainting among them. Key to their framework is what they call the *shift map* $t : \widetilde{O} \to \mathbb{Z}^2$, which can be defined in terms of the correspondence map as follows $t(x) = \varphi(x) - x$. For the inpainting application, they minimize energy (1.1) for the smallest possible patch domain $\Omega_p$: the center of the patch plus its four immediate neighbors in the discrete image rectangular lattice. The minimization is seen as a gaph labelling problem on the lattice, where the labels are the shifts $t$ (or equivalently, the positions on the known portion of the image). Energy (1.1) corresponds to a pairwise Markov Random Field and is minimized using graph cuts. To reduce the computational burden due to the large number of labels, the authors adopt a hierachical multiscale approach. An initial correspondence map is computed at a coarse scale and refined in subsequent scales. In this refinement steps, labels correspond to small perturbations of the coarse map. In this way, the number of labels is reduced. The resulting scheme works well if an approximate solution can be computed at a coarse scale, however it is not able to recover from large errors in the coarse scale. In Liu and Caselles (2013) they alleviate this problem by including fine scale features in the coarse scale.

An interesting approach was proposed by He and Sun (2012). The authors analize the statistics of the offsets to the nearest neighbor between known patches. Only the most frequent offsets are then used as labels for the model of Pritch et al. (2009). This results in faster running times, and also in better results for images with mostly periodic frotal textures.




**(a)** DSC model **(b)** Shift map model **(c)** KT model

**Figure 1.1:** Comparison of variational inpainting models. (a) The model (1.1) in Demanet et al. (2003) minimizes the error between the patch centered at $x$, and the one centered at $\varphi(x)$. The yellow pixel $x + y$ is copied from $\varphi(x + y)$, and has to be similar to the pixel the same relative position in the patch centered at $\varphi(x)$. (b) Shift Map is a particular case of energy (1.1), in which the patch consists of a pixel and its four neighbors. (c) The work of Komodakis and Tziritas (2007) minimizes the overlap error of patches at a coarse grid (grey pixels). Pixel $x + y$ does not have a correspondence.

Exemplar-based methods provide impressive results in recovering textures and repetitive structures. However, their ability to recreate the geometry without any example is limited and not well understood. Different strategies have been proposed for combining geometry and texture inpainting. Some rely on human intervention for constraining the geometry Sun et al. (2005). Others usually decompose the image in structure and texture components. The structure is reconstructed using some geometry-oriented scheme, and this is used to guide the texture inpainting Bertalmío et al. (2003); Cao et al. (2011); Drori et al. (2003); Jia and Tang (2004).

### Non-local image regularization

Let us finally note that the works in texture synthesis of Efros and Leung (1999); Wei and Levoy (2000) have also influenced the development of non-local methods for other applications, such as denoising Awate and Whitaker (2006); Buades et al. (2005), superresolution Protter et al. (2009) and regularization of inverse problems Gilboa and Osher (2008); Peyré et al. (2011). As opposed to the case of inpainting, in these contexts the estimation of a

pixel value may involve many locations in the image. The resulting correspondence is not one-to-one, but rather one-to-many, usually encoded as a weight function $w : \Omega \times \Omega \to \mathbb{R}$, with $\Omega$ being the image domain. For each $x$, $w(x, \cdot)$ weights the contribution of each image location to the estimation of $x$. Inspired by regularization techniques used in the context of graphs or discrete data and trying to formulate the non-local means denoising method Buades et al. (2005); Awate and Whitaker (2006) as a variational model, Gilboa and Osher Gilboa and Osher (2008, 2007) proposed the following functional

$$\mathcal{E}(u) = \int_\Omega \int_\Omega w(x, y)(u(x) - u(y))^2 dy dx \tag{1.3}$$

which can be considered as a non-local version of the Dirichlet integral. The weights $w$ are considered as known. The minimum of (1.3) should have a low pixel error $(u(x) - \hat{u}(y))^2$ whenever $w(x, y)$ is high. When the weights are *Gaussian*

$$w(x, y) \propto \exp\left(-\frac{1}{T}\|p_u(x) - p_{\hat{u}}(y)\|^2\right), \tag{1.4}$$

the non-local means algorithm results from the first step of Jacobi's iterative method for solving the Euler-Lagrange equation of (1.3). Here, $\|\cdot\|$ is a weighted $L^2$-norm in the space of patches and $T$ is a parameter that determines the selectivity of the weights $w$.

Other variational approaches for non-local denoising have also been proposed in Kindermann et al. (2006); Brox et al. (2008); Peyré et al. (2011); Pizarro et al. (2010).

## 1.2 Contributions

Despite all the efforts for combining them, geometry and texture inpainting are still quite separate fields, each one with its own analysis and implementation tools. Variational models as the one introduced in this paper provide common tools and thus might help in unifying of both trends. We therefore propose a variational framework for non-local image inpainting as a contribution to the modeling and analysis of texture-oriented methods.

The main contributions of this work are the following:

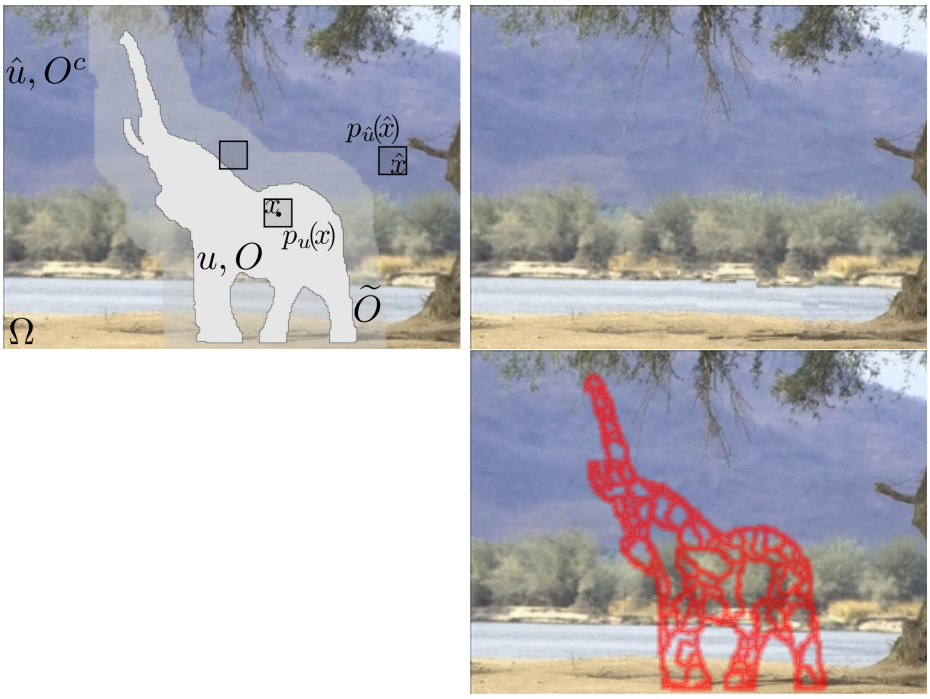



**Figure 1.2: Inpainting problem.** Top, left: on a rectangular image domain $\Omega$, missing data $u$ in a region $O$ has to be reconstructed using the available image $\hat{u}$ over $O^c := \Omega \setminus O$. A patch centered at $x \in O$ is denoted by $p_u(x)$. The set of centers of incomplete patches is $\widetilde{O} := O + \Omega_p$, where $\Omega_p$ denotes the patch domain. Top right: the image shows a completion obtained using the *patch NL-medians*, a scheme derived from the general formulation presented in this work. Bottom: the resulting completion shows a correspondence map which is a piece-wise translation. The red curves show the boundaries between the regions of constant translation. In each of these regions, the image is copied rigidly from a corresponding region in $O^c$.

### The variational formalism

We propose a variational framework for exemplar-based image inpainting (Chapter 2). The proposed formulation can be seen as a generalization of (1.2) by considering different patch similarity criteria other than the squared $L^2$-norm. Via the selection of the patch similarity criterion different inpainting schemes can be naturally derived. In this work we discuss four of them, *patch NL-means*, *-medians*, *-Poisson* and *-gradient medians*, corresponding to similarity criteria based on $L^2$- and $L^1$-norms between patches or their

gradients. The *patch NL-means* is related to the inpainting methods of Kawai et al. (2009); Wexler et al. (2007). Methods related to the the *patch NL-Poisson* and *patch NL-medians* have been used in Kwatra et al. (2005) in the context of texture synthesis.

The proposed inpainting models encode self-similarity as a correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$. But we show that such models can be obtained as a Gamma limit of a broader class of non-local regularization models in which the image redundancy and self-similarity is encoded by a non-local weight function $w : \widetilde{O} \times \widetilde{O}^c \to \mathbb{R}$, which serves as a probabilistic correspondence. This probabilistic correspondence model has the form of a Gibbs free energy, where a parameter $T > 0$ controls the entropy of the weights. The resulting weights $w$ for $T > 0$ corresponds to the exponential weights of Eq. (1.4) used for non-local denoising and regularization Awate and Whitaker (2006); Buades et al. (2005). When $T \to 0$, we recover the correspondence maps inpainting energy. Heuristically, one can say that the optimal similarity weights $w$ converge to $\delta(y - \varphi(x))$ where $\varphi : \widetilde{O} \to \widetilde{O}^c$ is an optimal correspondence map for (1.2).

The case of $T > 0$ is not well suited for the inpainting application (it results in a blurry completion), but it is still insightful to present our inpainting energies in the broader context of the more general non-local regularization framework. This allows us to relate existing models for non-local regularization with exemplar-based inpainting and to provide intuitive interpretations of the energy. In particular we discuss the connections of our model and the Gibbs free energy in the context of statistical mechanics. Interestingly, similar energies have been used to approach the problems of quantization and clustering (Rose (1998)) and point matching by a rigid transormation (Rangarajan et al. (1997)). These problems, at first sight seemingly unrelated, are all connected by the fact that they require the computation of an unknown correspondence map. Furthermore, the non-local regularization framework is still interesting in itself, since it can be applied to other inverse problems problems in which the similarity weights are unknown and have to be estimated together with the image (Facciolo et al. (2009), Peyré et al. (2011)).

We provide a theoretical study of two models contained in the general Gibbs energy formulation ($T > 0$): the patch NL-means, and the patch NL-Poisson model. For both of them we prove the existence of minima and their regularity. For the Gamma limit as $T \to 0$ of the patch NL-means energy, we prove the existence of optimal solutions of the limit functional which are

measurable correspondence maps which also minimize (1.2).

The energy for $T > 0$ is introduced in Sections 2.1 and 2.2 and in Section 2.3 we derive the inpainting energy as the limit case when $T = 0$. The connections with related models in inpainting, regularization and statistical mechanics approaches to clustering and point matching are described on Chapter 5.

### The inpainting schemes

The inpainting is performed as an iterative minimization process alternating between weight computation and image synthesis steps, described in Section 3.1. These steps depend on the patch metric. As opposed to the non-local means denoising algorithm in which only the central pixel from each exemplar is used, we use the whole patch. This causes some blur due to patch overlap, but provides stability and convergence in fewer iterations. For $T > 0$, we prove the convergence to critical points of the alternating optimization scheme with respect to the variables $u$ and $w$ (which coincides with the Expectation-Maximization (EM) algorithm) for both models, patch NL-means and -Poisson (see Sections 3.2 and 3.3).

In the limit case $T = 0$, when the fuzzy correspondence $w$ converges to a dense correspondence map, this iterative process generates a sort of patch work, as the one shown in Fig. 1.2. The inpainting domain is partitioned into arbitrarily shaped segments which show an exact copy (of image or gradient values) of some region in the hole's complement. Transitions between the copied segments take place in a band around the boundary between the segments. The width of the band is the size of the patch. The four inpainting schemes differ in the way this blending is done (and in the partition found). Methods based on the $L^2$-norm perform a smooth blending, whereas those based on the $L^1$-norm favor sharper transitions. In Sections 4.1, 4.2, 4.3 and Chapter 7 we provide a comprehensive comparison on real and synthetic problems, showing the benefits and limitations of the four schemes studied.

In particular, both gradient-based methods, *patch NL-Poisson* and *-gradient medians*, combine the exemplar-based interpolation with PDE-based diffusion schemes. This results in a smoother continuation of the information across the boundary and inside the inpainting domain, and in a better propagation of structures. Furthermore, the inclusion of gradients in the patch similarity criterion allows to handle additive brightness changes.

Of particular interest, is the regularity of the correspondence map obtained. In Section 4.2 we prove a mild regularity result, namely the existence of optimal correspondence maps $\varphi$ which are uniform limits of maps of bounded variation with finitely many values. This result is interesting in connection with the experimental observation that the computed $\varphi$ copies rigidly parts of the image outside the inpainting domain, behaving locally as a translation. This observation is also at the root of the roto-translation constraint imposed to $\varphi$ in Aujol et al. (2010).

### Implementation: convergence analysis of PatchMatch Barnes et al. (2009) algorithm

The most time consuming step in the minimization of the inpainting energies is the computation of the optimal matching between patches in the inpainting domain and patches in the region of available data. Recently Barnes *et al.* Barnes et al. (2009) introduced the PatchMatch, an efficient algorithm based on heuristics to solve the problem of matching patches between images. In Section 6.3 we prove its convergence in probability and we compute a bound on its convergence rate.

## 1.3 Notation

Images are denoted as functions $u : \Omega \to \mathbb{R}$, where $\Omega$ denotes the image domain, a rectangle in $\mathbb{R}^N$. We will commonly refer to points in $\Omega$ as pixels. These will be denoted by $x$, $\hat{x}$, $z$, $\hat{z}$ or $y$, the latter for positions inside the patch. A patch of $u$ centered at $x$ is denoted by $p_u(x) := p_u(x, \cdot) : \Omega_p \to \mathbb{R}$, where $\Omega_p$ is a rectangle centered at 0. The patch is defined by $p_u(x, y) := u(x + y)$, with $y \in \Omega_p$. Let $O \subset \Omega$ be the hole or inpainting domain, and $O^c = \Omega \setminus O$. We assume that $O$ is an open set with Lipschitz boundary. We still denote by $\hat{u} : O^c \to R$ the known part of the image $u$: $\hat{u} := u|_{O^c}$.

Let us define some domains in $\Omega$ that are useful to work with patches. We denote by $\widetilde{\Omega}$ the set of centers of patches contained in the image domain, i.e. $\widetilde{\Omega} = \{x \in \Omega : x + \Omega_p \subseteq \Omega\}$. As was defined in the Introduction, we take $\widetilde{O}$, the *extended inpainting domain*, as the set of centers of patches that intersect the hole, *i.e.* $\widetilde{O} := O + \Omega_p = \{x \in \Omega : (x + \Omega_p) \cap O \neq \emptyset\}$. For a simplified presentation, we assume that $\widetilde{O} \subseteq \widetilde{\Omega}$, *i.e.* every pixel in

$\widetilde{O}$ is the center of a patch contained in $\Omega$. We denote $\widetilde{O}^c = \widetilde{\Omega} \setminus \widetilde{O}$. Thus, patches $p_{\tilde{u}}(y)$ centered at points $y \in \widetilde{O}^c$ are contained in $O^c$ (see Figure 1.2). Further notation will be introduced in the text.

Chapter 2

# Variational framework

In this chapter we present our variational framework building on top of previous models for image regularization with fuzzy correspondence maps. The proposed scheme can be seen as a generalization of the model (1.2) of Kawai et al. (2009); Kwatra et al. (2005); Wexler et al. (2007) with a probabilistic correspondence. As will be discribed in Section 2.3, the correspondence case can be recovered as a limit case of our framework.

## 2.1 Preliminaries

Our variational framework is inspired by the following non-local functional

$$\mathcal{F}_w(u) = \int_O \int_{O^c} w(x,\hat{x})(u(x) - \hat{u}(\hat{x}))^2 d\hat{x}dx \tag{2.1}$$

which comes as a straightforward adaptation of (1.3) to the inpainting problem. In this case $w : O \times O^c \to \mathbb{R}^+$ is a weight function that measures the similarity between patches centered in the inpainting domain and in its complement. Let us assume for the moment that the weights are known. As in the case of denoising, the minimization of (2.1) enforces a low pixel error $(u(x) - \hat{u}(\hat{x}))^2$ whenever the similarity $w(x,\hat{x})$ is high. In this way the similarity weights drive the information transfer from known to unknown pixels.

In Gilboa and Osher (2007) the weights are considered as known and remain fixed through all the iterations. While this might be appropriate in appli-

cations where they can be estimated from the noisy image, in the image inpainting scenario here addressed, the weights are not available and have to be inferred together with the image (as in Peyré et al. (2008); Protter et al. (2009)).

One of the novelties of the proposed framework is the inclusion of adaptive weights in a variational setting, considering the weight function $w$ as an additional unknown. Instead of prescribing explicitly the Gaussian functional dependence of $w$ w.r.t. $u$, we will do it implicitly, as a component of the optimization process. In doing so, we obtain a simpler functional, avoiding to deal with the complex, non-linear dependence between $w$ and $u$.

In our formulation, we will constrain $w(x, \cdot)$ to be a probability density function, which can be seen as a relaxation of the one-to-one correspondence map of Aujol et al. (2010); Demanet et al. (2003), providing a fuzzy correspondence between each $x \in O$ and the complement of the inpainting domain.

## 2.2 Proposed formulation: similarity weights

In this setting, we propose an energy which contains two terms, one of them is inspired by (2.1) and measures the coherence between patches in $\widetilde{O}$ and those in $\widetilde{O}^c$, for a given similarity weight function $w : \widetilde{O} \times \widetilde{O}^c \to \mathbb{R}$. This permits the estimation of the image $u$ from the weights $w$. The second term allows us to compute the weights given the image. The complete proposed functional is

$$\begin{aligned} &\mathcal{E}_{E,T}(u, w) = \mathcal{U}_E(u, w) - T\mathcal{H}(w), \\ &\text{subject to } \int_{\widetilde{O}^c} w(x, \hat{x}) d\hat{x} = 1, \ \forall x \in \widetilde{O}, \end{aligned} \tag{2.2}$$

where

$$\mathcal{U}_E(u, w) = \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x, \hat{x}) E(p_u(x) - p_{\hat{u}}(\hat{x})) d\hat{x} dx, \tag{2.3}$$

and $E(\cdot)$ is an error function for image patches (such as the squared $L^2$-norm). The second term is given by

$$\mathcal{H}(w) = -\int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x, \hat{x}) \log w(x, \hat{x}) d\hat{x} dx.$$

Except for a constant term, $\mathcal{H}(w)$ corresponds to the entropy of a joint probability over $\widetilde{O} \times \widetilde{O}^c$, given by $|\widetilde{O}|^{-1} w(x, y)$. Alternatively, we can also

interpret $\mathcal{H}(w)$ as the integral of the entropies of each individual probability $w(x,\cdot)$ (the inner integrals over $\widetilde{O}^c$).

We observe that the term $(u(x)-\hat{u}(\hat{x}))^2$ in (2.1), that penalizes differences between pixels, is substituted in (2.3) by the patch error function $E(p_u(x)-p_{\hat{u}}(\hat{x}))$. This has two implications.

The first is that minimizing (2.3) with respect to the image will force patches $p_u(x)$ and $p_{\hat{u}}(\hat{x})$ to be similar whenever $w(x,\hat{x})$ is high. The second implication has to be understood together with the inclusion of the second term, which integrates the entropy of each probability $w(x,\cdot)$ over $\widetilde{O}$. For a given completion $u$, and for each $x \in \widetilde{O}$, the optimum weights minimize the mean patch error for $p_u(x)$, given by

$$\int_{\widetilde{O}^c} w(x,\hat{x})E(p_u(x)-p_{\hat{u}}(\hat{x}))d\hat{x},$$

while maximizing the entropy. This can be understood as a minimization of the mean patch error subject to a level of randomness, measured by the entropy: The energy $\mathcal{E}_{E,T}$ can be seen as a Lagragian and $T$ corresponds to the Lagrange multiplier multiplying the entropy constraints.

An equivalent interpretation can be given in terms of the principle of maximum entropy (Jaynes (1957)), widely used for inference of probability distributions: if we fix the level of mean patch error $U_E$ and want to estimate the probability distribution, we may choose between all probability distributions the one that maximizes the entropy. In that case, the Lagrangian would be given by $\frac{1}{T}\mathcal{E}_{E,T}$. According to the principle of maximum entropy, the best representation for a distribution given a set of samples is the one that maximizes the entropy, *i.e.* the distribution which makes the less assumptions about the process.

The resulting weights are given by

$$w_{E,T}(u;x,\hat{x}) := \frac{1}{Z_{E,T}(u;x)} \exp\left(-\frac{1}{T}E(p_u(x)-p_{\hat{u}}(\hat{x}))\right),$$

with the normalizing factor $Z_{E,T}(u;x)$

$$Z_{E,T}(u;x) := \int_{\widetilde{O}^c} \exp\left(-\frac{1}{T}E(p_u(x)-p_{\hat{u}}(\hat{x}))\right) d\hat{x}. \tag{2.4}$$

Taking $E$ as the squared $L^2$-norm of the patch, then the resulting weights are Gaussian, given by formula (1.4). The parameter $T$ in (2.2) controls the

trade-off between both terms and is also the selectivity parameter of the Gaussian weights.

Let us describe in detail some patch error functions of interest in practice.

### The patch error function $E$

Patches are functions defined on $\Omega_p$, and if $\mathbb{P}$ denotes a suitable space of patches, we consider error functions $E : \mathbb{P} \to \mathbb{R}^+$ defined either as the weighted sum of pixel errors

$$E(p_u(x) - p_{\hat{u}}(\hat{x})) := g * \mathrm{e}(u(x+\cdot) - \hat{u}(\hat{x}+\cdot)) \\ = \int_{\Omega_p} g(h)\mathrm{e}(u(x+h) - \hat{u}(\hat{x}+h))dh,$$

where $\mathrm{e} : \mathbb{R} \to \mathbb{R}^+$, or gradient errors

$$E(p_u(x) - p_{\hat{u}}(\hat{x})) := g * \mathrm{e}(\nabla u(x+\cdot) - \nabla \hat{u}(\hat{x}+\cdot)) \\ = \int_{\Omega_p} g(h)\mathrm{e}(\nabla u(x+h) - \nabla \hat{u}(\hat{x}+h))dh.$$

where $\mathrm{e} : \mathbb{R}^2 \to \mathbb{R}^+$. Here, $g : \mathbb{R}^N \to \mathbb{R}^+$ denotes a suitable *intra-patch* kernel function. We assume that $g \in L^1(\mathbb{R}^N)^+$(*i.e.* $g$ is a nonnegative integrable function in the $L_1$ sense) and $\int_{\mathbb{R}^N} g(h)\, dh = 1$. As an example, one could take $g(h) = \mathcal{N}(h|0, a\boldsymbol{I})$, the Gaussian probability density function with 0 mean and isotropic standard deviation $a$. In our mathematical statements we will consider a function $g$ with compact support in $\mathbb{R}^N$. We will consider four patch error functions.

**Patch non-local means.** In this case we use $\mathrm{e}(r) = |r|^2$ and the patch error function is a weighted squared $L^2$-norm that we denote by

$$E_2(p_u(x) - p_{\hat{u}}(\hat{x})) = \|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,2} = g * |u(x+\cdot) - \hat{u}(\hat{x}+\cdot)|^2.$$

$\mathbb{P}$ can be taken as the set of $L^2$ functions in $\Omega_p$.

**Patch non-local medians.** If we set $\mathrm{e}(r) = |r|$ then the patch error function results in a weighted $L^1$-norm:

$$E_1(p_u(x) - p_{\hat{u}}(\hat{x})) = \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,1} = g * |u(x+\cdot) - \hat{u}(\hat{x}+\cdot)|.$$

$\mathbb{P}$ can be taken as the set of $L^1$ functions in $\Omega_p$.

**Patch non-local Poisson.** Let us take $\mathbb{P}$ as the space $W^{1,2}(\Omega_p)$, and consider pixel errors $\mathrm{e}(\,\cdot\,)$ that are a convex combination of intensity and gradient errors. Thus, for $\lambda \in [0,1)$ we define the following patch error function:

$$\begin{aligned} E_{\lambda,2}(p_u(x) - p_{\hat{u}}(\hat{x})) = \\ \lambda\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 + (1-\lambda)\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2,\nabla}^2 = \\ g * \left(\lambda|u(x+\cdot) - \hat{u}(\hat{x}+\cdot)|^2 + (1-\lambda)|\nabla u(x+\cdot) - \nabla\hat{u}(\hat{x}+\cdot)|^2\right). \end{aligned}$$

Note that if we set $\lambda = 1$ we get $E_2$.

**Patch non-local gradient medians** We take $\mathbb{P}$ as the space of square integrable functions of bounded variation in $\Omega_p$ Ambrosio et al. (2000). As in the previous case, we will consider a convex combination of intensity and gradient errors with coefficient $\lambda \in [0,1)$, but with a $L_1$ norm for the gradient component:

$$\begin{aligned} E_{\lambda,1}(p_u(x) - p_{\hat{u}}(\hat{x})) = \\ \lambda\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 + (1-\lambda)\|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,1,\nabla} = \\ g * \left(\lambda|u(x+\cdot) - \hat{u}(\hat{x}+\cdot)|^2 + (1-\lambda)|\nabla u(x+\cdot) - \nabla\hat{u}(\hat{x}+\cdot)|\right). \end{aligned}$$

In an abuse of notation we have used $\nabla$ to denote the distributional derivative.

By plugging each of these patch error functions in the energy $\mathcal{E}_{E,T}$ we get different inpainting functionals. Let us define the following notational shorthands to refer to them:

$$\mathcal{E}_{2,T} := \mathcal{E}_{E_2,T}, \quad \mathcal{E}_{1,T} := \mathcal{E}_{E_1,T}, \quad \mathcal{E}_{2,\lambda,T} := \mathcal{E}_{E_{2,\lambda},T} \quad \text{and} \quad \mathcal{E}_{1,\lambda,T} := \mathcal{E}_{E_{1,\lambda},T}.$$

We use similar shorthands for $\mathcal{U}_E$ (*e.g.* $\mathcal{U}_{2,\lambda} = \mathcal{U}_{E_{2,\lambda}}$). As will be discussed below, the patch error function determines not only the similarity criterion but also the image synthesis, and thus is a key element in the framework.

The last two patch error functions take the gradient of the image into account. Let us mention that the use of nonlocal energies with gradient terms for deblurring and denoising problems has been proposed in Kindermann et al. (2006).

## Existence and regularity of minimizers

In this section we present results stating the existence of minimizers for the patch NL-means $\mathcal{E}_{2,T}$, patch NL-medians $\mathcal{E}_{1,T}$ and patch NL-Poisson $\mathcal{E}_{2,\lambda,T}$ functionals. The corresponding proofs can be found in Appendix A.

Let us recall that $\Omega$ is a rectangle in $\mathbb{R}^N$ and that $u : \Omega \to \mathbb{R}$ is such that $u|_{O^c} = \hat{u}$. We assume that $\hat{u} : O^c \to \mathbb{R}$ with $\hat{u} \in L^\infty(O^c)$. We also assume that $u$ is extended by symmetry and then by periodicity to $\mathbb{R}^N$.

Let us introduce some notation. Let $C_c(\mathbb{R}^N)$ be the set of continuous functions with compact support in $\mathbb{R}^N$ and by $C_c(\mathbb{R}^N)^+$ we denote the set of nonnegative functions in $C_c(\mathbb{R}^N)$. We also recall the notation for Sobolev functional spaces. As usual, if $Q$ is an open set we denote by $W^{1,p}(Q)$, $1 \leq p \leq \infty$, the space of functions $v \in L^p(Q)$ such that $\nabla v \in L^p(Q)^N$. By $W^{1,p}(Q)^+$ we denote the set of nonnegative functions in $W^{1,p}(Q)$. We denote by $W^{2,p}(Q)$ (resp. by $W^{2,p}_{\text{loc}}(Q)$), $1 \leq p \leq \infty$, the space of functions $v \in L^p(Q)$ such that $\nabla v \in L^p(Q)^N$ and $D^2 v \in L^p(Q)^{N\times N}$ (resp. the functions $v \in W^{2,p}(Q')$ for any subdomain $Q'$ included in a compact set of $Q$).

We define the admissible class of functions

$$\mathcal{A}_{2,T} := \{(u, w) : u \in L^\infty(\Omega),\, u = \hat{u} \text{ in } O^c, w \in \mathcal{W}\},$$

where $\mathcal{W} := \left\{ w \in L^1(\widetilde{O} \times \widetilde{O}^c) : \int_{\widetilde{O}^c} w(x, y)\, dy = 1 \text{ a.e. } x \in \widetilde{O} \right\}$.

The following propositions state the existence of smooth minima and smooth probability distributions representing the fuzzy correspondences between $\widetilde{O}$ and $\widetilde{O}^c$, both for patch NL-means and patch NL-medians models.

**Proposition 2.1** (Minimizers of patch NL-means.)**.** *We assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. Then there exists a minimum $(u, w) \in \mathcal{A}_{2,T}$ of $\mathcal{E}_{2,T}$. Moreover, for any minimum $(u, w) \in \mathcal{A}_{2,T}$ we have that $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

**Proposition 2.2** (Minimizers of patch NL-medians.)**.** *Under the same hypothesis as in the previous proposition, there exists a minimum $(u, w) \in \mathcal{A}_{2,T}$ of $\mathcal{E}_{1,T}$. For any minimum $(u, w) \in \mathcal{A}_{2,T}$ we have that $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

Let us now consider the existence of minimizers for the patch NL-Poisson model $\mathcal{E}_{2,\lambda,T}$. For simplicity we set $\lambda = 0$. Similar results hold for $\lambda \in [0, 1)$. In this case, the set of admissible solutions is defined as follows:

$$\mathcal{A}_{2,0,T} := \{(u, w) \in \mathcal{A}_{2,T} : u \in W^{1,2}(O),\ u|_{\partial O} = \hat{u}|_{\partial O^c}\}.$$

**Proposition 2.3** (Minimizers of patch NL-Poisson.)**.** *We assume that $\hat{u} \in W^{2,2}(O^c) \cap L^\infty(O^c)$ and $g \in W^{1,\infty}(\mathbb{R}^N)^+$ has compact support in $\Omega_p$. There exists a minimum $(u, w) \in \mathcal{A}_{2,0,T}$ of $\mathcal{E}_{2,\lambda,T}$ with $\lambda = 0$. Moreover for any solution $(u, w) \in \mathcal{A}_{2,0,T}$ we have $u \in W^{1,2}(O) \cap W^{2,p}_{loc}(O) \cap L^\infty(O)$ for all $p \in [1, \infty)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

As argued in Appendix A, exitence also holds for the patch NL-gradient medians under suitable hypothesis on $\hat{u}$ and $g$ (Lema A.9).

## 2.3 Proposed formulation: getting a correspondence

Our variational framework for inpainting stems from the adaptation of a denoising model. As a consequence, our model considers similarity weights, which can be seen as a probabilistic correspondence map. The spread of this probabilistic weights is controlled by $T$, the weight given to the entropy term.

The results in this section prove that, as intuition suggests, when $T \to 0$ the energy $\mathcal{E}_{E,T}$ converges to an energy $\mathcal{E}_{E,0}$ which has minimizers that correspond to correspondence maps. This limit case is the most relevant case for the application of inpainting. The convergence results only consider patch NL-means and patch NL-Poisson. Although we do not prove analogous results patch NL-medians or patch NL-gradient medians, we will still define energies with $T = 0$ for patch NL-medians and patch NL-gradient medians (and implement the resulting schemes.

For this limit case, we require a more general definition for the similarity weights $w$. The mapping $x \mapsto w(x, \cdot)$ defines for each $x$ a probability density function over $\widetilde{O}^c$. As a consequence, it only models probability measures that have a density (with respect to the Lebesgue measure) in $\widetilde{O}$. In the limit case when $T = 0$ we need to consider as well other probability measures over $\widetilde{O}$ for which there is no density function.

**Definition 2.4** (Measurable measure-valued map)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^N$, $\mathcal{Y} \subseteq \mathbb{R}^M$ be open sets, $\mu$ be a positive Radon measure in $\mathcal{X}$ and $x \mapsto \nu_x$ be a function that assigns to each $x$ in $\mathcal{X}$ a Radon measure $\nu_x$ on $\mathcal{Y}$. We say that the map is $\mu$-measurable if $x \mapsto \nu_x(B)$ is $\mu$-measurable for any Borel set $B$ in $\mathcal{Y}$. In the case that $\nu_x(\mathcal{Y}) = 1$ $\mu$-a.e. in $\mathcal{X}$, then we say that $\nu$ is a measurable probability-valued map.*

A correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$ can be represented by a probability-valued map with all probabilities $\nu_x$ given by a Dirac's delta function on $\varphi(x)$. Indeed, $\varphi$ is a measurable map if and only if $x \in \widetilde{O} \mapsto \nu_x = \delta_{\varphi(x)}(y)$ is measurable probability-valued map. We denote by $\nu^\varphi$ the measure determined by the correspondence map $\varphi$.

We now state the Gamma limit of the model when $T \to 0$.

**Proposition 2.5.** *For the patch NL-means and patch NL-Poisson models, when $T \to 0$ the energy $\mathcal{E}_{E,T}$ Gamma-converges to the energy*

$$\mathcal{E}_{E,0}(u, \nu) := \int_{\widetilde{O}} \int_{\widetilde{O}^c} E(p_u(x) - p_{\hat{u}}(\hat{x}))d\nu_x(\hat{x})dx. \tag{2.5}$$

*where $x \mapsto \nu_x$ is a measurable probability valued map, with respect to the Lebesgue measure restricted to $\widetilde{O}$. In particular, the minima of $\mathcal{E}_{E,T}$ converge to minima of $\mathcal{E}_{E,0}$.*

This functional has several minimizers w.r.t. $\nu$. Any measurable probability-valued map $\nu$, such that

$$\operatorname{supp}(\nu_x) \subseteq \arg\min_{\hat{x} \in \widetilde{O}^c} E(p_u(x) - p_{\hat{u}}(\hat{x}))$$

is a minimizer. In particular, there exist minima of the energy $\mathcal{E}_{E,0}$ of the form $(u, \nu^\varphi)$ for a measurable correspondence map $\varphi$. More precisely, let us consider the case of patch NL-means and define the space of admissible solutions

$$\mathcal{A}_{2,0} := \{(u, \nu) : u \in L^\infty(\Omega), u = \hat{u} \text{ in } O^c, \nu \in \mathcal{MP}\},$$

where $\mathcal{MP}$ the set of measurable measure-valued maps $\nu \geq 0$ in $\widetilde{O} \times \operatorname{cl}(\widetilde{O}^c)$ with respect to the Lebesgue measure restricted to $\widetilde{O}$. The following proposition holds.

**Proposition 2.6.** *We assume that $g \in C_c(\mathbb{R}^N)$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. There exists a minimum $(u^*, \nu^*) \in \mathcal{A}_{2,0}$ of $\mathcal{E}_{2,0}$ such that $\nu^* = \nu^\varphi$ where $\varphi : \widetilde{O} \to \mathrm{cl}(\widetilde{O}^c)$ is a measurable map.*

For these minima of (2.5)

$$\varphi(x) \in \arg\min_{\hat{x} \in \widetilde{O}^c} E(p_u(x) - p_{\hat{u}}(\hat{x})),$$

*i.e.* $\varphi(x) \in \widetilde{O}^c$ is the center of a nearest neighbor of $p_u(x)$, with respect to the error function $E$ (we use the term nearest neighbor even though $E$ does not need to be a metric).

In the case of patch NL-means, the energy $\mathcal{E}_{2,0}$ can be seen as a relaxation of similar existing models Kawai et al. (2009); Wexler et al. (2007). Indeed, if we restrict the probability-valued map $\nu$ to be of the form $\nu^\varphi$, for some measurable correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$, then we can express the energy directly in terms of $\varphi$

$$\mathcal{E}_{E,0}(u, \varphi) = \int_{\widetilde{O}} E(p_u(x) - p_{\hat{u}}(\varphi(x)))dx. \tag{2.6}$$

Equation (1.2) is obtained as a particular case when $E$ is the squared $L^2$-norm. An equivalent formulation has been proposed by Peyré Peyré (2009), where the energy is interpreted as a regularization model based on the distance to the manifold of known patches.

In Kwatra et al. (2005) two models are presented for its application to texture synthesis: a gradient-based model and a robust model. Both correspond to particular cases of (2.6) using a $L^2$ gradient-based norm and a $L^{0.8}$-norm respectively.

In the following, for a unified presentation, we will use the notation $w$ for the weights both when $T > 0$ and when $T = 0$.

Chapter 3

# Minimization of the energies

We have formulated the inpainting problem as the constrained optimization

$$\begin{aligned} (u^*, w^*) &:= \arg\min_{u,w} \mathcal{E}_{E,T}(u, w), \\ \text{subject to } & u(x) = \hat{u}(x),\ \forall x \in O^c, \\ & \int_{\widetilde{O}^c} w(x, \hat{x}) d\hat{x} = 1, \end{aligned} \tag{3.1}$$

where $\mathcal{E}_{E,T}$ is the inpainting energy defined in (2.2).

In this chapter we first describe an alternating optimization algorithm to compute local minima of problem (3.1). We will consider $T \geqslant 0$. As we show in Section 3.2, the alternating optmization scheme can be related to the Expectation-Maximization algorithm (EM) widely used in statistics for maximum likelihood estimation (Dempster et al. (1957)). This leads to an interesting interpretation of the model, as the maximization of a probability on the space of patches, induced by the known patches of the image, that will be discussed in Section 5.1. In Section 3.3 we summarize results on the convergence of the scheme for the patch NL-means and patch NL-Poisson models.

## 3.1 Alternating optimization scheme

The objective $\mathcal{E}_{E,T}$ (3.1) is non-convex, and we can only compute a local minima. To that aim, we use an alternating minimization algorithm. At

each iteration, two optimization steps are solved: the constrained minimization of $E$ with respect to $w$ while keeping $u$ fixed; and the minimization of $E$ with respect to $u$ with $w$ fixed. This procedure yields the iterative scheme of Algorithm 1.

**Algorithm 1** Alternate minimization of $\mathcal{E}_{E,T}(u, w)$.

**Require:** Initial condition $u_0(x)$ with $x \in O$, tolerance $\tau > 0$.

1: **repeat**

2: $w_k = \arg\min_w \mathcal{E}_{E,T}(u_k, w)$, subject to $\int_{\widetilde{O}^c} w_k(x, \hat{x}) d\hat{x} = 1, \ \forall x \in \widetilde{O}$.

3: $u_{k+1} = \arg\min_u \mathcal{E}_{E,T}(u, w_k)$, subject to $u_{k+1}(x) = \hat{u}(x), \ \forall x \in O^c$.

4: **until** $\|u_{k+1} - u_k\| < \tau$.

### Weights update step for the case $T > 0$

For the weights update step, we keep $u$ fixed and minimize (2.2) with respect to $w$. For $T > 0$, the weights can be computed by defining the Lagrangian

$$\mathcal{L}(w) = \mathcal{E}_{E,T}(u, w) + \int_{\widetilde{O}} \lambda(x) \left( \int_{\widetilde{O}^c} w(x, \hat{x}) d\hat{x} - 1 \right) dx.$$

From the Euler-Lagrange equation $\delta_w \mathcal{L}(w) = 0$, and after normalizing the weights, we obtain

$$w_{E,T}(u; x, \hat{x}) := \frac{1}{Z_{E,T}(u; x)} \exp\left( -\frac{1}{T} E(p_u(x) - p_{\hat{u}}(\hat{x})) \right),$$

where the normalizing factor $Z_{E,T}(u; x)$ is given by

$$Z_{E,T}(u; x) := \int_{\widetilde{O}^c} \exp\left( -\frac{1}{T} E(p_u(x) - p_{\hat{u}}(\hat{x})) \right) d\hat{x}. \tag{3.2}$$

### Weights update step for the case $T = 0$

In this case the weights are actually a measurable probability-valued map and we minimize the following energy:

$$\mathcal{E}_{E,0}(u, \nu) := \int_{\widetilde{O}} \int_{\widetilde{O}^c} E(p_u(x) - p_{\hat{u}}(\hat{x})) d\nu_x(\hat{x}) dx. \tag{3.3}$$

This functional has several minimizers w.r.t. $\nu$. In practice, we compute a minimizer that is a correspondence map, by setting $\nu_x = \delta_{\varphi(x)}$ where $\varphi(x) \in \widetilde{O}^c$ is the center a nearest neighbor of $p_u(x)$, *i.e.*

$$\varphi(x) \in \operatorname*{arg\,min}_{\hat{x} \in \widetilde{O}^c} E(p_u(x) - p_{\hat{u}}(\hat{x})).$$

This is supported by Proposition 2.6 which states that (for patch NL-means and patch NL-Poisson), there are minimizers in which the measurable probability map $\nu$ is associated to a measurable correspondence map.

### Image update step

In this section we present the derivation of the image update step corresponding to the four patch error functions mentioned earlier. First we will present the cases when image patches are compared using the squared $L^2$-norm and the $L^1$-norm.

Before moving to the derivation of the these schemes, let us remark that with the change of variables $z := x + h$, $\hat{z} := \hat{x} + h$, the image energy term can be expressed as an accumulation of pixel errors:

$$\begin{aligned}\mathcal{U}_E(u) &= \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x, \hat{x}) \int_{\Omega_p} g(h) \mathrm{e}(u(x+h) - \hat{u}(\hat{x}+h)) dh d\hat{x} dx \\ &= \int_{O} \int_{O^c} m(z, \hat{z}) \mathrm{e}(u(z) - \hat{u}(\hat{z})) d\hat{z} dz + \mathrm{C}, \end{aligned} \tag{3.4}$$

where C is a constant term. We have introduced the *pixel influence weights* $m : O \times O^c \to \mathbb{R}^+$ defined as

$$\begin{aligned} m(z, \hat{z}) &:= g * (w(z - \cdot, \hat{z} - \cdot)) \\ &= \int_{\Omega_p} g(h) w(z - h, \hat{z} - h) dh. \end{aligned} \tag{3.5}$$

Here redefine the weights by extending their domain to $\mathbb{R}^N \times \mathbb{R}^N$, *i.e.* $w : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^+$, by setting $w(x, \hat{x}) = 0$ outside of $\widetilde{O} \times \widetilde{O}^c$.

For each pair of pixels $(z, \hat{z}) \in O \times O^c$, $m(z, \hat{z})$ weights the effective contribution of the pixel error between $u(z)$ and $\hat{u}(\hat{z})$ in the total value of the energy. The quantity $m(z, \hat{z})$ is computed by integrating the similarity $w(z-h, \hat{z}-h)$ between all patches that overlap $\hat{z}$ and those that overlap $z$ in

the *same relative position* (shown in Fig. 3.1). Note that the energy (3.4) corresponds to equation (2.1), with the patch similarity weights $w$ being substituted with the pixel influence weights $m$, a spacial convolution of $w$ with kernel $g$ (as in Kindermann et al. (2006); Pizarro et al. (2010)).

Let us also define the following:

$$k(w;z) := \int_{O^c} m(z,\hat{z})d\hat{z} = \int_{O^c} g * (w(z-\cdot,\hat{z}-\cdot))d\hat{z} = 1, \tag{3.6}$$

given that the weights $w$ and $g$ are normalized. Thus, for each $z$ $m(z,\cdot)$ can also be interpreted as a probability density function. Although $k(w;z) = 1$, we will keep the notation $k(w;z)$ in the following derivations. Later, in Section 4.5 we propose modifications of the energy for which $k(w;z)$ may vary.

For $T = 0$, assuming that $\nu = \nu^{\varphi}$ for some measurable correspondence map $\varphi$, the pixel influence weights result in

$$\begin{aligned} m(z,\hat{z}) &= \int_{\Omega_p} g(h)\chi_{\widetilde{O}}(z-h)\delta_{\varphi(z-h)}(\hat{z}-h)dh \\ &= \int_{\Omega_p} g(h)\chi_{\widetilde{O}}(z-h)\delta_{\varphi(z-h)+h}(\hat{z})dh. \end{aligned} \tag{3.7}$$

The characteristic function of $\widetilde{O}$, $\chi_{\widetilde{O}}(x)$, is 1 if its argument is in $\widetilde{O}$ and 0 otherwise. Its purpose in the above formula is to zero out the terms for which $z-h$ falls out of $\widetilde{O}$ since $\varphi$ is only defined in $\widetilde{O}$.

### Patch non-local means

If we use the weighted squared $L^2$-norm $\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,2}$ as a patch error function $E$ in (2.3), then the image energy term (3.4) is quadratic on $u$. Its minimum for fixed weights $w$ can be computed explicitly as a non-local average of the known pixels:

$$u(z) = \frac{1}{k(w;z)} \int_{O^c} m(z,\hat{z})\hat{u}(\hat{z})\mathrm{d}\hat{z}, \tag{3.8}$$

for $z \in O$. The weights in the average are the pixel influence weights $m$, obtained by convolving the Gaussian similarity weights with the patch kernel $g$. To obtain more insight about this, let us expand $m$ to obtain:

$$u(z) = \frac{1}{k(w;z)} \int_{\Omega_p} g(h) \int_{\widetilde{O}^c} w(z-h,\hat{x})\hat{u}(\hat{x}+h)d\hat{x}dy.$$

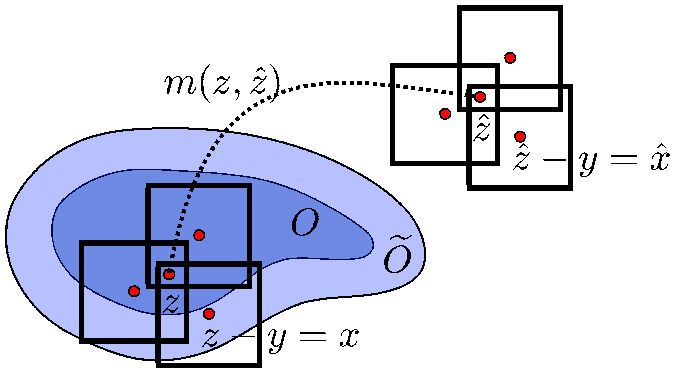



**Figure 3.1: Patch non-local inpainting.** The value at $z \in O$ is computed using contributions from all the patches that overlap it, *i.e.* those centered at $x \in O$ such that $z = x + h$ with $h \in \Omega_p$. The influence function $m(z, \hat{z})$ accumulates all contributions $w(z - h, \hat{z} - h)$ from patches centered at $\hat{z} - h$ to $z - h$.

There are two averaging processes involved. The outer integral goes through all patches $p_u(z - h)$ overlapping the target pixel $z$. Each patch suggests a value for $z$ resulting from the inner sum: a non-local average of the pixel at position $h$ in all patches $p_{\hat{u}}(\hat{x})$ in $\widetilde{O}^c$. This sum is weighted by the similarity between the patch $p_u(z - h)$ and each $p_{\hat{u}}(\hat{x})$.

Therefore, we can distinguish two types of pixel interactions. Interactions due to the patch overlap of nearby pixels in the image lattice and non-local interactions driven by the similarity weights. The latter can be controlled by the selectivity parameter $T$, but the extent of the overlap interactions is given by the patch size.

In particular for $T = 0$, equation (3.8) yields

$$u(z) = \int_{\Omega_p} g(h)\hat{u}(\varphi(z - h) + h)dh.$$

This blending may cause some blur, which leads us to consider the $L^1$-norm in the search of a more robust image synthesis.

### Patch non-local medians

We consider the $L^1$-norm patch error function in the image energy term, which corresponds to taking $\mathrm{e}(x) = |x|$ in (3.4). The Euler equation for $u$,

given the influence function $m$, can be formally written as

$$[\delta_u \mathcal{E}_{E,T}(u)](z) = \int_{O^c} \operatorname{sign}[u(z) - \hat{u}(\hat{z})] m(z, \hat{z}) d\hat{z} \ni 0.$$

This expression is multivalued, since $\operatorname{sign}(r) := r/|r|$ if $|r| > 0$ and $\operatorname{sign}(r) \in [-1, 1]$ if $r = 0$. Its solution for each $u(z), z \in O$ is obtained as a weighted median of the pixels of the complement $O^c$, with weights $m(z, \cdot)$.

Both schemes presented so far perform inpainting by transferring (by averages or medians) known gray levels into the inpainting domain. As we will see next, using a patch error function based on the gradient of the image yields methods which transfer gradients and compute the resulting image as the solution of a PDE. This results in better continuation properties of the solution, in particular at the boundary of the inpainting domain.

### Patch non-local Poisson

The patch NL Poisson method combines squared weighted $L^2$-norms of intensity and gradients. Expressing the resulting image energy in terms of the $m$ weights yields

$$\mathcal{U}_{\lambda,2}(u) = \int_O \int_{O^c} m(z, \hat{z})[(1-\lambda)|\nabla u(z) - \nabla \hat{u}(\hat{z})|^2 + \lambda(u(z) - \hat{u}(\hat{z}))^2] d\hat{z} dz \tag{3.9}$$

Notice that (3.9) can be rewritten as

$$\mathcal{U}_{\lambda,2}(u) \propto \int_O k(w; z)|\nabla u(z) - \boldsymbol{v}(w; z)|^2 dz + \frac{\lambda}{1-\lambda} \int_O k(w; z)(u(z) - f(w; z))^2 dz + C,$$

where $C$ is a constant term,

$$f(w; z) := \frac{1}{k(w; z)} \int_{O^c} m(z, \hat{z}) \hat{u}(\hat{z}) d\hat{z}$$

is the solution of the patch NL-means image update step and the field $\boldsymbol{v}(w; \cdot) : O \to \mathbb{R}^N$ is given by

$$\boldsymbol{v}(w; z) := \frac{1}{k(w; z)} \int_{O^c} m(z, \hat{z}) \nabla \hat{u}(\hat{z}) d\hat{z}. \tag{3.10}$$

This energy balances two terms. The first one imposes $u$'s gradient to be close (in the $L^2$ sense) to a *guiding* vector field $\boldsymbol{v}(w;\cdot)$ computed as a non-local weighted average of the image gradients in the known portion of the image. The second term correspondos to a quadratic attachment to the solution of the patch NL-means image update.

In this case the Euler-Lagrange equation w.r.t. $u$ becomes:

$$\begin{cases} \operatorname{div}[k(w;z)\nabla u(z)] - \dfrac{\lambda}{(1-\lambda)}k(w;z)u(z) = \\ \qquad\qquad \operatorname{div}[k(w;z)\boldsymbol{v}(w;z)] - \dfrac{\lambda}{(1-\lambda)}k(w;z)f(w;z), & z \in O, \\ u(z) = \hat{u}(z), & z \in \partial O \setminus \partial\Omega, \\ \nabla u(z)\cdot n_\Omega(z) = 0, & z \in \partial O \cap \partial\Omega. \end{cases} \tag{3.11}$$

Here $n_\Omega(z) \in \mathbb{R}^N$ denotes the outgoing normal to $\Omega$, for $z \in \partial\Omega$. The problem is linear and can be solved for instance with a conjugate gradient scheme.

Observe that when $\lambda = 0$ the resulting PDE is a Poisson equation. In the energy above, the attachment to $f(w,\cdot)$ vanishes, and only gradients are transfered to the inpainting domain. In this case, the patch similarity weights $w$ are computed based only on the gradients (and thus also the pixel influence weights $m$). In most cases however, the gradient is not a good feature for measuring the patch similarity, and it is convenient to consider also the gray level/color data. Typically we will set $\lambda \leqslant 0.1$, in this way we include some intensity information in the computation of the weights, without departing too much from the Poisson equation.

### Patch non-local gradient medians

Lastly, we consider the $L^1$-norm of the gradient. As before, the image energy can be written as follows

$$\mathcal{U}_{\lambda,1}(u) \propto \int_O \int_{O^c} m(z,\hat{z})|\nabla u(z) - \nabla\hat{u}(\hat{z})|d\hat{z}dz + \frac{\lambda}{1-\lambda}\int_O k(w;z)(u(z) - f(w;z))^2 dz + C,$$

with the same boundary conditions as with the patch NL-Poisson. When $\lambda > 0$, we minimize this energy using the fixed point algorithm described in Section 6.2, based on the projection method presented in Chambolle (2004).

When $\lambda = 0$, the resulting energy can be minimized with an implicit gradient descent. Given an iterate $u^t$, the next is computed as

$$\min_{u^{t+1}} \int_O \int_{O^c} m(z,\hat{z}) |\nabla u^{t+1}(z) - \nabla \hat{u}(\hat{z})| d\hat{z} dz + \frac{1}{2\delta t} \|u^{t+1} - u^t\|^2. \quad (3.12)$$

This energy for $u^{t+1}$ is of the same kind as $\mathcal{U}_{\lambda,1}$ for $\lambda > 0$. Thus, at each step $u^{t+1}$ is computed using the same fixed point algorithm described in the Sectio6.2.

As patch NL-Poisson, this scheme transfers gradients and interpolates the gray levels using the boundary conditions. With the use of the $L^1$ error function, we expect the solution of patch NL-GM to retain more small scale detail than that of patch NL-Poisson.

## 3.2 Equivalence with EM

Let us notice that the iterations of the alternating optimization algorithm coincide with the Expectation Maximization algorithm (EM). Indeed, the weights update corresponds the E-step, while the image update is the M-step. Let us denote the optimal weights for a given $u$ as

$$w_{E,T}(u;x,\hat{x}) = \frac{1}{Z_{E,T}(u;x)} \exp\left(-\frac{1}{T}\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,E}\right), \quad (3.13)$$

where $Z_{E,T}(u;x)$ is the corresponding normalization factor. Then we may write the energy $\mathcal{E}_{E,T}$ for a given $(u,w)$ as

$$\frac{1}{T}\mathcal{E}_{E,T}(u,w) = \int_{\widetilde{O}} \mathrm{KL}(w_{E,T}(u;x,\cdot), w(x,\cdot))\, dx - \mathcal{L}_E(u),$$

where

$$\mathcal{L}_E(u) = \int_{\widetilde{O}} \log\, Z_{E,T}(u;x)\, dx \quad (3.14)$$

corresponds to the so called *marginal likelihood* in the context of EM. The alternating optimization algorithm converges (modulo subsequences) to stationary points of $\mathcal{L}_E(u)$.

Note that

$$-T\mathcal{L}_E(u) = \mathcal{E}_{E,T}(u, w_{E,T}(u)) \leq \mathcal{E}_{E,T}(u,w) \quad \forall (u,w)$$

and

$$\min_{(u,w)} \mathcal{E}_{E,T}(u,w) = \min_{u} \min_{w} \mathcal{E}_{E,T}(u,w) = \min_{u} \mathcal{E}_{E,T}(u, w_{E,T}(u)) = \min_{u} -T\mathcal{L}_E(u).$$

Thus, the functional $\mathcal{E}_{E,T}(u,w)$ is equivalent to $-\mathcal{L}_E(u)$ in the sense that both have the same minima.

The alternating optimization algorithm converges to a critical point of both of them. More precisely:

**Proposition 3.1.** *Any limit point of the sequence $\{u^k\}_{k=0}^{\infty}$ defined by Algorithm 2, is a stationary point $u^*$ of $\mathcal{L}_E(u)$ and $\mathcal{L}_E(u^k)$ converges monotonically to $\mathcal{L}_E(u^*)$.*

*Proof.* Indeed,

$$\begin{aligned}\mathcal{L}_E(u^{k+1}) - \mathcal{L}_E(u^k) = & \\ & -\frac{1}{T}\mathcal{E}_{E,T}(u^{k+1}, w_{E,T}(u^k)) + \frac{1}{T}\mathcal{E}_{E,T}(u^k, w_{E,T}(u^k)) \\ & + \int_{\widetilde{O}} \mathrm{KL}(w_{E,T}(u^{k+1}; x, \cdot), w_{E,T}(u^k; x, \cdot))\, dx \\ & - \int_{\widetilde{O}} \mathrm{KL}(w_{E,T}(u^k; x, \cdot), w_{E,T}(u^k; x, \cdot))\, dx \\ & \geq -\frac{1}{T}\mathcal{E}_{E,T}(u^{k+1}, w_{E,T}(u^k)) + \frac{1}{T}\mathcal{E}_{E,T}(u^k, w_{E,T}(u^k)) \geq 0.\end{aligned}$$

Thus $\mathcal{L}_E(u^k)$ is increasing. □

Let us notice that one can prove Proposition 3.1 by adapting the proof of Theorem 2 in Wu (1983).

## 3.3 Convergence of the alternating optimization scheme

In this section we state convergence results of the alternating optimization scheme to a critical point of the continuous energy functionals both for the case of patch NL-means and -Poisson models.

We consider first the case of probabilistic weights, with $T > 0$. Let $\mathcal{E}_{E,T}$ be one of the patch NL-means energy $\mathcal{E}_{2,T}$ or patch NL-Poisson $\mathcal{E}_{2,\lambda,T}$. Similarly, $\mathcal{A}_E$ denotes $\mathcal{A}_{2,T}$ or $\mathcal{A}_{2,\nabla,T}$ (the spaces of admissible functions defined in Section 2.2). Let us recall the alternating optimization scheme. This time we make explicit the sets of admissible functions.

**Algorithm 2** Alternate minimization of $\mathcal{E}_{E,T}$.

Initialization: choose $u^0$ with $\|u^0\|_\infty \leq \|\hat{u}\|_\infty$ .

For each $k \in \mathbb{N}$ solve za

$$w^{k+1} = \operatorname*{arg\,min}_{w \in \mathcal{W}} \mathcal{E}_{E,T}(u^k, w), \tag{3.15}$$

$$u^{k+1} = \operatorname*{arg\,min}_{(u,w^{k+1}) \in \mathcal{A}_E} \mathcal{E}_{E,T}(u, w^{k+1}), \tag{3.16}$$

The following propositions hold for patch NL-means, and patch NL-Poisson with $\lambda = 0$. The proofs can be found in Appendix C.

**Proposition 3.2.** *Algorithm 2 converges (modulo a subsequence) to a critical point $(u^*, w^*) \in \mathcal{A}_{2,T}$ of $\mathcal{E}_{2,T}$. The solution obtained has the smoothness described in Proposition 2.1, that is $u^* \in W^{1,\infty}(O)$ and $w^* \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

**Proposition 3.3.** *Algorithm 2 converges (modulo a subsequence) to a critical point $(u^*, w^*) \in \mathcal{A}_{2,0,T}$ of $\mathcal{E}_{2,0,T}$. The solution obtained has the smoothness described in Proposition 2.3, that is $u^* \in W^{1,2}(O) \cap W^{2,p}_{loc}(O) \cap L^\infty(O)$ for any $p \in [1, \infty)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

Let us point out that the convergence of the alternating optimization (Algorithm 2) holds also in the discrete domain.

For the case $T = 0$, we limit our analysis to the discrete case. In the discrete case, the convergence can be proved thanks to the convexity of $\mathcal{E}_{E,0}(u, \nu)$ in each variable when the other is fixed.

**Proposition 3.4.** *In the discrete case, algorithm 2 with $T = 0$ converges (modulo a subsequence) to a critical point of the energy $\mathcal{E}_{E,0}$.*

*Proof.* Since $u^k$ and $\nu^k$ are bounded, there is a subsequence $(u^{k_j}, \nu^{k_j})$ converging to $(\bar{u}, \bar{\nu})$. Notice that if $\nu$ is fixed, then the solution of $\min_u \mathcal{E}_{E,0}(u, \nu)$

is unique. Clearly $\bar{\nu}$ is a minimum of $\nu \to \mathcal{E}_{E,0}(\bar{u}, \nu)$. Proceeding as in Bertsekas (1999), one can prove that $u^{k_j+1} - u^{k_j}$ converges to 0 and deduce that $\bar{u}$ is a minimum of $u \to \mathcal{E}_{E,0}(u, \bar{\nu})$. □

Chapter 4

# Observations on the energies and their minima

In this chapter, the first three sections describe some characteristics of the solutions found by the minimization process, as well as the differences between the schemes derived from the proposed framework. Spetial interest is paid to the regularity of the correspondence maps, discussed in Section 4.2.

Based on the observations made about the behaviour of the different models, in Sections 4.4 and 4.5 we present some useful extensions of the energy.

The analysis will consist sometimes of simple mathematical arguments and sometimes of empirical observations based on synthetic experiments, designed to control and isolate certain specific properties. The aim of this section is to provide insight on the many factors that determine the behaviour of the different methods on more complex, real inpainting problems.

## 4.1 Effect of the selectivity parameter

First we consider the inpainting of a regular texture (shown in Figure 4.1) with two different mean intensities. The inpainting domain hides all patches on the boundary between the dark and bright textures. With this example we can test the ability of each method to *create* an interface between both regions. Situations like these are common in real inpainting problems, for

instance due to inhomogeneous lighting conditions. We have also added Gaussian noise with standard deviation $\sigma = 10$ to show the influence of the selectivity parameter $T$. Each column of Figure 4.1 shows the results of the four methods described in the previous section. We have tested each method with $T = 0$ (top row), and $T > 0$, chosen approximately to match the expected deviation of each patch error due to the noise. For both gradient-based methods we take $\lambda = 0$, *i.e.* no image component.

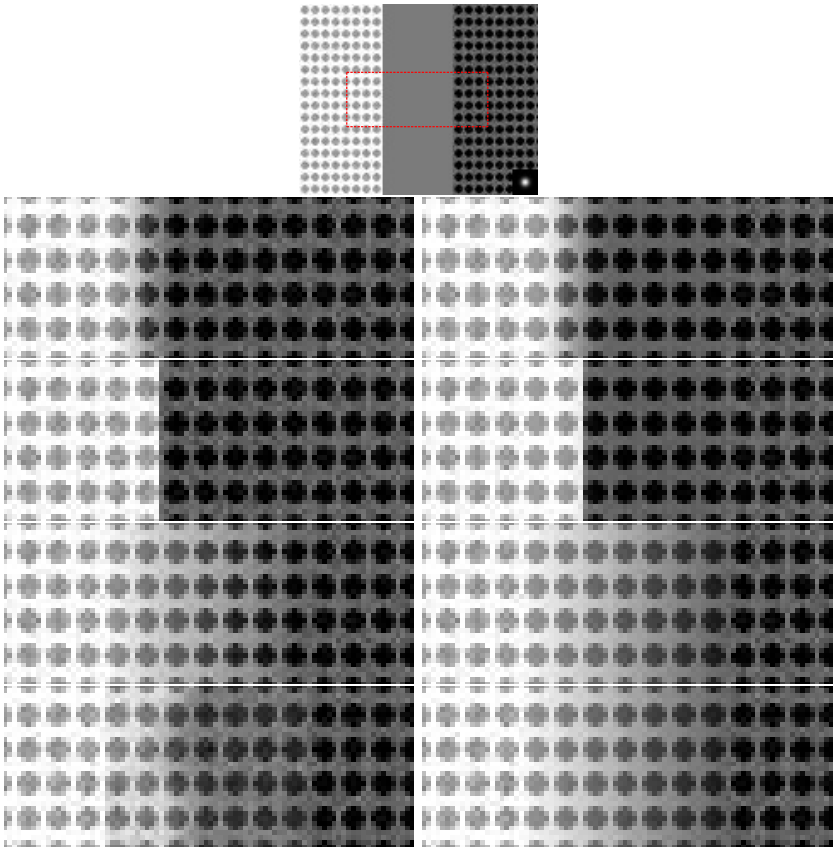


**Figure 4.1: Inpainting of a synthetic texture.** The initial condition is shown in the top row. In the rest of the rows we show a zoom (region in the red rectangle) of the results of patch NL-means, -medians, -Poisson and -GM, from top to bottom. In the frist column $T = 0$, and in the second column $T = 200$, $T = 14$, $T = 400$ and $T = 20$, respectively. The intra-patch weight kernel $g$ is shown in the bottom right corner of the initial condition, it has a standard deviation $a = 5$ and the patch size is $s = 15$.

The first notorious difference is on how the methods handled the transition

between the dark and bright textures. Patch NL-means produces a smooth transition whereas a sharp step is obtained with the patch NL-medians. On the other hand, both gradient-based methods yield a much smoother shading of the texture, due to the fact that the image update step is computed as the solution of a PDE which diffuses the intensity values present at the boundary of the inpainting domain. For the case of patch NL-Poisson this interpolation is linear, since this is a solution of the homogeneous Poisson equation (*i.e.* Laplace equation).

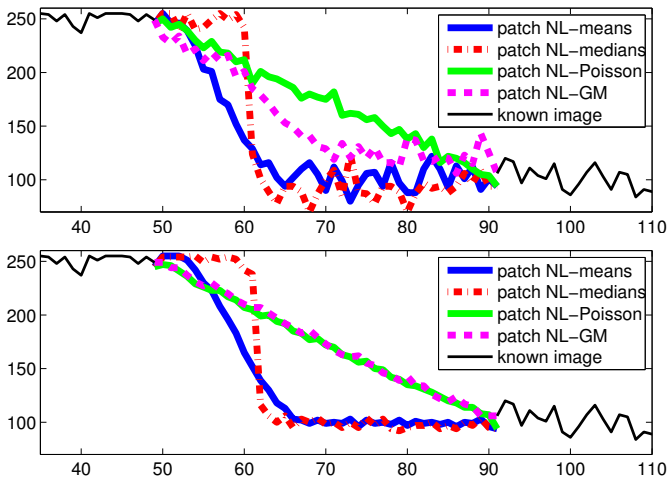



**Figure 4.2: Profiles of the results in Figure 4.1.** The profiles are taken from an horizontal line going between the circles in Figure 4.1. Top: results with $T = 0$ and bottom: results with $T > 0$.

As expected, the results using a higher value of $T$ show some denoising. This effect can be better appreciated in the profiles shown in Figure 4.2, which depict the image values for a horizontal line between the circles. In the usual context of inpainting, in which the available data is not perturbed by noise, this denoising translates into an undesirable loss of texture quality (some details are treated as noise, a common effect in image enhancement). For the rest of the inpainting experiments shown we will consider $T = 0$. In other applications such as denoising or image regularization, the case of $T > 0$ becomes relevant (see Peyré et al. (2011) and our work in Facciolo et al. (2009)).

## 4.2 Regularity of optimal correspondence maps

Let us focus now on the solution of the functional when $T = 0$. In this case, the weights $w$ have to be replaced by a measurable probability- valued map $\nu$, which in turn can be associated with a correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$. For a minimum $(u, \varphi)$, $\varphi(x)$ corresponds to the center of the most similar patch to $p_u(x)$ (the nearest neighbor).

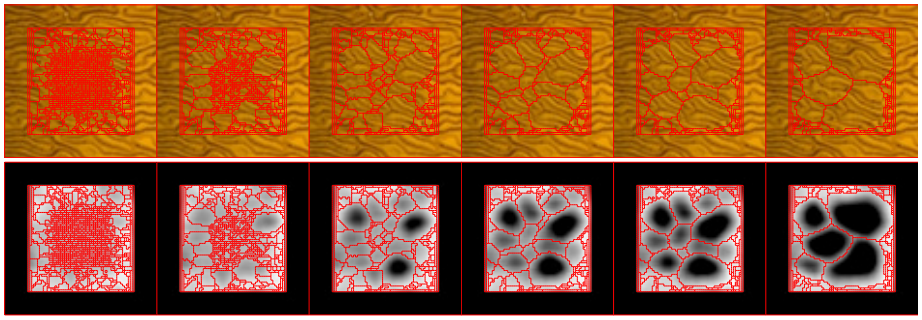


**Figure 4.3: Minimization of the energy.** Some iterations of the minimization procedure corresponding to the patch NL-medians (all schemes have a similar behaviour). The top row shows the first 5 iterations and the result obtained after convergence (rightmost column). The bottom row shows the corresponding distribution of the patch error (energy density). Notice the emergence of coherent copy regions and how the energy concentrates along their boundaries.

### Copy regions and transition bands

In Figure 4.3 we show some iterates of the alternating minimization process for the patch NL-median applied to the texture problem in Figure 4.5 (the other schemes show a similar evolution). The red curves depict the boundaries of the regions with constant offset with respect to the nearest neighbor. This offset is given by $t(x) = \varphi(x) - x$, for $x \in \widetilde{O}$. Following Barnes et al. (2009), we will refer to $t$ as the *nearest-neighbor field* (NNF). The minimization process starts from a highly complex NNF. Then regions of constant offset start to grow from the boundaries towards the interior of the inpainting domain, creating a (rather simple) partition of $\widetilde{O}$.

For that, let us analyze how would the inpainting look like for the simple case in which $O$ is partitioned in two regions of constant NNF, $R_1$ with

$t(x) = t_1$ and $R_2$ with $t_2$ (Figure 4.4). For a patch centered in $R_i$ we have that $\nu_x(\hat{x}) = \delta(\hat{x} - (x + t_i))$. It can be seen that the pixel influence function of a pixel $z \in O$ is given by

$$m(z, \hat{z}) = \alpha_1 \delta(\hat{z} - (z + t_1)) + \alpha_2 \delta(\hat{z} - (z + t_2))$$

for $\hat{z} \in O^c$, where $\alpha_i = \int_{\Omega_p} g(y) \chi_{R_i}(z - y) dy$, with $i = 1, 2$ is the area of the intersection of the patch centered at $z$ with $R_i$. Notice that due to the normalization of $g$, $\alpha_1 + \alpha_2 = 1$. Thus, for the patch NL-means we have

$$u(z) = \alpha_1 \hat{u}(z + t_1) + \alpha_2 \hat{u}(z + t_2),$$

while the patch NL-medians selects the region with larger overlap with the patch at $z$:

$$u(z) = \hat{u}(z + t_{i_M}) \qquad \text{with} \qquad i_M = \arg\max_{i=1,2} \alpha_i.$$

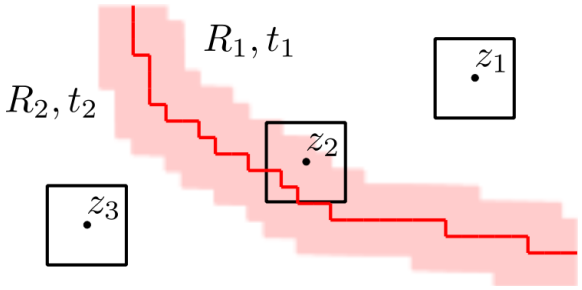



**Figure 4.4: Two copy regions and the transition band.** In regions $R_1$ and $R_2$, which have a constant NNF, data is rigidly translated (copied) from corresponding source regions in the complement. The transition between these copy regions takes place on a band whose width coincides with the patch size.

Pixels in the red band in Figure 4.4 receive two contributions ($\alpha_1, \alpha_2 > 0$). Outside this band, in both regions $R_1$ and $R_2$, the image $u$ results from a rigid translation (*i.e.* a *verbatim* copy) from two corresponding regions in $O^c$. The transition between both *copy regions* takes place at the red *transition band*. Patch NL-means performs a smooth blending whereas patch NL-medians creates a sharp cut. For gradient-based methods, the same reasoning applies at the gradient level.

This argument generalizes to an arbitrary number of regions. The value of $u$ at each pixel $z$ is determined by the copy regions overlapped by the

patch centered at $z$, weighted by the overlap area. The transition bands are defined as the centers of patches intersecting at least two different copy regions. Outside these bands, the resulting image is an exact copy (of intensities or gradients) of a corresponding source region in $O^c$.

The bottom row in Figure 4.3 shows the evolution of the patch error $E[p_u(x) - p_{\tilde{u}}(n(x))]$. Recall from (2.6) that, for $T = 0$, the energy is computed as the sum of these errors. The energy is concentrated around the transition bands, since patches that do not overlap any band are an exact copy of the source patch. This provides some intuition on why the minimization of the energy might favour a piece-wise constant NNF (or a correspondence map which is a piece-wise translation).

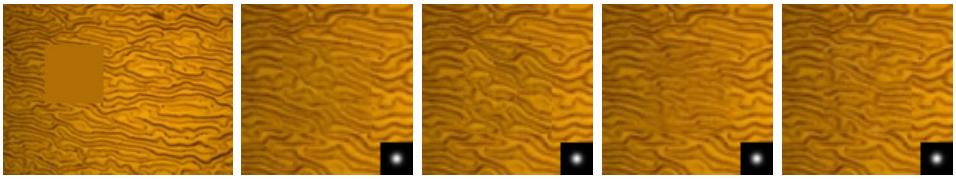


**Figure 4.5: Inpainting of structured texture.** From left to right: Initial condition, result of patch NL-means, -medians, -Poisson and -GM (all results are zoomed-in regions). Results with $s = 15$, $a = 5$ and $T = 0$. The treatment of color images is described in Section 4.5.

The results obtained with the proposed schemes are compared in Figure 4.5. We can see that patch NL-means and -Poisson show some smoothing due to the blending between copy regions. Both $L^1$-based schemes obtained sharper results (for patch NL-medians it is easy to identify the regions in the complement that have been replicated).

Figure 4.6 shows another manifestation of the effects of the transitions. The image to be inpainted consists of Gaussian noise. Using this synthetic texture allows us to measure quantitatively the perceptual quality of the inpainted texture via the local variance (the mean is well reproduced by all methods). We estimate the local variance by smoothing the image of squared differences w.r.t. the mean. In the interior of the copy regions, away from their boundaries (red curves), the reproduction of the source texture is exact, and thus the variance is preserved. As expected the variance decreases along the red curves, where pixel values are synthesized using incoherent contributions from different copy regions. Both for the intensity and the gradient model, the $L^2$-norm causes a higher decay of the variance than the $L^1$-norm.

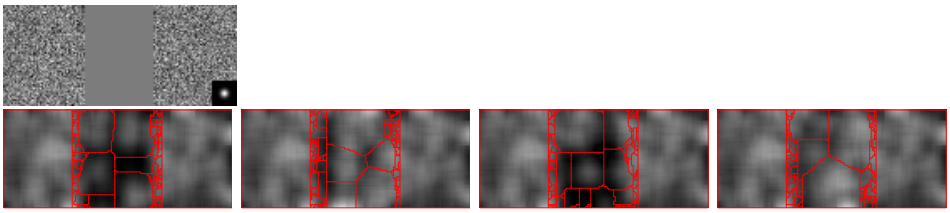


**Figure 4.6: Inpainting of random texture.** Top: initial condition (the gray square is the hole). Bottom, from left to right: images of the local variance superimposed with the boundaries between copy regions (red curves) for patch NL-means, -medians, -Poisson and -GM. Notice the attenuation of the variance over the red curves, specially for $L^2$-based methods.

## Regularity of optimal correspondence maps: mathematical analysis

The empirical observations discussed in the previous section, lead us to study the existence of regular optimal correspondence maps. The following result gives us some information on the the existence of correspondences with some regularity.

**Theorem 4.1.** *Let $X$ be an open bounded subset of $\mathbb{R}^N$ with Lipschitz boundary and $Y$ be a compact subset of $\mathbb{R}^m$. Let $U : X \times Y \to \mathbb{R}$ be a Lipschitz continuous function. For each $x \in X$, let $M(x) := \{y \in Y : U(x,y) = \min_{\bar{y}\in Y} U(x,\bar{y})\}$. Then there exists a selection of the multifunction $x \in X \to M(x) \subseteq Y$, i.e., a function $S : X \to Y$ such that $S(x) \in M(x)\ \forall x \in X$, which is a uniform limit of functions in $BV(X)^m$.*

As a uniform limit of functions in $BV(X)^m$, $S$ inherits some of their properties. In particular, $\mathcal{H}^{N-1}$-a.e. $x \in X$ is either a point of approximate continuity of $S$, or a jump point with two limits.

The proof of Theorem 4.1 can be found in Appendix B, where we also included a brief summary of the properties of $BV$ functions.

The result can be immediately applied to our case with $X = \widetilde{O}$, $Y = \widetilde{O}^c$, by showing that patch error function is Lipschitz continuous. This is shown in the Appendix A in Lemmas A.2, A.6 A.8 and A.9 for patch NL-means, patch NL-medians, patch NL-gradient medians and patch NL-Poisson respectively. These Lemmas hold under the assumptions of the

corresponding existence results in Section 2.2. Theorem 4.1 implies that the offsets map $t(x) = \varphi(x) - x$ is a uniform limit of maps of bounded variation (in this case from $\widetilde{O}$ to $\mathbb{R}^2$).

Notice that the result does not say that all optimal correspondence maps are regular. Take for instance the completion of periodic textures such as the one in Figure 4.1 but without the Gaussian noise. In that case, an exact replication of the periodic pattern can be achieved with its correspondence map jumping continuously from one period to the other. In this case, there are many minima, and some have regular correspondence maps. In view of Propositions 2.1 and 2.3, this raises the question if the solution obtained by annealing, i.e. by solving $\mathcal{E}_{2,T}$ (or $\mathcal{E}_{2,\lambda,T}$) and letting $T \to 0+$ is indeed a regular solution in the sense described in Theorem 4.1. We are not able to answer this question, at present.

This type of regularity is much milder that the one piece-wise translational maps observed for some images. However, it is likely that stronger regularity results require making further assumptions about the image $\hat{u}$. In Section 5.1 we discuss possible directions from derived from a geometric interpretation of the energy in the space of patches.

## 4.3 Local minima

The alternating minimization scheme is only guaranteed to converge to a local minima. Some local mimima do not correspond to a visually plausible inpainting. The different schemes have different behaviours with respect to this issue. We illustrate this with two examples.

### Interpolation of a line

We consider a very simple image consisting on a green vertical line over a dark background. The inpainting domain is a horizontal gap as shown in Table 4.1. The inpainting region is initialized with the background color. For this evaluation we fix the size of the patch and increase the width of the gap. For narrow gaps, all methods are able to join both ends of the green vertical line. When increasing the gap, at a certain width the gray initialization prevails, the methods get stuck on a local minima, not being capable of recovering the vertical line. The first column of Table 4.1 shows the maximum width (in pixels) that produces a good reconstruction. Observe that

the combined schemes propagate the line through wider gaps, and that the optimal mixing parameter $\lambda$ for this purpose is around the values that we have proposed in the previous section. Basically these schemes have two propagation mechanisms: a *local* one, by diffusion of the intensity values by the PDE, and a *non-local* one by transference of gradients from $O^c$. Intuitively, when $\lambda > 0$ these mechanisms reinforce each other: the diffused values allow a better estimation of the weights, and therefore the transference of more appropriate gradients, which will help to diffuse the intensity values further in the next image update step. On the other hand, intensity-based methods depend only on the iteration of weights computation and image update to propagate information.

The second column of Table 4.1 shows the results obtained by incorporating the confidence mask later described in Section 4.5. An alternative way to prolong the geometric structures is to increase the patch size as will be discussed in Section 6.4.

| Method | GAP | GAP ($t_c = 5$) |
|---|---|---|
| P $\lambda = 0$ | 13 | 40 |
| P $\lambda = 0.1$ | 16 | 46 |
| P $\lambda = 0.5$ | 11 | 34 |
| M | 9 | 29 |
| GM $\lambda = 0$ | 15 | $> 56$ |
| GM $\lambda = 0.01$ | 21 | $> 56$ |
| GM $\lambda = 0.001$ | 40 | $> 56$ |
| Md | 7 | 42 |

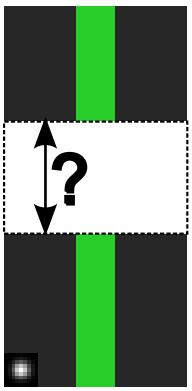



**Table 4.1: Geometric interpolation.** The inpainting domain is shown in white and the patch domain is shown in the lower right corner ($9 \times 9$ pixels). The table reports the maximum gap width (measured in pixels) for which the algorithm is capable of recovering the vertical line. P, M, Md and GM stand for patch NL-Poisson, -means, -medians and -gradient median respectively. The rightmost column shows the maximal gaps obtained with the use of a confidence masks with decay $t_c = 5$ (see Section 4.5).

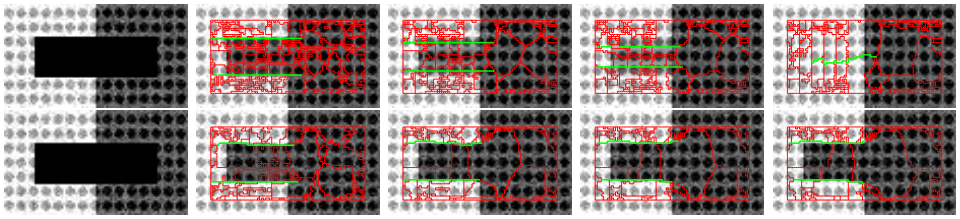


**Figure 4.7: Stiffness effect.** Left: Initial condition. The black rectangle is the inpainting domain. Evolution of patch NL-means (top) and patch NL-medians with same initialization and parameters. The boundaries between copy regions are superimposed on red. Those between bright and dark copy regions have been highlighted.

### Stiffness of the $L^1$-based methods

$L^1$ methods are more reluctant to make changes during the minimization process, and therefore they are more likely to get trapped in local minima. The same robustness that favors sharp transitions between copy regions makes them more greedy. Once a set of neighboring patches have settled on a locally stable solution (typically a region of constant NNF), it is hard for the algorithm to change that local decision. Figure 4.7 shows a comparison between patch NL-means and -medians. Both methods are given a bad initialization. We have highlighted the boundary between bright and dark copy regions (shown in green). A copy front is advancing from the boundary carrying correct information and meets an already settled copy front, which has taken an undesirable decision based on the initialization. Pixels on the boundary of the mistaken front start receiving contributions from patches in the advancing front. Initially these contributions will be outliers in the distribution from which the pixel value is estimated. The median will discard these outliers, and the pixel value will not change unless of course, patches in the advancing front outnumber the mistaken ones.

## 4.4 Decoupling image and weights update steps

In the variational setting, the image synthesis and the weights update are *coupled* through a unique patch error function $E$. One can envision a broader family of methods in which different patch comparison criterions can be chosen independently for the weights update step and for the image

update step. If we denote by $E_u$ and $E_w$ the patch error functions for image update and the weight update respectively, we have the following algorithm

**Algorithm 3** Alternate minimization of $\mathcal{E}_{E_u,T}$ and $\mathcal{E}_{E_w,T}$.

Initialization: choose $u^0$ with $\|u^0\|_\infty \leq \|\hat{u}\|_\infty$ .

For each $k \in \mathbb{N}$ solve

$$w^{k+1} = \operatorname*{arg\,min}_{w \in \mathcal{W}} \mathcal{E}_{E_w,T}(u^k, w), \tag{4.1}$$

$$u^{k+1} = \operatorname*{arg\,min}_{(u,w^{k+1}) \in \mathcal{A}_{E_u}} \mathcal{E}_{E_u,T}(u, w^{k+1}), \tag{4.2}$$

We now discuss two examples. Let us observe that such schemes are non-variational. For the two cases described bellow, it can be proven, that they converge (modulo the extraction of a subsequence) to a Nash equilibrium of two different energy functionals, one for the computation of weights when the image is fixed, and the other for the synthesis of the image, when the weights are fixed (the proof follows that of Proposition 3.3).

In practice, however, we have observed that these schemes may present oscillations.

### Pixel-wise image synthesis

Part of the inspiration for our formalism comes from (1.3), an energy which models the (pixel) non-local means. One could devise an iterative scheme version of the non-local means algorithm, alternating between weights update and pixel-wise image synthesis. To that aim, we define $E_u$ using a patch size of a single pixel

$$E_u(p_u(x) - p_{\hat{u}}(\hat{x}) = (u(x) - \hat{u}(\hat{x}))^2.$$

One can think of $E_u$ as a particular case of the squared $L^2$-norm $\|\cdot\|^2_{g,2}$ when the patch weights $g$ has zero variance.

In this case the image update step yields (compare with (3.8)):

$$u_{k+1}(x) = \int_{\widetilde{O}^c} w_k(x, \hat{x}) \hat{u}(\hat{x}) d\hat{x}. \tag{4.3}$$

Similar iterative schemes have been applied to image denoising (Awate and Whitaker (2006); Brox et al. (2008)). An immediate question is how does such a scheme relates with the variational patch-wise version when applied to the problem of inpainting. One of the first observations to be made, is that both works (Awate and Whitaker (2006); Brox et al. (2008)) perform a slightly different image synthesis, which for the case of inpainting reads

$$u_{k+1}(x) = u_k(x)(1-\tau) + \tau \int_{\widetilde{O}^c} w_k(x,\hat{x})\hat{u}(\hat{x})d\hat{x}.$$

The image update is a convex combination of the current image $u_k$ and the pixel-wise non-local means synthesis, where $\tau \in (0,1]$ is the time step. Note that with $\tau = 1$ the equation reduces to (4.3). In practice $\tau \simeq 0.2$ (Awate and Whitaker (2006); Brox et al. (2008)). This is necessary for the iterative scheme to converge (otherwise it might sometimes present an oscillatory behaviour). However, for such small values of the time step, the convergence will be slow, implying more steps of weight computation and image synthesis. This is particularly undesirable in inpainting applications, where the initialization is typically far away from the solution.

We have not observed such instabilities with the patch-wise synthesis, allowing convergence to be reached in a few iterations. The patch-wise synthesis is computationally more expensive than the pixel-wise one. However, in either case, the cost of the image update is minor compared to that of the weights update. Therefore, the patch-wise synthesis yields usually considerably lower run times.

From the point of view of the quality of the results obtained, let us briefly comment that copy regions also appear with the pixel-wise scheme. As with the patch NL-medians, the transitions between these regions are sharp, improving the rendering of some textures. However methods with a pixel-wise synthesis show a considerably poorer ability to propagate structures. More generally, patch-wise methods propagate the information from the boundary deeper inside the inpaining domain, alleviating the dependence on the initialization. This is a direct consequence of copying patches instead of pixels. We note that these observations are based on results obtained using equation (4.3) (*i.e.* with $\tau = 1$) for which the method did converge.

### Different combinations of gradient and intensities for image and weights updates

Gradient-based methods produce smooth interpolations and enforce the continuity of the image at the boundary of the inpainting domain, which are generally desirable features. For this to be done variationally the similarity weights $w$ have to be computed using patches of the gradient, which in most cases do not provide a reliable measure of patch similarity. For this reason, both for patch NL-Poisson and patch NL-gradient medians, we define the patch error function as a convex combination between a gradient patch error function and the squared $L^2$-norm of intensities (the patch error function of patch NL-means). In this way, by controlling the coefficient $\lambda$ of the convex combination we can take the image values into account for the computation of the weights, and at the same time, to synthesize the image with a diffusion PDE.

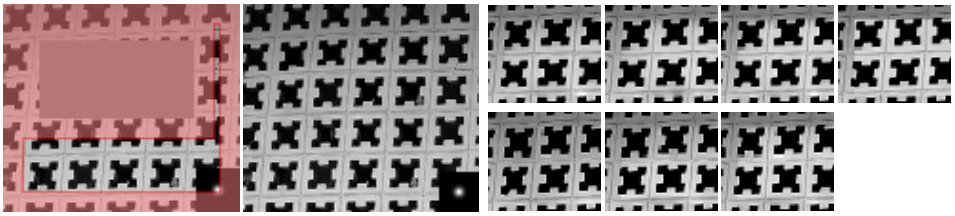


**Figure 4.8: Linear combination of gradient-based methods with NL-means.** First image starting from the left: Initialization. The gray rectangle is the inpainting domain. Only patches centered outside the red area are available. Second image: Result obtained with patch NL-GM, using $\lambda = 0$. A similar result is obtained with the patch NL-Poisson with $\lambda = 0$ (not shown). Details: Top, from left to right: results with patch NL-Poisson with $\lambda = 0.01, 0.05, 0.1, 1$ (the latter corresponds to patch NL-means). Bottom: results with patch NL-GM with $\lambda = 0.001, 0.005, 0.01$. All details have been linearly stretched for display. Differences are most noticeable in the top and in the left of the images.

In Figure 4.8 we show some results corresponding to the combination of both gradient schemes with patch NL-means while varying the mixture coefficient $\lambda$. The image shows a periodic pattern with an illuminance gradient. Most of the dark exemplars are incomplete, and thus only bright exemplars from the bottom of the image are available. The rightmost detail shows the result of patch NL-means: the image has been completed using bright patches and presents a discontinuity on the upper side of the hole. On the other hand, a completion using gradients only (see result of the patch NL-GM) manages

to interpolate both the texture and the shading. The small images on the right show results of both gradient methods with different values for $\lambda$. The value of the mixing parameter $\lambda$ should be carefully selected since it mixes two different magnitudes (norms of gradients and gray levels). With $\lambda \sim 0.1$ for patch NL-Poisson and $\lambda \sim 0.01$ for patch NL-GM, some of the good continuation properties are preserved and enough color information is added to the patch comparison criterion.

However, as will be shown in Chapter 7, in many cases, one cannot find a value of $\lambda$ which balances a gradient-based blending of the image and at the same time produces reliable similarity weights. For such cases it is usefull to extend gradient-based methods by considering two mixture coefficients: $\lambda_w \in [0,1]$ for the weights update step and $\lambda_u \in [0,1]$ controlling the image synthesis (by minimizing $\mathcal{U}_{\lambda_u,p}$ with respect to $u$). This allows to combine the benefits of intensity- and gradient-based methods into a single algorithm.

## 4.5 Extensions

### Color images

An energy for color images can be obtained by defining a patch error function for color patches as the sum of the error functions of the three scalar components:

$$E(p_{\mathbf{u}}(x) - p_{\hat{\mathbf{u}}}(\hat{x})) = \sum_{i=1}^{3} E(p_{u_i}(x) - p_{\hat{u}_i}(\hat{x})),$$

where $\mathbf{u} : \Omega \to \mathbb{R}^3$ is the color image, and $u_i$, with $i = 1, 2, 3$, its components (analogously for gradient-based errors). Given the weights, each channel is updated using the corresponding scheme for scalar images. All channels are updated using the same weights.

### Confidence mask

For large inpainting domains, it is useful to introduce a mask $\kappa : \Omega \to (0, 1]$ which assigns a confidence value to each pixel, depending on the certainty of its information (see also Criminisi et al. (2004); Komodakis and Tziritas

(2007)). This will help in guiding the flow of information from the boundary towards the interior of the hole, eliminating some local minima and reducing the effect of the initial condition. The resulting image energy term takes the form

$$\mathcal{U}_E(u) = \int_{\widetilde{O}} \int_{\widetilde{O}^c} \kappa(x) w(x, \hat{x}) E(p_u(x) - p_{\hat{u}}(\hat{x})) d\hat{x} dx.$$

The effect of $\kappa$ on the image update step can be seen on the pixel influence weights $m$

$$m(z, \hat{z}) = \int_{\Omega_p} \chi_{\widetilde{O}^c}(\hat{z} - y) g(y) \kappa(z - y) w(z - y, \hat{z} - y) \mathrm{d}y.$$

Thus, the contribution of the patch $p_u(z - y)$ to the evidence function is now weighted by its confidence. Patches with higher confidence will have a stronger influence. With the inclusion of the confidence mask, the normalization coefficient $c(z)$ becomes:

$$c(z) = \int_{O^c} m(z, \hat{z}) d\hat{z} = \int_{\Omega_p} g(y) \kappa(z - y) dy.$$

On the similarity weights, the confidence mask has the effect of modifying the selectivity parameter $T$ by a locally varying $T/\kappa(x)$. If the confidence is high, the effective selectivity $T/\kappa(x)$ will be lower, thus increasing the selectivity of the similarity measure. When $T = 0$ the weights are substituted by a Dirac's delta independently of $\kappa$.

For the experiments shown in this paper, the confidence mask was set to

$$\kappa(x) = \begin{cases} (1 - \kappa_0) \exp\left(-\frac{d(x, \partial O)}{t_c}\right) + \kappa_0 & \text{if } x \in O, \\ 1 & \text{if } x \in O^c, \end{cases}$$

which shows an exponential decay w.r.t. the distance to the boundary inside the hole $d(\cdot, \partial O)$. Here $t_c > 0$ is the decay time and $\kappa_0 > 0$ determines the asymptotic value reached far away from the boundary. Setting $t_c = 0$ amounts to using a constant confidence mask. Table 4.1 shows the effect of using a confidence mask with $t_c = 5$ and $\kappa_0 = 0.1$, allowing the restoration of the vertical line for much wider gaps, and thus alleviating the dependence on the gray initial condition.

## Interpolation of sparsely sampled images

The framework we presented has been generalized to the interpolation of sparsely sampled images in Facciolo et al. (2009). This problem differs from the typical inpainting scenario in that there are no complete patches in $\widetilde{O}^c$ to copy from. Other exemplar-based methods used for interpolation of sparse samples, generally based on *sparseland* model (Aharon et al. (2006); Candes and Wakin (2008)), resort to a dictionary with complete patches built in a previous training stage (Aharon et al. (2006); Elad et al. (2005); Mairal et al. (2008)). The extension of our framework proposed in Facciolo et al. (2009) allows to use the incomplete patches from the sparsely sampled image as the only source of information. A description of this method is out of the scope of this thesis. The interested reader can refer to Facciolo et al. (2009).

Chapter 5

# Discussion and connections

In this chapter we discuss connections between the proposed framework and related models. We start by providing a geometric description of the energy in the space of patches in Section 5.1. In Sections 5.2 and 5.3 we review similar energies in statistical mechanics, and their application to related geometric problems like point clustering and rigid registration between two sets of points. In Section 5.4 we revisit related work in image inpainting and non-local image regularization. Finally, in 4.5 we describe some extensions of the proposed framework.

## 5.1 Interpretation in the space of patches

The proposed functionals for $T > 0$ and $T = 0$, have a simple geometric interpretation in the space of patches. We will exploit this geometric viewpoint to review the problem of the regularity of the correspondence map, and to draw connections to similar approaches for related geometrical problems.

Similar geometric interpretations for related non-local regularization models have been proposed by Peyré (2009) and Peyré et al. (2011). We will discuss them later in Section 5.4.

## Manifolds of patches

We consider the problem from the point of view of the manifold of patches. For a continuous image, the space of patches is a functional space. Thus, for simplicity, we consider the case of continuous images with sampled patches. We also assume that $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is smooth (suppose $u$ is acquired by an optical system with a smoothing point spread function $K$). The patch of $u$ centered at $x$ is given by a sampling of $u$ on a neighborhood of $x$. Let $\Pi = \Omega_p \cap \mathbb{Z}^2$, which is a discrete neighborhood of $0 \in \mathbb{Z}^2$. Denoting by $D$ be the cardinal of $\Pi$, we have that $p_u(x) = (u(x+h) : h \in \Pi) \in \mathbb{R}^D$. Given a image $u$, the map $p_u : \Omega \to \mathbb{R}^D$ defines a parameterization of a manifold of $\mathcal{M}_u$, given by the image of $p_u$.

## $T > 0$: density model in the space of patches

As shown in Proposition 3.1 the alternating minimization scheme is equivalent to an EM algorithm applied to the maximiztion of the marginal likelyhood:

$$\mathcal{L}_{E,T}(u) = \int_{\widetilde{O}} \log\, Z_{E,T}(u; x)\, dx \tag{5.1}$$

where

$$Z_{E,T}(u) = \int_{\widetilde{O}^c} \exp\left(-\frac{1}{T} E(p_u(x) - p_{\hat{u}}(\hat{x}))\right) d\hat{x}.$$

We can interpret the above expressions in terms of a probabilistic model in the space of patches, by noticing that $Z_{E,T}(u; x)$ is a density estimate (in the patch space) of the set of patches in $\widetilde{O}^c$: it corresponds to the total unnormalized similarity of patch $p_u(x)$.

Thus, the image is considered as an ensemble of overlapping patches. Known patches in $\widetilde{O}^c$ are fixed, forming a patch density model used to estimate the patches in $\widetilde{O}$. The minimizers $(u^*, w^*)$ are obtained when the patches of the inpainted image are in regions of high density of known patches in the patch space.

### $T = 0$: manifold model in the space of patches

When $T = 0$, minimizing the energy $\mathcal{E}_{E,0}$ on $(u, \nu)$ corresponds to minimizing the following energy, depending only on the image $u$

$$\mathcal{L}_{E,0}(u) = \int_{\widetilde{O}} \inf_{\hat{x} \in \widetilde{O}^c} \{E(p_u(x) - p_{\hat{u}}(\hat{x}))\} dx \tag{5.2}$$

For a closed set $C \subset \mathbb{R}^D$, we denote by $d_C$ the distance of a point $p \in \mathbb{R}^D$ to $C$, measured by $E$, that is $d_{C,E}(p) = \inf_{q \in C} E(p - q)$. Thus, instead of a density model, the energy penalizes $d_{\mathcal{M}_{\hat{u}}}$ the distance to the manifold of known patches.

#### Remarks on the existence of regular correspondence maps: the point of view of the manifold of patches

Let us explain the role of the selection Theorem 4.1, considering the problem from the point of view of the manifold of patches. As before, we consider a continuous image domain but we sample the patches. $\mathcal{M}_u \subset \mathbb{R}^D$ is the manifold of patches given by the image of $p_u$. By suitable assumptions on the kernel $K$ we may assume that $p_u$ is of class $C^1$.

For a fixed $u$, the computation of the correspondence map can be seen as the following problem. Given two images $u : \Omega_u \to \mathbb{R}$, $\bar{u} : \Omega_{\bar{u}} \to \mathbb{R}$ with the corresponding manifolds of patches $\mathcal{M}_u, \mathcal{M}_{\bar{u}}$, let us consider the map $\mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p)$ that associates to each point $p \in \mathbb{R}^D$ the set of points of $\mathcal{M}_{\bar{u}}$ of minimal distance to $p$, that is

$$\mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p) := \{q \in \mathcal{M}_{\bar{u}} : d_{\mathcal{M}_{\bar{u}}}(p) = \|p - q\|\}.$$

What is the regularity of correspondence maps $\varphi$ that associate to each point $x \in \Omega_u$ a point $\varphi(x) \in \Omega_{\bar{u}}$ such that $p_{\bar{u}}(\varphi(x)) \in \mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p_u(x))$ ?

The regularity of the correspondence maps is related to the regularity of the parameterizations $p_u$ and $p_{\bar{u}}$ of the manifolds $\mathcal{M}_u$ and $\mathcal{M}_{\bar{u}}$, and to the regularity of the projection map $\mathrm{proj}_{\mathcal{M}_{\bar{u}}}(x)$. Our purpose is to describe what is the generic geometric situation and justify the need of using selection theorems in order to ensure the existence of mildly regular correspondences. For that, let us recall some results on the differentiability and the singular set of semiconcave functions Cannarsa and Sinestrari (2004).

A function $f$ defined on an open set $A \subset \mathbb{R}^D$ is semiconcave with modulus $\omega$ if there is a nondecreasing upper semicontinuous function $\omega : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\lim_{\rho \to 0+} \omega(\rho) = 0$ and

$$\lambda f(p) + (1-\lambda) f(q) - f(\lambda p + (1-\lambda) q) \leq \lambda(1-\lambda)|p-q|\omega(|p-q|)$$

for any pair $p, q$ such that the segment $[p, q]$ is contained in $A$ and for any $\lambda \in [0, 1]$. If $w(\rho) = \rho$ we say that the modulus is linear.

The distance $d_C$ to a closed set $C \subset \mathbb{R}^D$ is a locally semiconcave function in $\mathbb{R}^N \setminus C$ with linear modulus and $d_C^2$ is semiconcave in $\mathbb{R}^D$ with linear modulus.

**Proposition 5.1.** *(Cannarsa and Sinestrari (2004), Chapter 3) If $f$ is semiconcave, then the superdifferential $D^+ f(p) \neq \emptyset$. If $D^+ f(p)$ is a singleton, then $f$ is differentiable at $p$. If $D^+ f(p)$ is a singleton for any $p \in A$, then $f \in C^1(A)$.*

**Proposition 5.2.** *(Cannarsa and Sinestrari (2004), Proposition 4.1.2) Suppose that $f : A \to \mathbb{R}$ is semiconcave with a linear modulus. Then the gradient of $f$ is a function in $BV_{loc}(A, \mathbb{R}^D)$ and the set $\Sigma(f)$ of singularities of $f$,* i.e. *the points where $f$ is not differentiable, coincides with the jump set $S_{Df}$ of the gradient of $f$, which is a $\mathcal{H}^{D-1}$-rectifiable set.*

**Proposition 5.3.** *(Cannarsa and Sinestrari (2004), Proposition 4.4.1) Let $C \subset \mathbb{R}^D$ be a closet set. Then $d_C$ is semiconcave with linear modulus in $\mathbb{R}^D \setminus C$. $d_C$ is differentiable at $p \notin C$ if and only if* $\mathrm{proj}_C(p)$ *is a singleton. In that case*

$$\mathrm{proj}_C(p) = p - d_C(p) D d_C(p).$$

Let us apply these results to the distance map $d_{\mathcal{M}_{\bar{u}}}$. If $p \notin \Sigma(d_{\mathcal{M}_{\bar{u}}})$, then $d_{\mathcal{M}_{\bar{u}}}$ is differentiable at $p$ and $\mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p)$ is a single point on $\mathcal{M}_{\bar{u}}$. Thus, the function $d_{\mathcal{M}_{\bar{u}}}$ is differentiable at the points $\mathcal{M}_u \setminus \Sigma(d_{\mathcal{M}_{\bar{u}}})$. Hence the mapping $\mathrm{proj}_{\mathcal{M}_{\bar{u}}}$ is well defined on $\mathcal{M}_u \setminus \Sigma(d_{\mathcal{M}_{\bar{u}}})$ and is given by

$$p \in \mathcal{M}_u \setminus \Sigma(d_{\mathcal{M}_{\bar{u}}}) \to \mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p) = p - d_{\mathcal{M}_{\bar{u}}}(p) D d_{\mathcal{M}_{\bar{u}}}(p).$$

Hence it is a $BV$ function (see Proposition 5.2). Since the map $x \in \Omega_u \to p_u(x)$ is differentiable, the mapping $x \in \{\bar{x} \in \Omega_u : p_u(\bar{x}) \notin \Sigma(d_{\mathcal{M}_{\bar{u}}})\}$ to $\mathrm{proj}_{\mathcal{M}_{\bar{u}}}(p_u(x))$ is a BV function. By Proposition 5.1 we may assume that it is $C^1$ at the points in $\mathcal{M}_u \setminus \Sigma(d_{\mathcal{M}_{\bar{u}}})$.

Generically, we may assume that $\mathcal{M}_u$ intersects $\Sigma(d_{\mathcal{M}_{\bar{u}}})$ transversally, thus we may assume that the intersection $\mathcal{M}_u \cap \Sigma(d_{\mathcal{M}_{\bar{u}}})$ is a rectifiable curve in $\mathcal{M}_u$.

Let $\mathcal{MP}_2(\bar{u})$ be the set of points $y \in \Omega_{\bar{u}}$ where the rank of the differential of $p_{\bar{u}}$ at $y$ is 2. Since the differential of $p_{\bar{u}}$ is given by the map $dp_{\bar{u}}(y)(v) = \{v \cdot \nabla \bar{u}(y+h) : h \in \Pi\}$ and

$$\|dp_{\bar{u}}(y)(v)\|_2^2 = \sum_{h \in \Pi} \langle \nabla \bar{u}(y+h) \otimes \nabla \bar{u}(y+h)(v), v \rangle,$$

the rank of $dp_{\bar{u}}(y)$ is the rank of the $2 \times 2$ matrix $\sum_{h \in \Pi} \nabla \bar{u}(y+h) \otimes \nabla \bar{u}(y+h)$.

Since we can give the structure of a differentiable manifold to $p_{\bar{u}}(\mathcal{MP}_2(\bar{u}))$, if we assume that $\Omega_{\bar{u}} = \mathcal{MP}_2(\bar{u})$ and $p_{\bar{u}}$ is injective, then $x \in \Omega_u \to \varphi(x) \in \Omega_{\bar{u}}$ is a (locally) $BV$ function.

Thus, we have seen that there are several obstructions to the regularity of $\varphi$. The first is the position of the manifold $\mathcal{M}_u$ relative to the singularities of the distance map to $\mathcal{M}_{\bar{u}}$. Intuitively, if they intersect, generically this intersection is a rectifiable curve. At those points, we have to select the correspondence map so that it is a $BV$ map. More significantly, the map $p_{\bar{u}}$ may not be injective and then we need again a selection. We can only guarantee the regularity of $\varphi$ if $\text{proj}_{\mathcal{M}_{\bar{u}}}(p_u(x))$ is a point in $p_{\bar{u}}(\mathcal{MP}_2(\bar{u})$ (that is, in the image of the set of points where $p_{\bar{u}}$ is injective) where the inverse of $p_{\bar{u}}$ is differentiable. Thus, regularity without using the selection theorem can only be guaranteed at the points of $\mathcal{MP}_2(\bar{u})$ where $p_{\bar{u}}$ is injective. But we only know that $p_{\bar{u}}$ is locally injective at the points of $\mathcal{MP}_2(\bar{u})$ and we can only expect a $BV$ regularity result on $\varphi$ in all $\Omega_u$ by a proper selection of the inverse map of $p_{\bar{u}}$. For that reason, we used the extension to the $BV$ case of the Kuratowski-Ryll-Nardewsky selection theorem in order to have a global statement in all $\Omega_u$.

## 5.2 Connections with statistical mechanics

For simplicity, we restrict ourselves to the discrete case where $O$ is a subset of a domain $\Omega$ in $\mathbb{Z}^2$. In the context of statistical mechanics (Mézard and Montanari (2009)) we consider $\widetilde{O}$ as a system of *particles* $x \in \widetilde{O}$. Each particle $x$ can be in one of a set of possible *states* $\hat{x} \in \widetilde{O}^c$. The *configuration* of the system is determined by the state of all its particles. Thus, we can encode the configuration with the correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$, which

assigns a state to each particle. The set of all possible configurations is then $\Phi = \{\varphi : \widetilde{O} \to \widetilde{O}^c\}$, which has a total of $|\widetilde{O}^c|^{|\widetilde{O}|}$ elements.

A given image completion $u$, which we consider fixed, defines an energy for each configuration $\varphi$

$$U_E(u, \varphi) = \sum_{x \in \widetilde{O}} E(p_u(x) - p_{\hat{u}}(\varphi(x)) = \sum_{x \in \widetilde{O}} E_x(u, \varphi(x)).$$

To simplify notation, we have defined the energies $E_x(u, \varphi(x)) = E(p_u(x) - p_{\hat{u}}(\varphi(x))$ for $x \in \widetilde{O}$. The notation makes explicit the dependence on $u$.

Let us assume a given probability distribution $P(\varphi)$ defined over $\Phi$, the configuration space. The *Gibbs free energy* with temperature $T$ is then defined as follows

$$\mathcal{G}_{E,T}(u, P) = \sum_{\varphi \in \Phi} P(\varphi) U_E(u, \varphi) + T \sum_{\varphi \in \Phi} P(\varphi) \log P(\varphi).$$

The Gibbs free energy is considered as a function of the probability distribution on the space of configurations $P$. The Gibbs free energy is relevant in statistical physics since it provides a criterion for the equilibrium distribution for a system at constant temperature: A system at constant temperature is in equilibrium if the Gibbs energy is at its minimum. The distribution that attains the minimum is the Boltzmann distribution:

$$P_{E,T}(u; \varphi) = \frac{1}{Z_{E,T}(u)} \exp\left(-\frac{1}{T} U_E(u, \varphi)\right), \tag{5.3}$$

where

$$Z_{E,T}(u) = \sum_{\varphi \in \Phi} \exp\left(-\frac{1}{T} U_E(u, \varphi)\right)$$

is a normalization factor called the partition function.

This can be easily seen. From the previous equation, we have that

$$U_E(u, \varphi) = -T \log Z_{E,T}(u) - T \log P_{E,T}(u, \varphi).$$

Substituting this expression into the Gibbs free energy yields

$$\begin{aligned} \mathcal{G}_{E,T}(u, P) &= -T \log Z_{E,T}(u) + T \sum_{\varphi \in \Phi} P(\varphi) \log\left(\frac{P(\varphi)}{P_{E,T}(u; \varphi)}\right) \\ &= \mathcal{F}_{E,T}(u) + T \, \mathrm{KL}(P, P_{E,T}(u)), \end{aligned}$$

where we have introduced the *Helmholtz free energy*, given by $\mathcal{F}_{E,T}(u) = -T \log Z_{E,T}(u)$, and identified the second term as $T$ times the Kullback-Leibler divergence between probabilities $P$ and $P_{E,T}(u)$. The Helmholtz free energy does not depend on $P$, and the Kullback-Leibler divergence is zero when $P = P_{E,T}(u)$ and is positive otherwise. Thus, it can be easily seen that the Boltzmann distribution minimizes the Gibbs free energy, and the minimum is the Helmholtz free energy.

The Gibbs free energy resembles our inpainting energy $\mathcal{E}_{E,T}$: The first term is the expected value of the energy under $P$, and the second is the negative entropy of $P$, multiplied by $T$. However, the sums in the definition of $\mathcal{G}$ and $Z_T$ are on the space of all possible configurations $\Phi$; this amounts to a total of $|\widetilde{O}^c|^{|\widetilde{O}|}$ terms, as opposed to the $\widetilde{O} \times \widetilde{O}^c$ terms in $\mathcal{E}_{E,T}$.

As we shall see next, the inpainting energy $\mathcal{E}_{E,T}$ corresponds to a restriction of the Gibbs free energy to a particular class of probability distributions $P$.

This comes as a consequence that $U_E(u, \varphi)$ has a very simple structure: it is the sum of independent energies for each particle $x \in \widetilde{O}$. There are no terms coupling different particles. This is what is called a non-interacting system.

This simplifies the expressions above. For instance for the partition function we have that

$$Z_{E,T}(u) = \prod_{x \in \widetilde{O}} \sum_{\varphi(x) \in \widetilde{O}^c} \exp\left(-\frac{1}{T} E_x(u, \varphi(x))\right) = \prod_{x \in \widetilde{O}} Z_{E,T}(u, x),$$

*i.e.* the global partition function $Z_T$ factorizes as a product of individual normalization factors for each $x \in \widetilde{O}$. Substituting in the definition of the Boltzmann distribution, we have that

$$\begin{aligned} P_{E,T}(u, \varphi) &= \prod_{x \in \widetilde{O}} Z_{E,T}(u, x)^{-1} \exp\left(-\frac{1}{T} \sum_{x \in \widetilde{O}} E_x(u, \varphi(x))\right) \\ &= \prod_{x \in \widetilde{O}} Z_{E,T}(u, x)^{-1} \exp\left(-\frac{1}{T} E_x(u, \varphi(x))\right) = \prod_{x \in \widetilde{O}} w_{E,T}(u; x, \varphi(x)). \end{aligned}$$

Due to the absence of interacting terms in the energy, the equilibrium probability factorizes as the product of the probabilities of each particle $x$ being at state $\varphi(x)$, *i.e.* the states of different particles are statistically independent. Each of these factors is given by the optimal similarity weights $w_{E,T}(u)$.

Let us now see which is the non-interacting expression for the Gibbs free energy. The Gibbs free energy is defined for any probability distribution on the configuration space, including those in which states of different particles are not independent. Let us ignore those cases, and consider a restriction of the Gibbs free energy to those distributions that factorize over the particles:

$$P(\varphi) = \prod_{x \in \widetilde{O}} w(x, \varphi(x)).$$

In that case, it can be shown that the Gibbs free energy reduces to

$$\begin{aligned} \mathcal{G}_{E,T}(u, w) &= \sum_{x \in \widetilde{O}} \sum_{\varphi(x) \in \widetilde{O}^c} w(x, \varphi(x)) E_x(u, \varphi(x)) \\ &\qquad + T \sum_{x \in \widetilde{O}} \sum_{\varphi(x) \in \widetilde{O}^c} w(x, \varphi(x)) \log(w(x, \varphi(x)) \\ &= \mathcal{U}_E(u, w) - T\mathcal{H}(w) = \mathcal{E}_{E,T}(u, w) \end{aligned}$$

When restricting the Gibbs free energy to non-interacting probability distributions, we obtain our energy $\mathcal{E}_{E,T}$. The minimizer of this restricted Gibbs free energy will still be the Boltzmann distribution $P_{E,T}(u)$, since it factorizes over the particles.

Furthermore, the Helmholtz free energy results in

$$\mathcal{F}_{E,T}(u) = -T \log \prod_{x \in \widetilde{O}} Z_{E,T}(u; x) = -T \sum_{x \in \widetilde{O}} \log Z_{E,T}(u; x) = \mathcal{L}_{E,T}(u),$$

which coincides with the marginal log likelihood defined in Section 3.2.

## 5.3 Deterministic annealing for related geometric problems

In the previous sections we interpreted our model from a geometric point of view in the space of patches, and we showed that the inpainting energy has the form of the Gibbs free energy of a system of particles in the context of statistical mechanics. Interestingly, statistical mechanics has inspired similar approaches for the solution of related geometric problems for sets of points in $\mathbb{R}^D$, such as clustering (see Rose (1998) and references therein) and rigid registration of two point sets (Rangarajan et al. (1997)), among

others. In this section we review the similarities and the differences between these problems and ours.

These methods have in common that they look for a correspondence map between points by minimizing a non-convex energy. To avoid local minima they relax the energy by using a probabilistic correspondence, and gradually reduce the randomness of the assignment until the get a deterministic assignment. The rationale behind this approach, is that the problem for a random assignment is easier to solve (it may be convex). This scheme, called deterministic annealing, is inspired by statistical mechanics, thus we will first mention the basic statistical mechanics framework.

In chemistry an annealing procedure is often used to drive a system to configuration of minimal energy. It consists in embedding the system in a thermal bath at a high temperature, and slowly lowering the temperature, maintaining at all time the system at thermal equilibrium.

### Deterministic annealing for clustering

Let us now review the statistical mechanics formulation of clustering of Rose (1998). We use his notation. The purpose is to partition the set of data points into subgroups each of which as homogeneous as possible. For that one defines a cost criterion to be minimized which may depend on the application. To fix ideas, let us mention the case of quantization. In that case given a source vector we want to represent it by a codevector $y(x)$ from a codebook so that the expected distortion is minimized:

$$\mathcal{D} = \sum_x p(x) d(x, y(x)).$$

Here $p(x)$ is the distribution of the source and $d(x, y)$ represents a distortion measure. The mapping of the source vectors $x$ to a codebook vector $y$ determine a partition of the source.

A probabilistic framework is defined by randomizing the encoding rule (fuzzy clustering). Then the distortion measure can be written as

$$\mathcal{D} = \sum_x \sum_y p(x, y) d(x, y(x)) = \sum_x p(x) \sum_y p(y|x) d(x, y),$$

where $p(x, y)$ is now the joint probability distribution and $p(y|x)$ is the association probability relating input vector $x$ to the codevector $y$. If we

minimize $\mathcal{D}$ with respect to $(y, p(y|x))$ we obtain a hard assignment $x \to y(x)$. But one can ask for the minimization of $\mathcal{D}$ subject to a level of randomness, measured by the entropy

$$\mathcal{H} = \sum_x \sum_y p(x,y) \log\, p(x,y).$$

The solution can be obtained using Lagrange multipliers and minimizing the Lagrangian $\mathcal{G} = \mathcal{D} - T\mathcal{H}$, where $T$ is the Lagrange multiplier. Note that $\mathcal{G}$ has the form of a Gibbs free energy, when seen as a function of $p$.

The objective of minimal distortion is reached through deterministic annealing: $T$ is gradually reduced, and with it the level of distortion, until a hard assignment with minimal distortion is reached. The behavior of the association probabilities with respect to $T$ goes from a uniform distribution for large $T$ (input vectors are associated to the codebook with uniform distribution) to a hard assignment as $T \to 0+$. During this cooling scheme, the system undergoes a sequence of phase transitions in which the clusters split and the number of clusters increases. In this context the critical temperatures can be computed In some sense, when letting $T \to 0+$, the system self-organizes and looks for a better optimum. As in our case, the optimum codevectors and association probabilities for fixed $T$ are obtained using an alternating optimization scheme which alternates minimizations with respect to the codevectors and the association probabilities.

To understand the relation between our inpainting problem and quantization, we consider an image over a discrete domain. The objective is to find a correspondence which assings each unknown patch $p_u$ a known patch $p_{\hat{u}}$, such that the patch error $E$ is minimized. Thus the $x$ vectors correspond to the unknown patches, $p_u$, the codevectors correspond to the known patches $p_{\hat{u}}$ and the patch error function $E$ corresponds to the distorion. Then, the energy $\mathcal{G}$ corresponds to the inpainting energy with $T > 0$, $\mathcal{E}_{E,T}$. The difference between our approach and the quantization problem, is that for us the variables are the $x$ vectors, in the domain of the correspondence, whereas in the context of quantization, the variables are the $y$ vectors, in the codomain of the correspondence.

These are not interchangeable. When minimizing over $y$ we aim at covering the hole set of $x$ vectors, so that each $x$ vector has a nearby codevector. When minimizing over $x$, we do not need to cover the set of $y$ vectors. We only require that each vector $x$ is close to a codevector $y$. Indeed, when minimizing over $x$, the problem can be solved independently for each $x$. In

this setting, an annealing procedure yields a constant solution $x = \bar{y}$ for all $x$ vectors, where the vector $\bar{y}$ is the codevector closer to the average codevector. In the inpainting application, the $x$ vectors correspond to patches of an image and they are not independent since nearby patches overlap. Furthermore, there are constraints given by the boundary conditions of the inpainting domain. Thus, setting all patches as a constant might not be a solution. Still, we have observed that a deterministic annealing scheme has a bias towards this type of solutions.

### Deterministic annealing for point set registration

Deterministic annealing has also been used effectively to find a rigid transformation registering two sets of points Rangarajan et al. (1997). As before, let us denote by $x$ and $y$ the points in each set. Let us assume that both sets have the same number of points. The transformation is applied to the $y$ points, and should map each $y$ to match a single $x$ point. There is thus a bijective correspondence between both sets, we denote by $y(x)$, such that $\tau(y(x)) \approx x$. The aim is then to find a bijective correspondence $x \mapsto y(x)$ and the transformation $\tau$ (parametrized by a rotation angle, a scaling factor and a translation vector) such that the following matching error is minimized

$$\mathcal{D}(\tau, y(\cdot)) = \sum_x d(x, \tau(y(x))),$$

where $d(\cdot, \cdot)$ is a distance. The correspondence between both sets can be encoded with binary weights $w(x, y) \in \{0, 1\}$, resulting in the equivalent function

$$\mathcal{D}(\tau, w) = \sum_x \sum_y w(x, y) d(x, \tau(y)),$$

with the constraints $w(x, y) \in \{0, 1\}$, $\sum_x w(x, y) = 1$ and $\sum_y w(x, y) = 1$. These constraints on $w$ imply that the correspondence is bijective, and also that $w(x, y)$ is a permutation matrix.

The energy $\mathcal{D}$ is minimized with an alternating optimization scheme similar to ours. Instead of the image update step, the authors minimize for the parameters of $\tau$. Due to the bijection constraint, the minimization with respect to the weights is a combinatorial linear programming problem, known as the linear assignment problem. It turns out that a solution can be found by relaxing the binary constraints to positivity constraints $w(x, y) \geq 0$.

This means, in conjunction with the two sum constraints, that $w(x, y)$ is a doubly stochastic matrix.

The resulting minimization problem on $w$ is a linear problem with convex constraints. To solve it the authors propose an interior point method using a barrier function for the positivity constraint. This yields the following energy

$$\mathcal{G}(\tau, w) = \sum_x \sum_y w(x, y) d(x, \tau(y)) - T \sum_x \sum_y w(x, y) \log w(x, y),$$

subject to $\sum_x w(x, y) = 1$ and $\sum_y w(x, y) = 1$.

In this case the entropy term is motivated as a barrier function in an interior point method for the minimization of $\mathcal{D}$ with respect to the weights. Following that reasoning, the cooling of $T$ should be performed for each minimization with respect to $w$. This would require, for each weights update step, the minimization of energy $\mathcal{G}_T(\tau, \cdot)$ several times with decreasing $T$, before updating $\tau$. Instead the authors propose to perform the cooling as an outer loop, decreasing $T$ while they alternate between the minimizations with respect to $\tau$ and $w$.

## 5.4 Revisiting related work

Let us briefly review the connections of our work with other inpainting algorithms and also with existing variational models of non-local regularization which have been proposed in contexts such as image denoising.

The methods presented in Kawai et al. (2009); Wexler et al. (2007) are closely related to the patch NL-means scheme of (3.8), based in the optimization of an energy as in equation (1.2). In Wexler et al. (2007) the energy is interpreted as a likelihood for a MRF model. The method can be seen as an approximate EM algorithm for maximizing the log-likelihood w.r.t. the pixels in $O$, and some approximations have to be taken to make the optimization tractable. Based on heuristics, the authors also propose to use more robust estimators than the mean for the synthesis of pixels. Within the framework here proposed, robust estimators (as the median) naturally result from particular choices of the patch error functions $E(\cdot)$. In Kawai et al. (2009) the authors propose modifications of the energy which improve the results, such as some spatial localization of the similarity weights and

brightness invariance. The latter is achieved by introducing a multiplicative constant that matches the mean illuminance between each pair of patches.

Similar ideas have also been applied in the context of texture synthesis. In Kwatra et al. (2005) the authors propose two energies: a gradient-based and a robust one. The former can be obtained as a particular case of the patch NL-Poisson when $T \to 0$ (in the sense discussed in Section 2.2), while the latter corresponds to using an $L^{0.8}$-norm, also with $T \to 0$.

The patch NL-means algorithm is also related to the manifold image models of Peyré (2009). Equation (3.8) can be split into two steps analog to the manifold and image projection steps of Peyré (2009). First, for each patch centered in $\widetilde{O}$ we compute a new patch as a weighted average of all patches in the complement, according to the patch similarity weights $p_u^{MS}(z) := \int_{\widetilde{O}^c} w(z,\hat{z}) p_{\hat{u}}(\hat{z}) d\hat{z}$ with $z \in \widetilde{O}$. Doing this for each hole position yields an incoherent ensemble of patches. The image is obtained by averaging these patches: $u(z) = \frac{1}{\mathcal{A}(\Omega_p)} \int_{\Omega_p} p_u^{MS}(z-y, y) dy$. We use a density model, instead of the manifold model of Peyré (2009). Indeed, $p_u^{MS}(x)$ is the *mean shift* operator applied to $p_u(x)$. It is known that the iteration of this operator corresponds to an adaptive gradient ascent of the Parzen estimate of a PDF Cheng (1995), which in this case is generated by the set of patches in the complement of the hole. The use of a density model entails some advantages, mainly from the computational point of view: learning a manifold model is computationally costly.

In the following we will comment on the relation of this model with recent works on non-local regularization.

The UINTA algorithm, presented in Awate and Whitaker (2006) is a non-local denoising algorithm that minimizes the entropy of the patches in the image. Casting this idea to the context of inpainting the UINTA's entropy is estimated as the sample mean

$$E_U(u) := -\int_{\widetilde{O}} \log\left[\int_{\widetilde{O}^c} \exp(-\frac{1}{h}\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2) d\hat{x}\right] dx$$

where the inner integral is the probability of the patch $p_u(x)$ obtained as a Parzen density estimate. The corresponding Euler-Lagrange equation can be solved with a fixed point iteration which coincides with the patch NL-means scheme (3.8). In Awate and Whitaker (2006) this energy is minimized by considering all patches as independent (disregarding the overlap between neighboring patches), and evolving each of them according to a gradient

descent of $E_U$. Then an image is formed with the centers of these new patches. The repetition of this process yields an iterative application of the pixel NL-means (*i.e.* the standard non-local means algorithm).

In Brox et al. (2008) the authors use a variational principle for deriving the iterated pixel NL-means regularizer, and show its application to the restoration of texture. The underlying energy corresponds to the quadratic penalty between the solution image $u$, and a pixel NL-means type average of the noisy input image $\hat{u}$. The weights for this average are computed using $u$. Due to the dependence of the weights with the regularized image $u$, the minimizer is no longer a weighted average as NL-means, but the solution of a nonlinear optimization problem. It is shown that if the derivative of the nonlinear component is neglected, the resulting Euler-Lagrange equation matches the proposed fixed point algorithm: the iterated NL-means regularizer.

In Pizarro et al. (2010) the authors presented a variational framework for image denoising consisting in non-local regularization and data adjustment terms. Inpainting could be performed by considering only the data term as follows:

$$E_P(u) := -\int_{\widetilde{O}} \int_{\widetilde{O}^c} \exp(E(p_u(x) - p_{\hat{u}}(\hat{x})))d\hat{x}dx$$

This energy is the same as the one adapted from the UINTA algorithm $E_U$, without the logarithm. In Pizarro et al. (2010) the Euler-Lagrange equation is solved with a fixed point iteration. This model has two differences with our framework. First it allows to use a more general nonlinearity for the computation of the weights other than the exponential. Second, even in the case of the exponential, the methods differ in the normalization, for instance, when $E$ is the squared $L^2$-norm, the resulting scheme is as the patch NL-means, with the unnormalized weights.

After its introduction in Arias et al. (2009), our model has been interpreted as a non-local self-similarity regularizer in Peyré et al. (2011), where in conjunction with appropriate data fitting terms it has been applied to the solution of inverse problems, including inpainting, super-resolution and compressive sensing. In Peyré et al. (2011) a different patch-error function $E$ is used, namely the $L^2$-norm between patches (without squaring it). This choice is motivated as a patch-wise version of their work Peyré et al. (2008) on non-local Total Variation Gilboa and Osher (2008); Lezoray et al. (2007); Zhou and Schölkopf (2005) with adaptive weights. This patch-wise non-local TV is defined as the $L^1$-norm of the non-local gradient of the patch

valued image $p_u : \Omega \to \mathbb{R}^{\Omega_p}$. The non-local gradient is defined as a function $\nabla_w p_u : \Omega \times \Omega \to \mathbb{R}^{\Omega_p}$ given by $\nabla_w p_u(x, \hat{x}) := w(x, \hat{x})(p_u(x) - p_u(\hat{x}))$. Thus, the patch-wise non-local TV reads

$$\|\nabla_w p_u\| := \int_\Omega \int_\Omega w(x, \hat{x}) \|p_u(x) - p_u(\hat{x})\|_2 d\hat{x} dx.$$

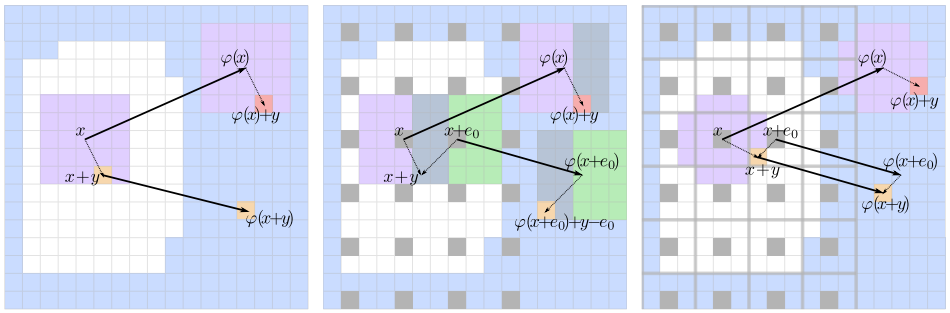
Note that in this sense, the model of the patch NL-medians corresponds to an anisotropic version of the non-local TV where the 2-norm in the integral is replaced by the 1-norm. Our work and the work of Peyré et al. (2011) are complementary. In Peyré et al. (2011) the regularization term is fixed, and the authors focus on the possibilities given by different data terms suited for different applications. On the other hand in this work we focus on the regularization term exploring its properties with different patch error functions $E$, and applying them to a problem in which the data term plays no role at all, since there is no data to adjust to.

### The model of Komodakis and Tziritas (2007) as a coarse correspondence model

As we commented in the introduction, several inpainting schemes can be considered as particular instances or relaxations of the inpainting functional proposed by Demanet et al. (2003). According to this model the unknown image is given by $u = \hat{u} \circ \varphi$, and is not considered a variable. The energy is given by

$$\begin{aligned} \mathcal{E}_{\mathrm{DSC}}(\varphi) &= \int_O \int_{\Omega_p} |\hat{u}(\varphi(x + h)) - \hat{u}(\varphi(x) + h)|^2 dh dx \\ &= \int_O \int_{\Omega_p} |u(x + h) - \hat{u}(\varphi(x) + h)|^2 dh dx. \end{aligned}$$

The pair-wise MRF model proposed by Komodakis and Tziritas (2007) does not fit into this formulation. In Komodakis and Tziritas (2007), the inpainting domain is covered by a coarse rectangular grid. The vertical and horizontal separation between nodes in the grid is $s$. Each node is connected by its four neighbors (left, right, up and down). We denote by $\mathcal{N}(a)$ the four neighborhood of $a$. If $a$ and $b$ are neighbors, we write $a \sim b$. At each grid node $a$ a known patch is attached, centered at position $\varphi(a) \in \widetilde{O}^c$. The size of these patches is $2s$. Such that the right half of the patch attached at node $a$, overlaps the left half of the patch attached at the




**(a)** $\mathcal{E}_{\text{DSC}}$ **(b)** $\mathcal{E}_{\text{KT}}$ **(c)** $\mathcal{E}_{\text{DSC-KT}}$

**Figure 5.1:** Comparison of variational inpainting models. (a) The energy $\mathcal{E}_{\text{DSC}}$ minimizes the error between the patch centered at $x$, and the one centered at $\varphi(x)$. The yellow pixel $x+y$ is copied from $\varphi(x+y)$, and has to be similar to the pixel the same relative position in the patch centered at $\varphi(x)$. (b) Energy $\mathcal{E}_{\text{KT}}$ minimizes the overlap error of patches of side $2s$ when attached at a grid where nodes (grey pixels) are separated by a gap of width $s$. (c) $\mathcal{E}_{\text{DSC-KT}}$ is a coarse version of $\mathcal{E}_{\text{DSC}}$, in which the mapping is block-wise a translation. The blocks are separated by gray lines. The domain of discrete cross shaped patch is shown.

right neighbor of $a$. The pair-wise potentials in the MRF penalize the $L^2$ error between overlapping patches, in the area of overlap. Let us denote by $\Omega_p(a, 2s) \subset \Omega$, a square neighborhood of side $2s$ centered at node $a$. The interaction potential between neighboring nodes $a$ and $b$ is given by

$$V_{a,b}(\varphi(a), \varphi(b)) = \int_{\Omega_p(a,2s)\cap\Omega_p(b,2s)} (\hat{u}(\varphi(a) + x - a) - \hat{u}(\varphi(b) + x - b))^2 dx.$$

Patches attached at nodes close to the boundary will intersect the known part of the image $O^c$. A node potential is defined which measures the squared $L_2$ error in the overlap area:

$$V_a(\varphi(a)) = \int_{\Omega_p(a,2s)\cap O^c} (\hat{u}(x) - \hat{u}(\varphi(a) + x - a))^2 dx$$

The energy is then defined as

$$\mathcal{E}_{\text{KT}}(\varphi) = \sum_a V_a(\varphi(a)) + \sum_{a\sim b} V_{a,b}(\varphi(a), \varphi(b))$$

In the following we show that a coarse version of $\mathcal{E}_{\text{KT}}$ can be written as a MRF similar to $\mathcal{E}_{\text{KT}}$. To that aim, we consider a coarse rectangular grid as

in Komodakis and Tziritas (2007), in which nodes are separated by a gap of size $s$ horizontally and vertically.

For each grid node $a$, we define a square patch of size $s$ (half the size of the patch used by Komodakis and Tziritas (2007)), denoted $\Omega_p(a, s)$. We constrain the correspondence map $\varphi$ to be a translation in each $\Omega_p(a, s)$, thus

$$\varphi(x) = \varphi(a) + x - a, \qquad \forall x \in \Omega_p(a, s).$$

With this correspondence map, the image defined as $u(x) = \hat{u}(\varphi(x)) = \hat{u}(\varphi(a) + x - a)$, *i.e.* by copying at $\Omega_p(a, s)$ the patch at $\varphi(a)$.

We also define a cross-shaped patch $\Omega_c(0, s)$, by adding to the four sides of $\Omega_p(a, s)$ a rectangle of sides $s$ and $s/2$ (see Figure REF). We use this cross-shaped patch to compare patches, whereas the square patch is only used to specify the regions in which $\varphi$ is a translation.

To compute a correspondence map defined over the nodes, we modify energy on $\mathcal{E}_{\text{DSC}}$ in two ways. First, by using the cross-shaped patch $\Omega_c$ to compare patches, and second by summing only over the grid. This yields the following energy:

$$\begin{aligned}\mathcal{E}_{\text{DSC-KT}}(\varphi) &= \sum_a \int_{\Omega_c(0,s)} (\hat{u}(\varphi(a+h)) - \hat{u}(\varphi(a)+h))^2 dh \\ &= \sum_a \int_{\Omega_c(a,s)} (\hat{u}(\varphi(x)) - \hat{u}(\varphi(a)+x-a))^2 dx\end{aligned}$$

Let us consider a term of the above sum for a node $a$. We assume that $\Omega_c(a, s)$ does not intersect the known part of the image. The cross-shaped patch $\Omega_c(a, s)$ intersects five square patches: $\Omega_p(a, s)$, and half of the patch attached to each of the four neighboring nodes. The term corresponding to $\Omega_p(a, s)$ vanishes, since in $\Omega_p(a, s)$, $\varphi(x) = \varphi(a) + x - a$. Thus we have

$$\int_{\Omega_c(a,s)} (\hat{u}(\varphi(x)) - \hat{u}(\varphi(a)+x-a))^2 dx = \sum_{b \in \mathcal{N}(a)} W'_{a,b}(\varphi(a), \varphi(b))$$

where

$$W'_{a,b}(\varphi(a), \varphi(b)) = \int_{\Omega_c(a,s) \cap \Omega_p(b,s)} (\hat{u}(\varphi(b)+x-b) - \hat{u}(\varphi(a)+x-a))^2 dx.$$

For each neighboring node $b \in \mathcal{N}(a)$, $W'_{a,b}(\varphi(a), \varphi(b))$ couples $\varphi(b)$ with $\varphi(a)$. The set $\Omega_c(a) \cap \Omega_p(b)$, corresponds to the half of the square patch $\Omega_p(b)$

adjacent to the square patch at $a$. Similarly, since $a$ is also a neighboring node of $b$, *i.e.* $a \in \mathcal{N}(b)$, there will be a second term coupling $\varphi(b)$ and $\varphi(a)$, given by:

$$W'_{b,a}(\varphi(b), \varphi(a)) = \int_{\Omega_c(b,s) \cap \Omega_p(a,s)} (\hat{u}(\varphi(a) + x - a) - \hat{u}(\varphi(b) + x - b))^2 dx$$

Note that the domain now is the half of $\Omega_p(a)$ adjacent to $b$. There are no other terms coupling the correspondences of $a$ and $b$, thus the full coupling is given by the following interaction potential:

$$\begin{aligned} W_{a,b}(\varphi(a), \varphi(b)) &= W'_{b,a}(\varphi(b), \varphi(a)) + W'_{a,b}(\varphi(a), \varphi(b)) \\ &= \int_{\Omega_c(a,s) \cap \Omega_c(b,s)} (\hat{u}(\varphi(a) + x - a) - \hat{u}(\varphi(b) + x - b))^2 dx \end{aligned}$$

Thus, we see that the terms in energy $\mathcal{E}_{\text{DSC-KT}}$ for which there is no intersection with the known portion of the image give rise to interaction terms very similar to those of $\mathcal{E}_{\text{KT}}$. In both cases the potential measures the error between patches in the region in which they overlap when attached to neighboring nodes at the grid. In Komodakis and Tziritas (2007) they use a square patch of size $2s$, and the area of overlap corresponds to a rectangular patch of $s \times 2s$ between nodes $a$ and $b$. In our case, it is the overlap area of two cross-shaped patches, resulting in a square patch $s \times s$ located between $a$ and $b$. Let us mention that if we used square patches of size $2s$ instead the cross-shaped patch in the energy $\mathcal{E}_{\text{DSC-KT}}$, then each node would have interactions with its eight nearest neighbors in the grid.

It can seen that when the cross-shaped patch at a node $a$ intersects $O^c$, we have a single node potential given by

$$W_a(\varphi(a)) = \int_{\Omega_c(a,s) \cap O^c} (\hat{u}(x) - \hat{u}(\varphi(a) + x - a))^2 dx.$$

In this case, the interaction potentials between $a$ and $b$ have to be computed over $\Omega_c(a, s) \cap \Omega_c(b, s) \cap O$. This is different from the formulation in Komodakis and Tziritas (2007) where the interaction potential between patches intersecting $O^c$ in computed in the same way as for patches that do not intersect $O^c$.

Chapter 6

# Implementation

In this chapter we describe some aspects of the implementation of the inpainting schemes described in the previous chapters. In Section 6.2 we describe the implementation of the image update step for the four inpainting schemes, considering a general $T \geqslant 0$. The implementation of the patch NL-means, patch NL-medians and patch NL-Poisson is straightforward. The patch NL-gradient medians deserves more attention. In Section 6.3 we describe the implementation of the weights update step, which is critical for performance. When $T = 0$, it amounts to the computation of a correspondence map as a nearest neighbor field. We describe PatchMatch, an efficient algorithm for computing an approximate nearest neighbor field (Barnes et al. (2009)), and provide a bound on its convergence rate. Finally, in Section 6.4 we describe a multiscale scheme, necessary to avoid bad local minima.

## 6.1 Discrete setting and notation

Throughout this chapter we consider discrete images defined over a rectangular bounded domain $\Omega := \{0, \dots, N\}^2$ and conrresponding discrete versions of the inpainting domain and its complement, $\widetilde{O} = \widetilde{O} \cap \mathbb{Z}^2$ and $\widetilde{O}^c = \widetilde{O}^c \cap \mathbb{Z}^2$. To avoid a cumbersome notation, we slightly modify it in this chapter (for instance some arguments of functions will be denoted as

subindices). The discrete energy reads:

$$\begin{aligned} &\mathcal{E}_{E,T}(u,w) = \mathcal{U}_E(u,w) - T\mathcal{H}(w), \\ &\text{subject to } \sum_{\hat{x}\in\widetilde{O}^c} w_{x\hat{x}} = 1,\ \forall x \in \widetilde{O}, \end{aligned} \tag{6.1}$$

where

$$\mathcal{U}_E(u,w) = \sum_{x\in\widetilde{O}} \sum_{\hat{x}\in\widetilde{O}^c} w_{x\hat{x}} E(p_u(x) - p_{\hat{u}}(\hat{x})), \tag{6.2}$$

and

$$\mathcal{H}(w) = -\sum_{x\in\widetilde{O}} \sum_{\hat{x}\in\widetilde{O}^c} w_{x\hat{x}} \log w_{x\hat{x}}.$$

Patches are now functions defined on a discrete domain $\Omega_p = \Omega_p \cap \mathbb{Z}^2$. The patch error functions $E$ is defined as the weighted sum of pixel errors

$$E(p_u(x) - p_{\hat{u}}(\hat{x})) = \sum_{h\in\Omega_p} g_h \mathrm{e}(u_{x+h} - \hat{u}_{\hat{x}+h}),$$

where $\mathrm{e} : \mathbb{R} \to \mathbb{R}^+$, or gradient errors

$$E(p_u(x) - p_{\hat{u}}(\hat{x})) = \sum_{h\in\Omega_p} g_h \mathrm{e}(\nabla u_{x+h} - \nabla \hat{u}_{y+h}).$$

where $\mathrm{e} : \mathbb{R}^2 \to \mathbb{R}^+$. Here, $g : \mathbb{Z}^N \to \mathbb{R}^+$ denotes the *intra-patch* kernel function, which is nonnegative, with support $\Omega_p$ and $\sum_{h\in\mathbb{Z}^N} g_h = 1$.

We need to define the discrete version of the gradient and divergence operators. We will use the notation $[A \to B] = \{f : A \to B\}$, the set of functions from $A$ to $B$.

We define $\nabla : [\Omega \to \mathbb{R}] \to [\Omega \to \mathbb{R}^2]$ as

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

and similarly for the component $\nabla u^2_{i,j}$. The notation $z = (i,j)$ refers to for locations on the image. Let us also define a divergence operator $\nabla\cdot : [\Omega \to \mathbb{R}^2] \to [\Omega \to \mathbb{R}]$,

$$\nabla \cdot p_{i,j} := \begin{cases} p^1_{i,j} & \text{if } i = 0, \\ -p^1_{i-1,j} & \text{if } i = N, \\ p^1_{i,j} - p^1_{i-1,j} & \text{otherwise,} \end{cases} + \begin{cases} p^2_{i,j} & \text{if } j = 0, \\ -p^2_{i,j-1} & \text{if } j = N, \\ p^2_{i,j} - p^2_{i,j-1} & \text{otherwise.} \end{cases}$$

These operators incorporate Neumann boundary conditions on the boundary of $\Omega$, and are dual operators, *i.e.* denoting $\langle \cdot, \cdot \rangle$ the usual scalar products in $[\Omega \to \mathbb{R}]$ and $[\Omega \to \mathbb{R}^2]$, $\langle \nabla u, p \rangle = -\langle u, \nabla \cdot p \rangle$, for all $u \in [\Omega \to \mathbb{R}]$, $p \in [\Omega \to \mathbb{R}^2]$.

To work with gradients in the discrete setting we need to define $O_e$ as the set of pixels with at least one neighbor in $O$ (according to the 4-connectivity). If we think as the discrete image as a lattice of nodes joined by edges, all variable edges are between nodes in $O_e$. Correspondingly, $\widetilde{O}$ is defined as the set of centers of patches that interesect $O_e$.

## 6.2 Image update step

For the computation of the image update step, we will rewrite the energy using the pixel influence weights $m : O_e \times O_e^c \to \mathbb{R}^+$:

$$m_{z\hat{z}} = \sum_{h \in \Omega_p} g_h w_{z-h,\hat{z}-h} \chi_{\widetilde{O}}(\hat{z} - h)$$

### Patch NL-means and patch NL-Poisson

The discrete version of the image energy term for patch NL-Poisson is given by

$$\mathcal{U}_{\lambda,2}(u) = \sum_{z \in O_e} \sum_{\hat{z} \in O_e^c} m_{z\hat{z}} [(1-\lambda)|\nabla u_z - \nabla \hat{u}_{\hat{z}}|^2 + \lambda (u_z - \hat{u}_{\hat{z}})^2]$$

The energy is quadratic in $u$ and can be minimized by solving the following linear equation:

$$\begin{cases} (1-\lambda)\nabla \cdot [k_z \nabla u_z] - \lambda k_z u_z = (1-\lambda)\nabla \cdot [k_z \boldsymbol{v}_z] - \lambda k_z f_z, & z \in O_e, \\ u(z) = \hat{u}(z), & z \in O_e \setminus O. \end{cases} \tag{6.3}$$

where he have defined

$$f_z := \frac{1}{k_z} \sum_{\hat{z} \in O_e^c} m_{z\hat{z}} \hat{u}_{\hat{z}} \qquad \text{and} \qquad \boldsymbol{v}_z := \frac{1}{k_z} \sum_{\hat{z} \in O_e^c} m_{z\hat{z}} \nabla \hat{u}_{\hat{z}}.$$

Equation (6.3) can be solved efficiently with a conjugate gradient scheme. When $\lambda = 1$, then the solution becomes simply $u_z = f_z$, and can be computed directly.

### Patch NL-medians

The discrete version of the image energy term for patch NL-Poisson is given by

$$\mathcal{U}_1(u) = \sum_{z \in O} \sum_{\hat{z} \in O^c} m_{z\hat{z}} |u_z - \hat{u}_{\hat{z}}|.$$

This problem can be solved for each $z$ independently, and its solution is given by a median of the known values $\hat{u}_{\hat{z}}$ weighted by $m_{z\hat{z}}$.

### Patch NL-gradient medians

With the notation defined above, we write the discretized image energy term as

$$\mathcal{U}_{\lambda,1}(u) = (1-\lambda) \sum_{z \in O_e} \sum_{\hat{z} \in O_e^c} m_{z,\hat{z}} |\nabla u_z - \nabla \hat{u}_{\hat{z}}| + \lambda \sum_{z \in O} k_z (u_z - f_z)^2,$$

with the same boundary conditions as with the patch NL-Poisson. When $\lambda > 0$, the second sum acts as a quadratic data attachment to $f_z$, patch NL-means' image update. Following Almansa et al. (2006); Chambolle (2004) we derive here a fixed point algorithm to minimize this energy.

In the case when $\lambda = 0$, the data attachment term vanishes and the fixed point algorithm cannot be directly applied. In that case, the resulting energy can be minimized with an implicit gradient descent. Given an iterate $u^t$, the next is computed as

$$u^{t+1} = \arg\min_u \sum_{z \in O_e} \sum_{\hat{z} \in O_e^c} m_{z,\hat{z}} |\nabla u_z - \nabla \hat{u}_{\hat{z}}| + \frac{1}{2\delta t} \sum_{z \in O} (u_z - u_z^t)^2.$$

This energy for $u^{t+1}$ is of the same type as $\mathcal{U}_{\lambda,1}$ for $\lambda > 0$. Thus, at each step $u^{t+1}$ is computed using the same fixed point algorithm, which we describe next.

### Fixed point algorithm

Let us define $U := [O_e \to \mathbb{R}]$ and $V := [O_e \to \mathbb{R}^2]$. In these spaces we will consider the usual scalar products and will denote them as $\langle \cdot, \cdot \rangle_U$ and $\langle \cdot, \cdot \rangle_V$. Notice that, with these definitions, the gradient and divergence are not dual operators when restricted to $O_e$, *i.e.* in general $\langle u, \nabla \cdot p \rangle_U \neq -\langle \nabla u, p \rangle_V$.

We are looking for a completion $u \in \widehat{U} := \{u \in U : u_z = \hat{u}_z, \forall z \in O_e \setminus O\}$ as the solution of the following problem:

$$\min_{u \in \widehat{U}} \sum_{z \in O_e} \sum_{\hat{z} \in D} m_{z\hat{z}} |\nabla u_z - v_{\hat{z}}| + \frac{1}{2\delta t} \sum_{z \in O} c_z (u_z - f_z)^2, \tag{6.4}$$

where $v : D \to \mathbb{R}^2, m : O_e \times D \to \mathbb{R}^+ \cup \{0\}, f : O \to \mathbb{R}$ and $c : O_e \to \mathbb{R}^+$ are given. For the sake of generality we consider a generic domain $D$ instead of the complement $O_e^c$. This generalization is relevant, due for instance to computational considerations as will be discussed shortly. Let us remark that, since $c_z > 0$ for $z \in O_e$, the energy is convex and the minimum is unique.

For simplicity we extend $f$ over $O_e$ by defining $f_z = \hat{u}_z$ for $z \in O_e \setminus O$. We use the fact that for any $\eta \in \mathbb{R}^2$, $|\eta| = \sup_{|\varepsilon|<1} \eta \cdot \varepsilon$, and write the energy as:

$$\min_{u \in \widehat{U}} \sup_{|\varepsilon_{z\hat{z}}| \leqslant 1} \sum_{z \in O_e} \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}} \cdot (\nabla u_z - v_{\hat{z}}) + \frac{1}{2\delta t} \sum_{z \in O_e} c_z (u_z - f_z)^2.$$

We have defined the unit field $\varepsilon \in W := \{\varepsilon : O_e \times D \to \mathbb{R}^2\}$.

Interchanging the minimum with the supremum, we obtain the following expression for the optimum $u^*$ in the inpainting domain

$$u_z^* = f_z + \delta t c_z^{-1} \nabla \cdot \mathcal{L}\varepsilon_z, \qquad z \in O. \tag{6.5}$$

We have defined the linear operator $\mathcal{L} : W \to V$ as $\mathcal{L}\varepsilon_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}}$.

We can extend the expression (6.5) to $O_e$, by defining a new divergence operator

$$\overline{\nabla} \cdot : V \to U \quad \text{as} \quad \overline{\nabla} \cdot \varepsilon_z := \chi_z^O \nabla \cdot \varepsilon_z.$$

Here $\chi_z^O = 1$ if $z \in O$ and $\chi_z^O = 0$ otherwise. We also define a modified gradient $\overline{\nabla} : U \to V$ as the negative adjoint operator of $\overline{\nabla} \cdot$ , given by

$$\overline{\nabla} u_z := \nabla [\chi_z^O u_z].$$

Besides allowing to extend (6.5) these operators are relevant since, as opposed to the usual gradient and divergence operators, they are still dual when they are restricted to functions over $O_e$: $\langle \overline{\nabla} \cdot p, u \rangle_U = -\langle p, \overline{\nabla} u \rangle_V$, for $p \in V$, $u \in U$.

Substituting expression (6.5) in the energy, we arrive to the following dual formulation:

$$\min_{|\varepsilon_{z\hat{z}}|\leqslant 1} -\sum_{z\in O_e} \delta t \left( \mathcal{L}\varepsilon_z \cdot \nabla \left[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon\right]_z + \frac{1}{2}c_z^{-1}[\overline{\nabla}\cdot\mathcal{L}\varepsilon_z]^2 \right) - \sum_{z\in O_e}\left[\mathcal{L}\varepsilon_z\cdot\nabla f_z - \sum_{\hat{z}\in D} m_{z\hat{z}}\varepsilon_{z\hat{z}}\cdot v_{\hat{z}}\right].$$

Using that $\nabla[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon]_z = \overline{\nabla}[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon]_z$ for all $z\in\Omega$, we can rewrite the dual problem with the more compact expression

$$\min_{|\varepsilon_{z\hat{z}}|\leqslant 1} \sum_{z\in O_e}\left[\frac{\delta t}{2}c_z^{-1}[\overline{\nabla}\cdot\mathcal{L}\varepsilon_z]^2 - \mathcal{L}\varepsilon_z\cdot\nabla f_z + \sum_{\hat{z}\in D} m_{z\hat{z}}\varepsilon_{z\hat{z}}\cdot v_{\hat{z}}\right]. \tag{6.6}$$

Adding the Lagrange multipliers corresponding to the restrictions $|\varepsilon_{z\hat{z}}|\leqslant 1$, $\sum_{z,\hat{z}\in O_e\times D}\lambda_{z\hat{z}}(|\varepsilon_{z\hat{z}}|^2-1)$, and deriving with respect to $\varepsilon$ we get the Euler equation

$$m_{z\hat{z}}[-\nabla f_z + v_{\hat{z}} - \delta t\overline{\nabla}[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon]_z] + \lambda_{z\hat{z}}\varepsilon_{z\hat{z}} = 0. \tag{6.7}$$

The *Karush-Kuhn-Tucker* Theorem yields the existence of the Lagrange multipliers with values

$$\lambda_{z\hat{z}} = m_{z\hat{z}}|-\nabla f_z + v_{\hat{z}} - \delta t\overline{\nabla}(c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon)_z|.$$

The solution of (6.6) is computed with the semi-implicit scheme

$$\varepsilon_{z\hat{z}}^{t+1} = \frac{\varepsilon_{z\hat{z}}^t + \nu\, m_{z\hat{z}}[-\nabla f_z + v_{\hat{z}} - \delta t\overline{\nabla}[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon^t]_z]}{1+\nu\, m_{z\hat{z}}|-\nabla f_z + v_{\hat{z}} - \delta t\overline{\nabla}[c^{-1}\overline{\nabla}\cdot\mathcal{L}\varepsilon^t]_z|}, \tag{6.8}$$

where $\nu$ is a time step small enough for assuring the convergence of the fixed point iteration. Finally the solution of the primal problem is recovered with (6.5).

### Convergence

Let us first observe that $\nabla f_z - \overline{\nabla} f_z = \nabla[(1-\chi_z^O)f_z] = \nabla[\chi_z^{O_e\setminus O} f_z]$ for all $z\in\Omega$, which allows to express the dual energy (6.6) as

$$J^*(\varepsilon) := \frac{\delta t}{2}\|c^{-1/2}\overline{\nabla}\cdot\mathcal{L}\varepsilon + c^{1/2}f/\delta t\|_U^2 + \langle h,\varepsilon\rangle_W, \tag{6.9}$$

where $h_{z\hat{z}} := m_{z\hat{z}}(v_{\hat{z}} - \nabla[\chi_z^{O_e \setminus O} f_z])$ and $\langle \cdot, \cdot \rangle_W$ is the usual scalar product in $W$. This expression shows that, since $c_z > 0$, there is a unique minimum value for $\overline{\nabla} \cdot \mathcal{L}\varepsilon$. We prove now that the sequence defined in (6.8) converges to that minimum.

**Proposition 6.1.** *If $c_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \leqslant 1$ for $z \in O_e$ and the time step $\nu$ verifies $\nu < \frac{1}{4\delta t K}$, with*

$$K := \max_{i,j \in O_e} \left\{ \chi^O_{i,j} + \frac{c_{i,j}}{c_{i+1,j}} \chi^O_{i+1,j}; \chi^O_{i,j} + \frac{c_{i,j}}{c_{i,j+1}} \chi^O_{i,j+1} \right\}, \tag{6.10}$$

*the sequence $(\varepsilon^t)$ defined in (6.8) converges to a minimum of the energy (6.9).*

*Proof.* We only sketch the main steps of the proof. The derivations are similar to those in Almansa et al. (2006); Chambolle (2004). Let us first show that if $\nu < \frac{1}{4\delta t K}$, the energy $J^*$ decreases. Defining $\eta := \nu^{-1}(\varepsilon^{t+1} - \varepsilon^t)$ for a given $t \geqslant 0$, we have that the variation in the energy can be bounded by

$$\begin{aligned} J^*(\varepsilon^{t+1}) \leqslant & J^*(\varepsilon^t) + \frac{1}{2}\nu(\nu\delta t \|c^{-1/2}\nabla \cdot \mathcal{L}\eta\|_U^2 - \|\eta\|_W^2) \\ \leqslant & J^*(\varepsilon^t) + \frac{1}{2}\|\eta\|_W^2 \nu(4\nu\delta t K - 1), \end{aligned} \tag{6.11}$$

where $K$ is a constant such that $4K\|\eta\|_W^2 \geqslant \|c^{-1/2}\overline{\nabla} \cdot \mathcal{L}\eta\|_U^2$ for all $\eta \in W$. We will derive this constant later.

Therefore, by choosing $\nu < (4\delta t K)^{-1}$ the energy decreases with $\varepsilon^t$. Let $m := \lim_{t\to\infty} J^*(\varepsilon^t) \geqslant 0$. Let us prove now that $\varepsilon^t$ also converges. Since $|\varepsilon^t_{z\hat{z}}| \leqslant 1$ for all $z \in O_e$, $\hat{z} \in D$, $t \geqslant 0$, there exists a subsequence $(\varepsilon^{t_k})$ converging to $\bar{\varepsilon}$. From (6.8) we see that $(\varepsilon^{t_k+1})$ converges as well. Let $\bar{\varepsilon}' := \lim_{t_k\to\infty} \varepsilon^{t_k+1}$. Defining $\overline{\eta} := (\bar{\varepsilon}' - \bar{\varepsilon})/\nu$, we see that in the limit when $t_k \to \infty$,

$$J^*(\bar{\varepsilon}') \leqslant J^*(\bar{\varepsilon}) + \frac{1}{2}\|\overline{\eta}\|_W^2 \nu(4\nu\delta t K - 1).$$

Since $J^*(\bar{\varepsilon}') = J^*(\bar{\varepsilon}) = m$ we can conclude that $\overline{\eta} = 0$ and therefore $\bar{\varepsilon} = \bar{\varepsilon}'$. This implies that the whole sequence $(\varepsilon^t)$ converges. Furthermore the limit $\bar{\varepsilon}$ verifies the Euler equation (6.7), and thus is a minimizer of the dual energy $J^*$.

We now sketch the derivation of the expression for $K$. Let us define the following $c$-weighted norms: $\|u\|_c^2 := \langle c^{-1}u, u\rangle_U$ for $u \in U$, $\|\xi\|_c^2 := \langle c^{-1}\xi, \xi\rangle_V$

for $\xi \in V$ and $\|A\|_{cc}^2 := \sup_{\xi \in V} \left\{ \frac{\|A\xi\|_c^2}{\|\xi\|_\varepsilon^2} \right\}$ for operators $A : V \to U$. For any $\eta \in W$ we can write

$$\|c^{-1/2}\overline{\nabla} \cdot \mathcal{L}\eta\|_U^2 = \|\overline{\nabla} \cdot \mathcal{L}\eta\|_c^2 \leqslant \|\overline{\nabla} \cdot \|_{cc}^2 \|\mathcal{L}\eta\|_c^2.$$

On one hand we have that

$$\|\mathcal{L}\eta\|_c^2 \leqslant \|\eta\|_W^2 \sup_{z \in O_e} \left\{ c_z^{-1} \sum_{\hat{z} \in D} m_{z\hat{z}}^2 \right\} \leqslant \|\eta\|_W^2,$$

where first inequality is an application of *Cauchy-Schwarz*, in the second we have used that $\sum_i a_i^2 \leqslant (\sum_i |a_i|)^2$, and the fact that $c_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \leqslant 1$.

On the other hand, from the definition of the divergence operator, it can be shown that $\|\overline{\nabla} \cdot \|_{cc}^2 \leqslant 4K$. □

**Computational considerations**

Let us remark that for the case of inpainting ($D = O_e^c$) the domain of the dual variable $\varepsilon$ is $O_e \times O_e^c \subset \mathbb{R}^4$, and therefore the direct implementation of this algorithm is prohibitive (for large images with large inpainting domains $\varepsilon$ will not fit in memory). To circumvent this problem we threshold the pixel influence weights $m_{z\hat{z}}$, so as to keep for each $z$ only the $M$ largest contributions. This reduces the size of $\varepsilon$ to $M|O_e|$. Indeed, when $h \to 0$ the $m$ function is already sparse w.r.t. $\hat{z}$, being the number of non-zero elements less or equal than the patch size (in pixels). In our experiments we set $M$ to the size of the patch, capturing exactly the function $m$.

## 6.3 Computation of the Nearest Neighbor Field

For minimizing the functionals derived from (2.2), we use the alternating optimization schemes described in (Algorithms 1. Most of the computational load is caused by the updating of the weights. In this Section we discuss the convergence properties of PatchMatch, an algorithm recently introduced by Barnes et al. (2009), which we use to speed-up the computation of the similarity weights.

For $T > 0$, the computation of the weights is of order $\mathcal{O}(|\widetilde{O}||\widetilde{O}^c||\Omega_p|)$. This is also the case in the limit $T = 0$, namely for $\mathcal{E}_{E,0}$. In that case, as

shown in Proposition 2.6, there are minima given by probability-valued maps $\nu$ associated to a correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$. This allows us to express the energy directly in terms of the unknown map $\varphi$. Thus, when considering the optimization of $\mathcal{E}_{E,0}$ the weights update step is substituted by a minimization w.r.t. a correspondence map $\varphi$,

$$\varphi_x \in \arg\min_{\xi \in \mathbb{Z}^2} U_x(\xi), \quad \text{for all } x \in \widetilde{O},$$

where the energy $U_x$ corresponds to the patch error function

$$U_x(\xi) = \begin{cases} E(p_u(x) - p_{\hat{u}}(\xi)) & \text{if } \xi \in \widetilde{O}^c \\ +\infty & \text{otherwise.} \end{cases}$$

Although the patch error does not have to be a metric, we refer to $p_{\hat{u}}(\varphi_x)$ as the *nearest patch* or *nearest neighbor* of $p_u(x)$. Following Barnes et al. (2009), we refer to the correspondence map $\varphi : \widetilde{O} \to \widetilde{O}^c$ as the *nearest neighbor field* (NNF). A brute force search for the NNF also conveys $\mathcal{O}(|\widetilde{O}||\widetilde{O}^c||\Omega_p|)$ operations.

PatchMatch is a very efficient algorithm for approximating the NNF (Barnes et al. (2009)). The search for the nearest neighbor is performed simultaneously over the points in $\widetilde{O}$ based in the following heuristic: since query patches overlap, the *offset* $\varphi_x - x$ of a good match at $x$ is likely to lead to a good match for the adjacent points of $x$ as well. It is an iterative algorithm which starting from a random initialization, alternates between steps of propagation of good offsets and random search.

We need some notation and definitions. Pixels in $\widetilde{O} = \{x_1, x_2, \ldots, x_{|\widetilde{O}|}\}$ are sorted according to the lexicographical order in $\mathbb{Z}^2$. For any $x \in \widetilde{O}$, let $\mathcal{N}_4(x) = \{z \in \widetilde{O} : 0 < |z - x| \leqslant 1\}$ be its 4-neighborhood. We consider a transition probability kernel $Q : \widetilde{O}^c \times \mathcal{B} \to [0, 1]$, where $\mathcal{B}$ is a $\sigma$-algebra in $\widetilde{O}^c$ (the subsets of $\widetilde{O}^c$ in our discrete case). Finally, let us define the notation,

$$\eta \wedge_x \xi = \begin{cases} \eta & \text{if } U_x(\eta) \leqslant U_x(\xi) \\ \xi & \text{if } U_x(\eta) > U_x(\xi), \end{cases}$$

where $\xi, \eta \in \widetilde{O}^c$.

The PatchMatch algorithm is described in Algorithm 4. The computational order is $\mathcal{O}(|\Omega_p||\widetilde{O}|)$, whereas the memory requirements are of $\mathcal{O}(|\widetilde{O}|)$. For most applications, a few iterations after a random initialization are often sufficient. Our implementation, coded in $C$ without optimizations, running

on a notebook's dual core 1.8GHz CPU, takes between $3s$ to $4s$ to compute the correspondences for an image of $300 \times 255$, with a mask of $|\widetilde{O}| \approx 14.000$ pixels and $|\widetilde{O}^c| \approx 50.000$ pixels and a patch of size $7 \times 7$.

This algorithm can be extended to store queues of $L$ offsets in an $L$-Nearest Neighbors Field (Barnes et al. (2010)) (see also Arias et al. (2011)). This allows its application to the case $T > 0$, by truncating the support of $w(x, \cdot)$ to the $L$-nearest neighbors of $p_u(x)$. This increases the computational cost and the memory requirements by a factor of $L$. We note however, that although each iteration is more costly, the use of queues usually reduces the required number of iterations.

**Algorithm 4** PatchMatch with propagation of offsets.

*Initialization.* Choose $\varphi_x^0 \sim \mathcal{U}(\widetilde{O}^c)$, i.e., randomly with a uniform distribution.

For each $n \in \mathbb{N}$,

***Random search.*** For each $x \in \widetilde{O}$ draw $S\varphi_x^n \sim Q(\varphi_x^n, \cdot)$. Set

$$\varphi_x^{n+\frac{1}{2}} = \varphi_x^n \wedge_x S\varphi_x^n.$$

***Forward propagation.*** If $n$ is odd, for each $i = 1, \ldots, |\widetilde{O}|$, set

$$\varphi_{x_i}^{n+1} = \varphi_{x_i}^{n+\frac{1}{2}} \wedge_{x_i} \left( \bigwedge_{x_i} \{\varphi_{x_j}^{n+1} + x_i - x_j : x_j \in \mathcal{N}_4(x_i), j < i\} \right).$$

If $n$ is even, invert the direction of propagation (*backward propagation*).

## Convergence of the PatchMatch algorithm

In this Section we discuss the convergence properties of the PatchMatch algorithm (for $L = 1$, *i.e.* without considering queues). For simplicity we will prove the convergence result for a different version of the PatchMatch algorithm. The difference lies in the propagation step: In the original version of Barnes et al. (2009), a node $z$ will propagate the offset $\varphi_z - z$ to $x$. In

our simplified version, the absolute position $\varphi_z \in \widetilde{O}^c$ is propagated instead (Algorithm 5). The arguments for the simplified "position-propagation" case can be applied to the "offset-propagation" case (with more involved computations). The latter is much more relevant from the practical point of view. This is briefly discussed in Section 6.3. Before proceeding to the convergence result, it is necessary to add some additional structure.

We will consider that elements in $\widetilde{O}$ correspond to the vertices of a *directed acyclic graph* (DAG) $G = (\widetilde{O}, E)$, where $E \subset \widetilde{O} \times \widetilde{O}$ denotes the edge set. We define the edge set as follows:

$$E = \{(x, y) \in \widetilde{O} \times \widetilde{O} : y \in \mathcal{N}_4(x), x < y\}.$$

By $x < y$ we mean that $x$ precedes $y$ in the lexicographical order. Note that the lexicographical order is a topological order for the resulting DAG. Paths in $G$ will be denoted by $c = (c_1, \dots, c_{n_c}) \in \widetilde{O}^{n_c}$, where $n_c \in \mathbb{N}$ is the length of the path. Given any pair of nodes, $x, z \in \widetilde{O}$, we will denote by $\mathcal{P}(z, x)$ the set of paths from $z$ to $x$. A node $z \in \widetilde{O}$ is said to be an *ancestor* of $x$ if $\mathcal{P}(z, x) \neq \emptyset$. Note that if $z$ is an ancestor of $x$, then $z$ comes before $x$ in the lexicographical ordering. Similarly, $z$ is a *descendant* of $x$ if $\mathcal{P}(x, z) \neq \emptyset$ (*i.e.* $x$ is an ancestor of $z$). We will write $\mathcal{A}(x)$ and $\mathcal{D}(x)$ for the set of ancestors and descendants of node $x$, respectively.

**Algorithm 5** Propagation of positions.

Given $n \in \mathbb{N}$, and $\varphi^{n+\frac{1}{2}}$:

***Forward propagation.*** If $n$ is odd, for each $i = 1, \dots, |\widetilde{O}|$, set

$$\varphi_{x_i}^{n+1} = \varphi_{x_i}^{n+\frac{1}{2}} \wedge_{x_i} \left( \bigwedge_{x_i} \{\varphi_{x_j}^{n+1} : x_j \in \mathcal{N}_4(x_i), j < i\} \right).$$

If $n$ is even, invert the direction of propagation (*backward propagation*).

The following proposition provides a bound on the convergence rate (in probability) for Algorithm 5. Without loss of generality we will assume throughout this Section that $\min_\xi U_x(\xi) = 0$ for all $x \in \widetilde{O}$.

**Proposition 6.2.** *Assume that for each pair $x, y \in \widetilde{O}$, we have that $d_{x,y} := \|U_x - U_y\|_\infty < +\infty$. Assume that $\widetilde{O}^c$ is compact (and therefore finite) and*

*that $Q(x, A) > 0$, for all $x \in \widetilde{O}^c$, $A \subset \widetilde{O}^c$. Then, the sequence $(\varphi^n)$ defined by the PatchMatch algorithm converges to a minimizer of the total energy $U$, in the sense that*

$$\lim_{n\to\infty} P(U_x(\varphi_x^n) > \epsilon) = 0, \qquad \text{for all } \epsilon > 0, x \in \widetilde{O}.$$

*Moreover, we have that*

$$P(U_x(\varphi_x^{n+1}) > \epsilon) \leqslant \prod_{z\in\mathcal{A}(x)} C(z, \epsilon - \ell_{z,x}) P(U_x(\varphi_x^n) > \epsilon), \tag{6.12}$$

*where $\ell_{z,x}$ is the length of the minimal path from $z$ to $x$:*

$$\ell_{z,x} := \begin{cases} \min_{c\in\mathcal{P}(z,x)} \sum_{i=2}^{n_c} d_{c_{i-1},c_i} & \text{if } \mathcal{P}(z,x) \neq \emptyset, \\ +\infty, & \text{if } \mathcal{P}(z,x) = \emptyset, \end{cases}$$

*and for each $z \in \widetilde{O}$, $C(z, \cdot) : \mathbb{R} \to [0, 1]$ is a non-increasing function defined by:*

$$C(z, a) := \sup_{\eta\in\{U_z > a\}} Q(\eta, \{U_z > a\}).$$

*For $a > 0$, $C(z, a) < 1$.*

Before going into the proof, let us make a few remarks about the statement.

We require the supremum norm between the energies to be bounded. Although this seems a reasonable assumption for practical applications, it is violated in the case of the propagation of offsets. In that case, the $U$ energies, as a function of the offsets have different domains: an offset $\xi$ which is valid at $x$ ($x + \xi \in \widetilde{O}^c$) might fall out of $\widetilde{O}^c$ when translated to another point $z \in \widetilde{O}$. In the next section we show how this can be dealt with.

The other hypothesis is that $Q(x, A) > 0$ for all $x \in \widetilde{O}^c, A \subset \widetilde{O}^c$. In our discrete setting this implies that the random search can transition from any location in $\widetilde{O}^c$ to any other position in $\widetilde{O}^c$ with positive probability. This is sufficient for the algorithm to converge, as already noted by the authors.

Equation (6.12) states that probability that a node $x$ maintains an energy higher that $\epsilon$ after a complete iteration, decreases at least by a factor of

$$\prod_{z\in\mathcal{A}(x)} C(z, \epsilon - \ell_{z,x}).$$

This factor considers the random searches from all ancestors of $x$, which are the nodes that propagate their positions, directly or indirectly, to $x$.

The fact that the supremum norm between the energies of different nodes is bounded, implies that if the energy of $x$ is still above $\epsilon$, then necessarily, the energy of any of its ancestors $z$ have to be lower bounded by $\epsilon - \ell_{z,x}$ (we prove that bellow, in Lemma 6.3). Each ancestor $z$ has to have an energy greater than $\epsilon - \ell_{z,x}$ after the random search step, and therefore also before the random search. The quantity $C(z, \epsilon - \ell_{z,x})$ is an upper bound of the probability of drawing a random position with energy higher $\epsilon - \ell_{z,x}$ given that the current position has energy also higher than $\epsilon - \ell_{z,x}$.

As can be seen from (6.12), the efficiency of the PatchMatch is mostly given by the propagation steps, when nodes collaborate by sharing their findings. For comparison, consider a PatchMatch algorithm without propagation where each $\varphi_x$ is searched for independently for each $x \in \widetilde{O}$. In that case, the bound on the rate of convergence (6.12) reduces to

$$P(U_x(\varphi_x^{n+1}) > \epsilon) \leqslant C(x, \epsilon) P(U_x(\varphi_x^n) > \epsilon).$$

Thus, the speed-up given by the propagation corresponds to

$$\prod_{\substack{z \in \mathcal{A}(x) \\ z \neq x}} C(z, \epsilon - \ell_{z,x}).$$

Note that only those $z \in \mathcal{A}(x)$ with $\ell_{z,x} < \epsilon$ contribute to lower the bound.

The proof of Proposition 6.2 relies on the following lemma.

**Lemma 6.3.** *Assume that for each pair $x, y \in \widetilde{O}$, we have that $d_{x,y} := \|U_x - U_y\|_\infty < +\infty$. Let us consider an assignment $\varphi$ resulting from a propagation step. Then we have that for each pair of nodes $x, z \in \widetilde{O}$,*

$$U_x(\varphi_x) > \epsilon \quad \Rightarrow \quad U_z(\varphi_z) > \epsilon - \ell_{z,x}.$$

*Proof.* Let us consider a path $c \in \mathcal{P}(z, x)$. We have that for any $a > 0$,

$$U_{c_i}(\varphi_{c_i}) > a \quad \Rightarrow \quad U_{c_i}(\varphi_{c_{i-1}}) > a$$

(otherwise, $\varphi_{c_{i-1}}$ would have been *propagated* to node $c_i$). Since $\|U_{c_i} - U_{c_{i-1}}\|_\infty = d_{c_{i-1},c_i}$, we have that

$$U_{c_i}(\varphi_{c_i}) > a \quad \Rightarrow \quad U_{c_{i-1}}(\varphi_{c_{i-1}}) > a - d_{c_{i-1},c_i}.$$

A simple recursion results in

$$U_x(\varphi_x) > \epsilon \quad \Rightarrow \quad U_z(\varphi_z) > \epsilon - \sum_{i=2}^{n_c} d_{c_{i-1},c_i}.$$

Thus each path from $z$ to $x$ imposes a bound over $U_z(\varphi_z)$. The intersection of all of them is given by $U_z(\varphi_z) > \epsilon - \ell_{z,x}$. □

We now prove Proposition 6.2.

*Proof.* Let us consider $x \in \widetilde{O}$. Since $\varphi^{n+1}$ is the result of a search step followed by a propagation step, we can apply Lemma 6.3. Thus,

$$\begin{aligned} U_x(\varphi_x^{n+1}) > \epsilon \quad &\Rightarrow \quad U_z(\varphi_z^{n+1}) > \epsilon - \ell_{z,x}, \quad \forall z \in \widetilde{O}, \\ &\Rightarrow \quad U_z(\varphi_z^{n}) > \epsilon - \ell_{z,x} \text{ and } U_z(S\varphi_z^{n}) > \epsilon - \ell_{z,x}, \quad \forall z \in \widetilde{O}, \end{aligned}$$

where the last implication is due to the random search step. Taking probabilities we have that

$$\begin{aligned} P(U_x(\varphi_x^{n+1}) > \epsilon) &\leqslant P(U_z(\varphi_z^n) > \epsilon - \ell_{z,x} \text{ and } U_z(S\varphi_z^n) > \epsilon - \ell_{z,x}, \forall z \in \widetilde{O}) \\ &= \prod_{z \in \widetilde{O}} P(U_z(S\varphi_z^n) > \epsilon - \ell_{z,x} | U_z(\varphi_z^n) > \epsilon - \ell_{z,x}) \\ &\qquad\qquad P(U_z(\varphi_z^n) > \epsilon - \ell_{z,x}, \forall z \in \widetilde{O}) \\ &\leqslant \prod_{z \in \widetilde{O}} P(U_z(S\varphi_z^n) > \epsilon - \ell_{z,x} | U_z(\varphi_z^n) > \epsilon - \ell_{z,x}) P(U_x(\varphi_x^n) > \epsilon). \end{aligned}$$

The second equality is due to Bayes' formula for conditional probabilities, and to the conditional independence of the random searches given $\varphi^n$. In the latter inequality we have used that $P(U_z(\varphi_z^n) > \epsilon - \ell_{z,x}, \forall z \in \widetilde{O}) \leqslant P(U_x(\varphi_x^n) > \epsilon)$ since the l.h.s. corresponds to the probability of the intersection of several events, while the r.h.s. is the probability of only one of such events.

Let us now estimate a bound for the conditional probabilities in the last inequality. These measure the probability that a random search does not decrease the energy level. Let us defined the notation $\{U_z > a\} := \{\xi \in \widetilde{O} : U_z(\xi) > a\}$ to denote the upper level set of energy $a$. To simplify notation, given $z \in \widetilde{O}$ we denote by $P^{z,n}$ the probability distribution of $\varphi_z^n$ (*i.e.* $P(\varphi_z^n \in A) = P^{z,n}(A)$, for $A \in \mathcal{B}(\widetilde{O}^c)$). Then, by the definition of

conditional probability, we have:

$$\begin{aligned} P(U_z(S\varphi_z^n) > a | U_z(\varphi_z^n) > a) \\ &= \frac{1}{P(U_z(\varphi_z^n) > a)} P(U_z(S\varphi_z^n) > a \text{ and } U_z(\varphi_z^n) > a) \\ &= \frac{1}{P^{z,n}(\{U_z > a\})} \int_{\{U_z > a\}} P^{z,n}(d\eta) \int_{\{U_z > a\}} Q(\eta, d\xi) \\ &= \int_{\{U_z > a\}} \frac{P^{z,n}(d\eta)}{P^{z,n}(\{U_z > a\})} Q(\eta, \{U_z > a\}) \end{aligned}$$

Notice that $P_a^{z,n} := [P^{z,n}(\{U_z > a\})]^{-1} P^{z,n}$ is a probability when restricted to the upper level set $\{U_z > a\}$. Therefore, the following bound holds

$$P(U_z(S\varphi_z^n) > a | U_z(\varphi_z^n) > a) \leqslant \sup_{\eta \in \{U_z > a\}} Q(\eta, \{U_z > a\}) =: C(z, a).$$

The coefficient $C(z, a)$ a supremum of probabilities, thus $C(z, a) \in [0, 1]$. The function $C(z, \cdot) : \mathbb{R} \to [0, 1]$ is non-increasing, since for $a_1, a_2 \in \mathbb{R}$, $a_1 > a_2$ we have that $\{U_z > a_1\} \subseteq \{U_z > a_2\}$. Finally, since $U_z$ is non-negative with a minimum value of 0, for $a > 0$, we have that $Q(\eta, \{U_z > a\}) < 1$, for all $\eta \in \widetilde{O}^c$. Since $\widetilde{O}^c$ is compact (and therefore finite in the discrete setting) we have that $C(z, a) < 1$, for $a > 0$. □

**Registered propagation**

Let us now address the propagation of offsets (Algorithm 4). We extend the definition of the energies to $\mathbb{Z}^2$ by taking $U_x(\xi) = +\infty$ if $\xi \in \mathbb{Z}^2 \setminus \widetilde{O}^c$. For each pair of connected nodes $(z, x) \in E$ we are given a transformation $T_{x,z} : \widetilde{O}^c \to \mathbb{Z}^2$ (which is $T_{x,z}(\xi) = \xi - z + x$ in Algorithm 4). The *registered propagation* is defined as follows:

$$\varphi_{x_i}^{n+1} = \varphi_{x_i}^{n+\frac{1}{2}} \wedge_{x_i} \left( \bigwedge_{x_i} \{T_{x_i,z}(\varphi_z^{n+1}) : (z, x_i) \in E\} \right).$$

With analogous computations (slightly more involved), one can prove that this version of the PatchMatch also converges, with the following bound for the rate of convergence:

$$P(U_x(\varphi_x^{n+1}) > \epsilon) \leqslant \prod_{z \in \mathcal{A}(x)} C(z, \beta(T, z, x, \epsilon)) P(U_x(\varphi_x^n) > \epsilon), \tag{6.13}$$

where

$$\beta(T,z,x,\epsilon) = \max_{c\in\mathcal{P}(z,x)} \min_{i=1,\dots,n_c} \left\{ U^*(T,c,\epsilon)_i - \sum_{j=i+1}^{n_c} d(T)_{c_{j-1},c_j} \right\}, \quad (6.14a)$$

$$U^*(T,c,\epsilon)_i = \begin{cases} \epsilon & \text{if } i=1, \\ \min\{U_{c_{i-1}}(\xi) : \xi \in \widetilde{O}^c \setminus T^{-1}_{c_i,c_{i-1}}(\widetilde{O}^c)\} & \text{if } i=2,\dots,n_c, \end{cases} \quad (6.14b)$$

$$d(T)_{x,z} = \|U_x \circ T_{x,z} - U_z\|_{\infty, T^{-1}_{x,z}(\widetilde{O}^c)}. \quad (6.14c)$$

This is a consequence of the fact that we are allowing the transformations $T_{x,z}$ to map some $\xi \in \widetilde{O}^c$ outside $\widetilde{O}^c$, *i.e.* $T_{x,z}(\xi) \in \mathbb{Z}^2 \setminus \widetilde{O}^c$. If $T_{x,z}(\varphi_z) \in \mathbb{Z}^2 \setminus \widetilde{O}^c$, then it is not propagated to node $x$. Thus, the fact that $U_x(\varphi_x) \geq \epsilon$, implies that either $\varphi_z \in T^{-1}_{x,z}(\widetilde{O}^c)$ and $U_z(\varphi_z) > \epsilon - d(T)_{x,z}$, or $\varphi_z \in \widetilde{O}^c \setminus T^{-1}_{x,z}(\widetilde{O}^c)$, in which case, we do not have control over $U_z(\varphi_z)$ other than

$$U_z(\varphi_z) > \min\{U_z(\xi) : \xi \in \widetilde{O}^c \setminus T^{-1}_{x,z}(\widetilde{O}^c)\}.$$

This gives rise to the $U^*$ coefficients in (6.13).

Observe that in the case in which $T_{x,z}(\widetilde{O}^c) \subset \widetilde{O}^c$, for all $(z,x) \in E$, then one recovers

$$\beta(T,z,x,\epsilon) = \epsilon - \ell(T)_{z,x} = \epsilon - \min_{c\in\mathcal{P}(z,x)} \sum_{i=2}^{n_c} d(T)_{c_{i-1},c_i},$$

as in (6.12). The transformations $T_{x,z}$ should be chosen to lower the *edge cost* $d(T)_{x,z}$, by *registering* $U_x$ and $U_z$.

In Barnes et al. (2009) the propagation is performed with a transformation $T_{x,z}(\xi) = \xi - z + x$, which corresponds to the propagation of the offset $\xi - z$. Let us estimate the corresponding bound according to (6.13). We denote by $e_0 = (0,1)$ and $e_1 = (1,0)$. The parents of node $x \in \widetilde{O}$ are $x - e_i$ with $i = 0,1$, and correspondingly $T_{x,x-e_i}(\xi) = \xi + e_i$. Note that $\widetilde{O}^c \setminus T^{-1}_{x,x-e_i}(\widetilde{O}^c) = \widetilde{O}^c \setminus (\widetilde{O}^c - e_i)$. To simplify the discussion, we assume that

$$\min\{U_z(\xi) : \xi \in \widetilde{O}^c \setminus (\widetilde{O}^c - e_i)\} > \kappa > 0$$

for all $z \in \widetilde{O}$. Then, as can be seen from (6.14a) and (6.14b), for all $\epsilon < \kappa$, we have that $\beta(T,z,x,\epsilon) = \epsilon - \min_{c\in\mathcal{P}(z,x)} \sum_{i=2}^{n_c} d(T)_{c_{i-1},c_i}$.

For $d(T)_{x,x-e_i}$ we have

$$\begin{aligned} d(T)_{x,x-e_i} &= \|U_x T_{x,x-e_i} - U_{x-e_i}\|_{\infty, T^{-1}_{x,x-e_i}} \\ &= \sup_{\xi \in T^{-1}_{x,x-e_i}(\widetilde{O}^c)} |g * \mathrm{e}(u(T_{x,x-e_i}(x-e_i)+\cdot) - \hat{u}(T_{x,x-e_i}(\xi)+\cdot)) - \\ &\qquad\qquad g * \mathrm{e}(u(x-e_i+\cdot) - \hat{u}(\xi+\cdot))| \\ &\leqslant \sup_{\xi \in T^{-1}_{x,x-e_i}(\widetilde{O}^c)} |(g \circ T_{x,x-e_i} - g)| * \mathrm{e}(u(x-e_i+\cdot) - \hat{u}(\xi+\cdot)). \end{aligned} \tag{6.15}$$

We have used that $x = T_{x,x-e_i}(x-e_i)$ and that if $T$ is a translation $g * (f \circ T) = (g \circ T) * f$. The bound 6.15 corresponds to a patch error weighted by the kernel

$$\partial_i^+ g(h) := g \circ T_{x,x-e_i}(h) - g(h) = g(h+e_i) - g(h) \approx \partial_i g(h),$$

which is an approximation of the partial derivative of $g$. This is an interesting property, because if $g$ is smooth, $|\partial_i^+ g(h)|$ is small (recall that we should minimize $d(T)$ to maximize the coefficients $\beta$). This is essentially the Lipschitz estimate of Lemma A.2 in the present context.

The "propagation of offsets" exploits the overlap of neighboring patches in the image domain, suggesting that each node should be connected with its neighbors on the image grid. This supports the intuitions in Barnes et al. (2009).

## 6.4 Multiscale scheme

Exemplar-based inpainting methods show a critical dependence with the size of the patch. In Figure 6.1, we show completions obtained with patch NL-means using different patch sizes. We used a Gaussian intra-patch weight kernel $g$ with standard deviation $a$. The Figure shows two results with a small patch ($a = 4$) and one result with a large patch ($a = 19$). The latter is able to reproduce the periodic pattern of the lamps, but the completion is blurry due to the spacial overlap of the patches and presents many discontinuities at the boundary of the hole.

The results with the small patch (second and fourth columns) do not show these artifacts, but one of them has failed to reproduce the lamps. The

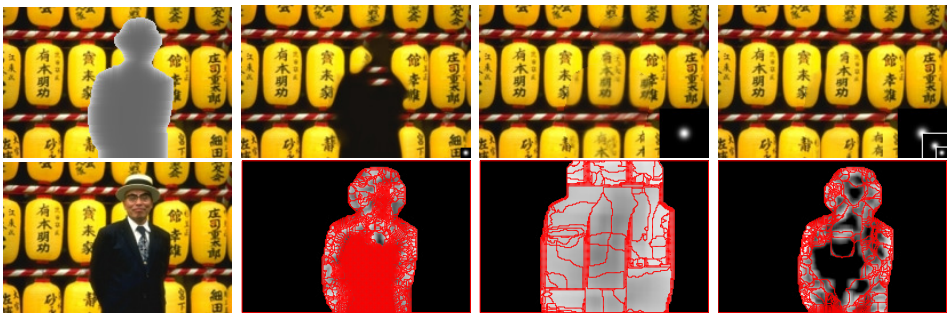


**Figure 6.1: Single scale vs. multiscale.** Left column: inpainting domain and initial condition. For the rest of the columns, from left to right: single scale inpainting with a $9 \times 9$ patch with $a = 4$; single scale inpainting with a $43 \times 43$ patch with $a = 19$; multi-scale inpainting with three scales, corresponding to patch sizes of $9 \times 9$ with $a = 4$, $21 \times 21$ with $a = 9$, and $43 \times 43$ with $a = 19$. All results have been computed with the patch NL-means scheme. The bottom row shows the boundaries between copy regions superimposed over the energy density image.

only difference between both is the initialization. The one in the second column was initialized with the original image, whereas the other one with the result obtained with the multiscale approach described in this section.

As in many state of the art exemplar-based inpainting methods (*e.g.* Kawai et al. (2009); Komodakis and Tziritas (2007); Wexler et al. (2007)), we will incorporate a multiscale scheme. This is usually motivated as an heuristic to avoid local minima, to find a good initialization and/or to alleviate the computational cost. However, as noted in Holtzman-Gazit and Yavneh (2008), as the example of Figure 6.1 suggests, inpainting is inherently a multiscale problem: images have structures of different sizes, ranging from large objects (as the lamps and their periodic distribution on the image) to fine scale textures (like the Chinese characters) and edges. The multiscale scheme responds to the fact that several patch sizes are needed to reproduce all these structures properly.

## Multiscale algorithm

In the following we describe the multiscale method we adopted, which goes along the lines of what is customary in the literature (Fang and Lien (2009); Kawai et al. (2009); Wexler et al. (2007)). It consists on applying sequentially the inpainting scheme on a Gaussian image pyramid, starting at the

coarsest scale. The result at each scale is upsampled and used as initialization for the next finer scale. The patch size is constant through scales.

Let us consider $S$ scales, the finest denoted with $s = 0$. We will specify the size of the image at the coarsest level $A_{S-1}$. Denoting the size of the image at the finest scale by $A_0$, we compute the sampling rate as $r := (A_0/A_{S-1})^{1/(S-1)} \in (0, 1)$. The width of the Gaussian filtering is associated to the subsampling factor as in Morel and Yu (2008). Let $a_0$ be the size of the patch and $\mathcal{E}_{a_0}$ the corresponding energy. We will add the superindex $s = 0, \ldots, S-1$ to the variables $u$ and $w$ to denote the scale. As before, the subindex 0 refers the initial condition, *i.e.* $u_0^s$ is the initial condition at scale $s$.

**Algorithm 6** Multiscale scheme.

**Require:** $u_0^S$, $S$, $a_0$ and $A_{S-1}$
1: Initialize: $(u^{S-1}, w^{S-1}) = \arg\min_{(u,w)} \mathcal{E}_{a_0}(u, w)$
2: **for** each scale $s = S-2, \ldots, 0$ **do**
3: Upsample $u^{s+1}$ to obtain $u_0^s$
4: $(u^s, w^s) = \arg\min_{(u,w)} \mathcal{E}_{a_0}(u, w)$
5: **end for**

The upsampling from $s+1$ to $s$ is obtained as in Wexler et al. (2007). The coarse weights $w^{s+1}$ are first interpolated to the finer image size, yielding $w_0^s$. These weights are then used to solve an image update step at the new scale: $u_0^s = \min_u \mathcal{E}_{a_0}(u, w_0^s)$. More conventional upsampling schemes by local interpolation (such as bilinear or splines) introduce a bias towards low-frequency non-textured regions. This exemplar-based upsampling avoids this bias.

Notice that keeping the patch size constant while filtering and reducing the image, is almost equivalent to enlarging the patch domain and filtering an image of constant size. The process can thus be seen as the sequential minimization of a series of inpainting energies with varying patch size given by $a_s = (1/r^s)a_0$, $s = 0, \ldots, S-1$, over a corresponding series of filtered images. In the coarsest scale $S-1$, a larger portion of the inpainting domain is covered by partially known patches. This makes the inpainting task easier and less dependent on the initialization. The energy at this scale should have fewer local minima. The dependency of the minimization process on the initial condition ensures that each single scale solution remains close to the coarse scale initialization. The multiscale algorithm *exploits* this

dependency to obtain an image $u^0$ which is approximately self-similar for all scales (or equivalently, for all patch sizes).

Figure 6.1 shows a comparison between single and multiscale results with the patch NL-means scheme. The multiscale result shows the benefits of large and small patch sizes. The missing lamps have been completed with the correct shape and spacing by the coarser stages, and the fine details are overall much less blurry and there are almost no discontinuities at the boundary of $O$. The bottom row shows the copy regions. The single scale results show a coarse partition with the large patch (the copying is more rigid), and one with many small regions with the smaller patch. The multiscale's NNF shows an intermediate partition, with some large regions inside of the hole and smaller ones around its boundary. The inpaintings at the finer scales work by refining the coarse partition obtained at coarser scales.

Chapter 7

# Experimental results

In this section we further demonstrate the performance of the proposed schemes on real inpainting problems and compare them with four representative state of the art methods. The images used were obtained from Komodakis and Tziritas (2007) and from the 100 images benchmark proposed by Kawai et al. (2009), available at `http://yokoya.naist.jp/research/inpainting/`.

## 7.1 Experimental setting

We consider four inpainting methods, variations of our proposed framework, namely patch NL-means, -medians, -Poisson and -gradient median. Gradient-based methods. In all cases we use the multiscale approach. To prevent blurring we set $T$, the selectivity of the similarity weights $w$, to $T = 0$. In this case, the weights select the nearest neighbor of each patch in $\widetilde{O}$. We use the CIE La*b* color space.

The weights update step dominates the computational load of the algorithms. We a variant use the PatchMatch algorithm Barnes et al. (2009) in Section 6.3, which estimates a list of the $L$ first nearest neighbors of each patch, and therefore can be used both for $T > 0$ and for $T \to 0$. The algorithm has a computational cost of $\mathcal{O}(|O| \times |\Omega_p| \times L)$ per iteration. Typically between 5 and 10 iterations are sufficient to obtain results comparable with the exhaustive search when using lists of size $L$ between 5 and 10.

For patch NL-means, -median and -Poisson, this represent a speed-up from 10min to less than 1min, for a 200x200 pixels image with a hole area of 20%, running on a 2Ghz CPU. The patch NL-gradient median may take longer, since the corresponding image update step can be time consuming, especially for small values of the mixing parameter $\lambda$ (see Appendix 6.2).

The results are shown in Figs. 7.1, 7.2 and 7.3, classified according to the nature of the inpainting problem. These figures include comparisons with the literature as well, as will be explained shortly below. Obtaining good results requires fixing the following parameters:

**Patch size.** For almost all experiments we used patches of size $s$ between $3 \times 3$ and $9 \times 9$. We used constant intra-patch weights ($g = 1/|\Omega_p|$), since the Gaussian ones require a larger support, reducing the available exemplars.

**Multiscale parameters.** The multiscale scheme has two parameters: the size of the coarsest image $A_{S-1}$ and the number of scales $S$. The former is the most critical one. In our experiments $A_{S-1}$ was set to a 20% of the original size, except for some cases which required less subsampling because there were too few exemplars in the hole's complement at the coarsest scale. The number of scales $S$ was set such that the subsampling rate $r = (A_0/A_{S-1})^{1/(S-1)} \approx (1/2)^{1/3} \approx 0.8$ as in Wexler et al. (2007).

**Confidence mask.** The confidence mask has two parameters, the asymptotic value $c_0$ and the decay time $t_c$. For all experiments we fix $c_0 = 0.1$ and used a decay time $t_c = 5$ except for small inpainting domains, in which we set $t_c = 1$.

For the mixing coefficient $\lambda$ of gradient-based methods we tested two configurations: low-$\lambda$ corresponding to $\lambda = 0.01$ and $\lambda = 0.001$ for patch NL-Poisson and patch NL-gradient medians, and high-$\lambda$ corresponding to $\lambda = 0.1$ and $\lambda = 0.01$ for patch NL-Poisson and patch NL-gradient medians. Recall that lower values of $\lambda$ give a higher weight to the gradient component of the energy. This is appropriate for structured images with strong edges. For the image in Figs. 7.1, 7.2 and 7.3 we used the same $\lambda$ configuration for both gradient-based schemes.

The rest of the parameters are also the same for the images in Figs. 7.2 and 7.3. Instead, for images in Fig. 7.1 we used different parameters for

intensity- and gradient-based methods. In these cases, image-based methods required larger patches than gradient-based ones.

For the sake of comparison we include here some results obtained using four representative methods of the state of the art. Two of them compute a coarse correspondence map, the *PatchWorks* (PW) method Pérez et al. (2004) and the approach of Komodakis and Tziritas (2007) (KT). The first one is greedy and the latter is iterative. Seams between blocks are eliminated a posteriori, using for instance a Poisson blending. The other two methods compute a dense correspondence map: *Resynthetizer* (R) Harrison (2005) (greedy) and the approach of Kawai et al. (2009) (KSY) (iterative). The latter is similar to the patch NL-means (see Section 5), with two improvements: locality of the nearest neighbor search and a correction that accounts for multiplicative brightness changes.

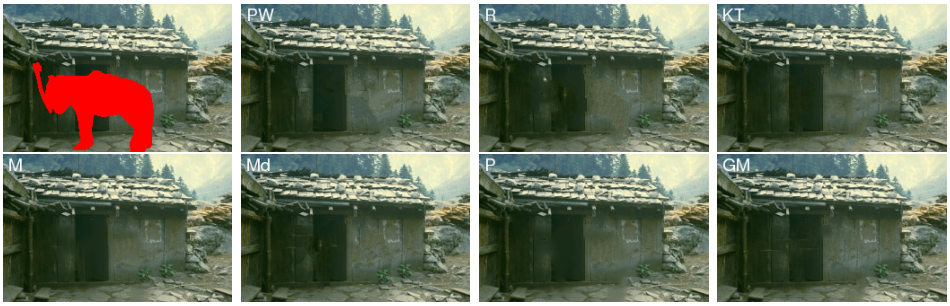
The results from KT and KSY were kindly provided to us by their authors. The results from KT were published in Komodakis and Tziritas (2007) and those from KSY can be found in Kawai et al. (2009) and are also available at `http://yokoya.naist.jp/research/inpainting/`. The Resynthetizer algorithm is implemented as a plug-in for the GIMP image editing software Kimball et al.. We used that implementation to generate the results labeled as "R". Our implementation of PatchWorks was written by Geoffrey Scoutheeten and was kindly made available to us by Simon Masnou. It does not include the blending post-processing step, so all seams are visible. We refer the reader to Pérez et al. (2004) and Cao et al. (2011) for results obtained using this efficient technique with blending.

## 7.2 Observations and comments

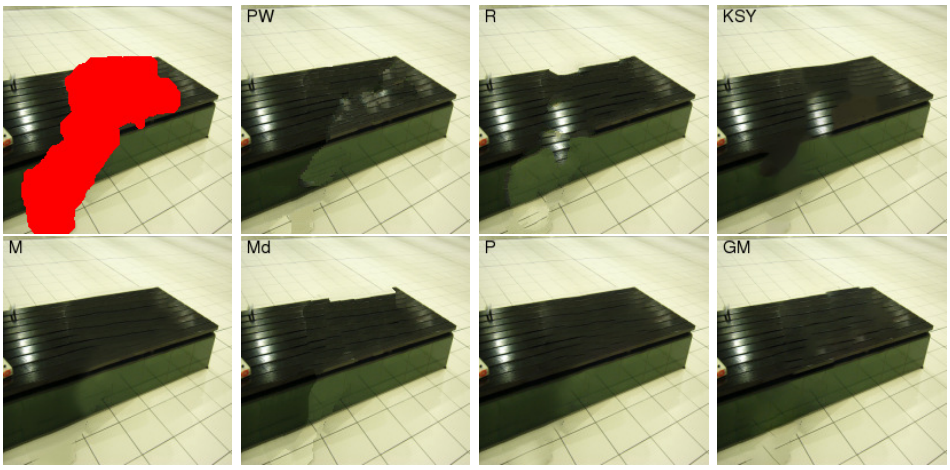
### Gradient vs. intensity

The proposed framework allows to design inpainting schemes based on the image gradient. So far, the gradient has mainly been used as a post-processing to remove seams between copy regions Pérez et al. (2004); Shen et al. (2005). Instead, in this framework, the gradient is used both for the image and weights update steps (also in Kwatra et al. (2005) for texture synthesis).

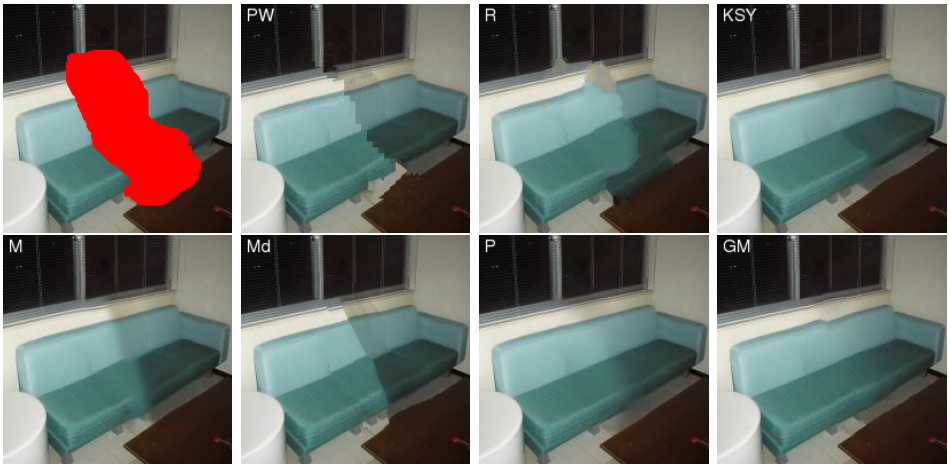
Gradient-based methods perform well in images with a strong structure (Figs. 7.1 and 7.3) but fail in images characterized by random textures. In




**(a)** *Cabin*




**(b)** *Station bench*




**(c)** *Sofa*

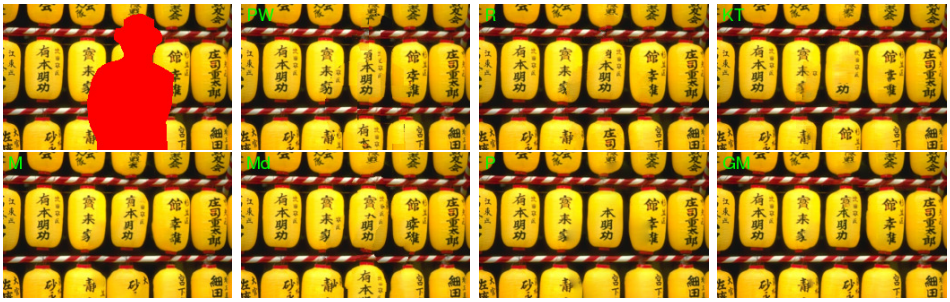
**Figure 7.1: Results on structured images.** PW: PatchWorks Pérez et al. (2004), R: Resynthesizer Harrison (2005), KT: method of Komodakis and Tziritas (2007) and KSY: method *et al.* Kawai et al. (2009). M, Md, P and GM stand for patch NL-means, -medians, -Poisson and -gradient medians.


**(a)** *Baseball*

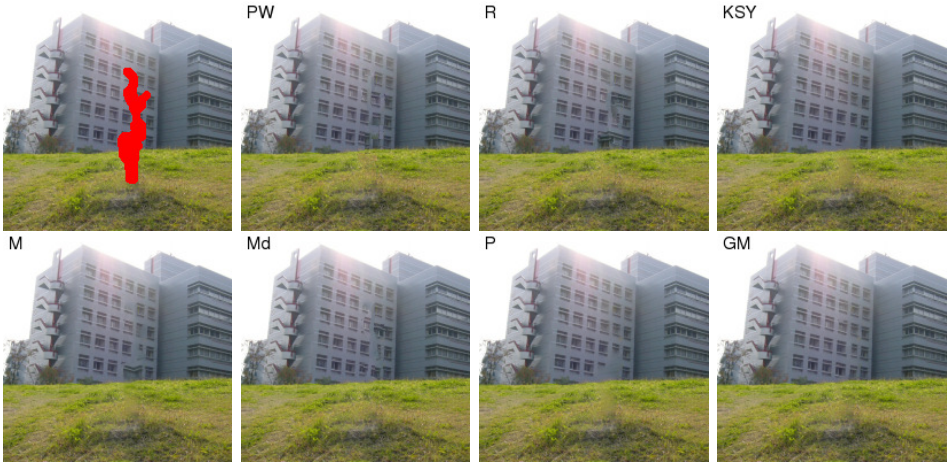

**(b)** *Bridge*


**(c)** *Golf*

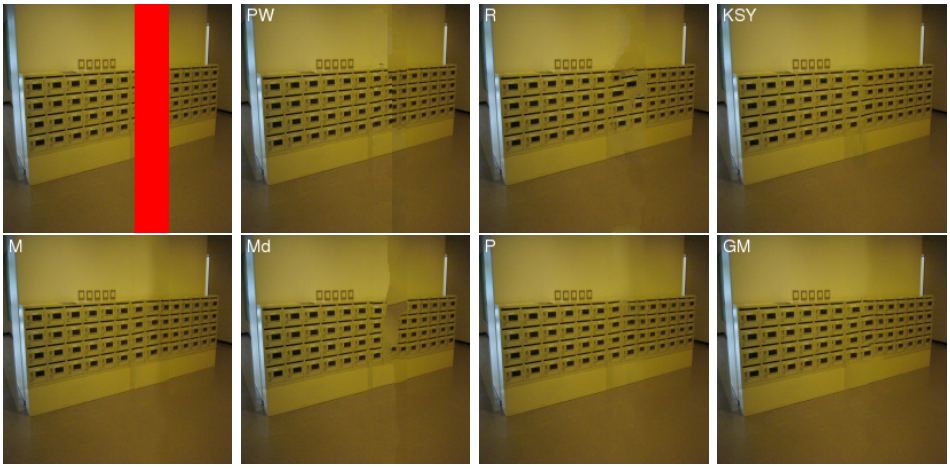
**Figure 7.2: Results on random textures.** PW: PatchWorks Pérez et al. (2004), R: Resynthesizer Harrison (2005) and KT: method of Komodakis and Tziritas (2007). M, Md, P and GM stand for patch NL-means, -medians, -Poisson and -gradient medians.



**(a)** *Matsuri*




**(b)** *Building*




**(c)** *Mailboxes*

**Figure 7.3: Results on periodic textures.** PW: PatchWorks Pérez et al. (2004), R: Resynthesizer Harrison (2005), KT: method of Komodakis and Tziritas (2007) and KSY: method of Kawai et al. (2009). M, Md, P and GM stand for patch NL-means, -medians, -Poisson and -gradient medians.

the following we are going to analyze the reasons for this behavior, discussing the benefits and limitations of using gradients both in the image and weights update steps. Many of the effects we are going to remark here have already been discussed in more detail in Chapter 4.

**Gradients in the image synthesis.** The image obtained with patch NL-Poisson and -gradient median is the result of copying gradients from the known portion of the image and then solving a PDE. Although we do not have a proof of it, in our experiments we observe that the synthesized image will not have edges which are not in $O^c$. This is not the case for intensity-based methods, which present discontinuities (seams) at the boundary of the hole and between copy regions. These seams are present in almost all images, but are particularly noticeable in images 7.1(b), 7.1(c), 7.3(a) and 7.3(c).

**Gradients in the patch similarity weights.** For patch NL-Poisson and -gradient median, the patch error is a combination of intensity and gradients. With the low values of $\lambda$ used, the gradient component dominates. For some textured images this may cause the method to fail. In *baseball* (Fig. 7.2(a)) for instance, segments of the sky have been reproduced in the snow. Fig. 7.4 shows results obtained with the extension to the gradient-based methods presented in Section 4.4. Different values of the mixture coefficient $\lambda$ are used for the image ($\lambda_u$) and weights update ($\lambda_w$). Results in Fig. 7.4 have $\lambda_w = 1$, *i.e.* the weights are computed based only on the image values. The image is updated using the corresponding image update step with a low value of $\lambda_u$. Such scheme is non-variational unless $\lambda_u = \lambda_w$. Although the results using $\lambda_w = 1$ improve for *baseball* and *bridge*, this will not be the case in general. As an example, the last row of Fig. 7.4 shows the result obtained for the *sofa* image. With such illumination changes, better results (as those shown in 7.1(c)) are obtained by giving more weight to the gradients in the patch comparison criterion (*i.e.* a low value of $\lambda_w$).

Gradient-based methods facilitate the prolongation of structures and edges, due to the reinforcement of local PDE diffusion and non-local propagation (in Section 4.3 we quantitatively compared the ability of each method to propagate a straight line). This allows to use smaller patches, which alleviates the computational load and amounts to more available exemplars and less blending due to patch overlap (either by averages or medians).

### Means vs. medians

It is notorious that $L^1$-based functionals perform better at the reproduction of fine texture. The results of the $L^2$ methods are smoothed by the spatial averaging of overlapping patches. On the other hand, patch NL-medians creates sharp discontinuities as in Fig. 4.1 when different copy regions meet (*e.g.* 7.1(b) and 7.1(c)). These discontinuities are very noticeable and in these cases some smoothing is desirable. This has been previously noted in Pérez et al. (2004), where the authors distinguish between color and texture seams. Color seams occur between copy regions with little texture having different colors (the results of PatchWorks and patch NL-medians in image 7.1(c) are a clear example). These seams are best removed with a smooth blending. On the other hand, texture seams require a sharp cut since a soft blending would smooth the texture.

The image synthesis obtained with patch NL-gradient medians offers an alternative to this dilema. The patch NL-gradient medians method perform sharp cuts between copy regions at the gradient level. As a gradient-based method does not introduce discontinuities, and texture is preserved since there is no blending in the transition bands between copy regions. This can be noticed in the results with patch NL-gradient medians for in images 7.4, 7.2(c) and in the grass in 7.3(b).

As was discussed in Section 4.3, $L^1$-based methods are more *stiff*, meaning that they show a higher dependence on the initialization (specially patch NL-medians). Although the confidence mask diminishes this effect, it still affects the results. This is the reason for some misalignments in straight lines, due to subsampling artifacts of the multiscale scheme (see for instance the results of patch NL-gradient median in images 7.1(b) and 7.1(c)). Also patch NL-medians generally requires the use of larger patches, particularly for structured images. This is not always possible, as in image 7.1(a), where we could not find a proper set of parameters. However we found a good result by after 3 iterations of the multiscale scheme.

### Which scheme to use?

There is not a clear answer to this question, since it largely depends on the image. Based on the previous observations, we can conclude that the gradient adds interesting features for the image synthesis in general, but its use in the patch similarity can be misleading for some strongly textured

**Figure 7.4: Results for non-variational gradient methods**. Results using the non variational extension of the patch NL-Poisson (left column) and patch NL-gradient median (right column) (see Section 4.4). The patch similarity is computed based solely on image values (setting $\lambda_w = 1$), whereas the image update step is mainly gradient-based by keeping $\lambda_u$ to a low value.

images. This leads us to consider different mixture parameters for the image and weights updates, $\lambda_u$ and $\lambda_w$ respectively, using the extension presented in Section 4.4. This simple modification adds more flexibility to gradient-based schemes, allowing to have the benefits of intensity- and gradient-based methods.

For textured images good results can be obtained with $\lambda_w \approx 1$ as shown in Fig. 7.4. For structured images (Figs. 7.1 and 7.3) low values of $\lambda_w$ will be beneficial, since this amounts to use a patch similarity measure invariant with respect to additive brightness changes. The value of $\lambda_u$ is not critical, as long as it is sufficiently small. In that case, the synthesis is dominated by the gradient-based term. Thus, this modification does not add an extra parameter.

Between patch NL-Poisson and -gradient medians, the latter yields a better reproduction of random textures. However its running times are still impractical.

Chapter 8

# Conclusions and future work

We presented a variational framework for exemplar-based image inpainting. The proposed formalism models in a unified manner the cases of probabilistic and one-to-one correspondences. We focus on the latter, more relevant for the current inpainting application. However, we provide a general presentation which evidences connections with related models for non-local image regularization and provides intuitive interpretations of the proposed energy.

We derived from the proposed framework four different inpainting schemes, corresponding to error functions based on the combinations of $L^1$- and $L^2$-norms, with image or gradient patches. For many images the inpainting achieved is a patch-work with arbitrarily shaped segments, which emerge during the minimization process. Transitions between these segments can be sharp or smooth depending on the used norm.

Not all images, show these copy regions, and in general the absence of copy regions is correlated with a bad result. We are currently researching on enforcing the correspondence map to be piece-wise translational, by adding a suitable regularization term in the weights update step.

One of the novelties of this work is the exploration of the use of gradients in exemplar-based inpainting methods. In our framework this implies not only copying gradients (gradient-based synthesis has been used before for removing seams between patches), but also using them for the computation of the similarity weights. Although this does not work well for images with random textures, we found that it provides some interesting benefits in

more structured images, such as additive brightness invariance and better propagation of structures.

The proposed functional shows a critical dependence with the patch size. Furthermore, it is non-convex and has many local minima. To tackle these issues we used a multiscale approach. We believe that inpainting is inherently a multiscale problem and are currently working on a variational formulations of multiscale inpainting.

Other direction of current and future research is the exploration of other patch error functions. In particular it would be interesting to study the patch comparison criteria based on structure tensors, which provide a more robust estimation of the morphological structure of the image. We can also consider correlation metrics such as the *Normalized Cross-Correlation* which could add invariance to multiplicative brightness changes (we thank the anonymous reviewer for suggesting this).

# Appendices of Part I

Appendix **A**

# Existence and regularity of minima for the similarity weights model

We shall consider here three models: patch NL-means, patch NL-medians and patch NL-Poisson.

Let us recall the notation we will use. Let $C_c(\mathbb{R}^N)$ be the set of continuous functions with compact support in $\mathbb{R}^N$. By $C_c(\mathbb{R}^N)^+$ we denote the set of nonnegative functions in $C_c(\mathbb{R}^N)$. As usual, if $Q$ is an open set we denote by $W^{1,p}(Q)$, $1 \leq p \leq \infty$, the space of functions $v \in L^p(Q)$ such that $\nabla v \in L^p(Q)^N$. By $W^{1,p}(Q)^+$ we denote the set of nonnegative functions in $W^{1,p}(Q)$. We denote by $W^{2,p}(Q)$ (resp. by $W^{2,p}_{\text{loc}}(Q)$), $1 \leq p \leq \infty$, the space of functions $v \in L^p(Q)$ such that $\nabla v \in L^p(Q)^N$ and $D^2 v \in L^p(Q)^{N \times N}$ (resp. the functions $v \in W^{2,p}(Q')$ for any subdomain $Q'$ included in a compact set of $Q$).

We assume here that $\Omega$ is a rectangle in $\mathbb{R}^N$ and $\hat{u} : O^c \to \mathbb{R}$ with $\hat{u} \in L^\infty(O^c)$. We assume that $u : \Omega \to \mathbb{R}$ is such that $u|_{O^c} = \hat{u}$. We also assume that $u$ is extended by symmetry and then by periodicity to $\mathbb{R}^N$. Let us assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, and that $g \in L^1(\mathbb{R}^N)^+$ with $\int_{\mathbb{R}^N} g(h)\, dh = 1$.

## Existence of minima for the patch NL-means model

In this Section we consider the patch NL-means model

$$\mathcal{E}_{2,T}(u,w) = \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 d\hat{x}dx + T\mathcal{H}(w). \qquad \text{(A.1)}$$

We implicitly understand that $\mathcal{E}_{2,T}(u,w) = +\infty$ in case that the second integral is not defined.

Let us recall the admissible class of functions

$$\mathcal{A}_{2,T} := \{(u,w) : u \in L^\infty(\Omega),\, u = \hat{u} \text{ in } O^c, w \in \mathcal{W}\}.$$

where

$$\mathcal{W} := \{w \in L^1(\widetilde{O} \times \widetilde{O}^c) : \int_{\widetilde{O}^c} w(x,\hat{x})\, d\hat{x} = 1 \text{ a.e. } x \in \widetilde{O}\}.$$

Our purpose is to prove the following results about the existence and regularity of minima of $\mathcal{E}_{2,T}$.

**Proposition A.1.** *Assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, that $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. Then there exists a minimum $(u,w) \in \mathcal{A}_{2,T}$ of $\mathcal{E}_{2,T}$. Moreover, for any minimum $(u,w) \in \mathcal{A}_{2,T}$ we have that $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

In other words, there are smooth minima and smooth probability distributions representing the fuzzy correspondences between $\widetilde{O}$ and $\widetilde{O}^c$. To prove Proposition A.1 we need the following Lemmas which will play also a key role in the rest of the paper.

**Lemma A.2.** *Assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. Assume that $u \in L^\infty(\widetilde{O} + \Omega_p)$. Then the functions*

$$\nabla_x g * (u(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2 \quad \text{and} \quad \nabla_{\hat{x}} g * (u(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2 \qquad \text{(A.2)}$$

*are uniformly bounded in $\widetilde{O} \times \widetilde{O}^c$ by a constant that depends on $\|\nabla g\|_{L^1}$, $\|u\|_\infty$, $\|\hat{u}\|_\infty$.*

*Proof.* Let $x \in \widetilde{O}$, $\hat{x} \in \widetilde{O}^c$. Observe that $\hat{x} + \Omega_p \subset O^c$. Let us observe that by approximating $u$ by smooth functions the following computations are easily justified. Using the $L^1$ integrability of $\nabla_h g$ and the boundedness of $u$, we compute

$$\begin{aligned}
& \nabla_{x+\hat{x}} \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))^2 \, dh \\
= \; & 2 \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_x u(x+h) - \nabla_{\hat{x}} \hat{u}(\hat{x}+h)) \, dh \\
= \; & 2 \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_h u(x+h) - \nabla_h \hat{u}(\hat{x}+h)) \, dh \\
= \; & - \int_{\mathbb{R}^N} \nabla_h g(h)(u(x+h) - \hat{u}(\hat{x}+h))^2 \, dh.
\end{aligned}$$

By our assumptions, the above integral is bounded by a constant that only depends on $\|\nabla g\|_{L^1}$, $\|u\|_\infty$ and $\|\hat{u}\|_\infty$. Now,

$$\begin{aligned}
& \nabla_{x-\hat{x}} \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))^2 \, dh \\
= \; & 2 \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_x u(x+h) + \nabla_{\hat{x}} \hat{u}(\hat{x}+h)) \, dh \\
= \; & 2 \int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_h u(x+h) + \nabla_h \hat{u}(\hat{x}+h)) \, dh \\
= \; & \int_{\mathbb{R}^N} g(h)(\nabla_h u(x+h)^2 - \nabla_h \hat{u}(\hat{x}+h)^2) \, dh \\
+ \; & 2 \int_{\mathbb{R}^N} g(h)(u(x+h)\nabla_h \hat{u}(\hat{x}+h) - \hat{u}(\hat{x}+h)\nabla_h u(x+h)) \, dh.
\end{aligned}$$

Integrating by parts, we have that the first term in the last equality is

bounded. Observe that

$$\begin{aligned}
&\int_{\mathbb{R}^N} g(h)(u(x+h)\nabla_h\hat{u}(\hat{x}+h) - \hat{u}(\hat{x}+h)\nabla_h u(x+h))\,dh \\
&\quad = -\int_{\mathbb{R}^N} g(h)((\hat{u}(\hat{x}+h)+1)\nabla_h u(x+h) - u(x+h)\nabla_h\hat{u}(\hat{x}+h))\,dh \\
&\qquad\qquad + \int_{\mathbb{R}^N} g(h)\nabla_h u(x+h)\,dh \\
&\quad = -\int_{\mathbb{R}^N} g(h)\nabla_h\left(\frac{u(x+h)}{(\hat{u}(\hat{x}+h)+1)}\right)(\hat{u}(\hat{x}+h)+1)^2\,dh \\
&\qquad\qquad - \int_{\mathbb{R}^N} \nabla_h g(h)u(x+h)\,dh \\
&\quad = \int_{\mathbb{R}^N} \nabla_h(g(h)(\hat{u}(\hat{x}+h)+1)^2)\frac{u(x+h)}{(\hat{u}(\hat{x}+h)+1)}\,dh \\
&\qquad\qquad - \int_{\mathbb{R}^N} \nabla_h g(h)u(x+h)\,dh \\
&\quad = \int_{\mathbb{R}^N} \nabla_h g(h)(\hat{u}(\hat{x}+h)+1)u(x+h)\,dh \\
&\qquad\qquad + 2\int_{\mathbb{R}^N} g(h)\nabla_h\hat{u}(\hat{x}+h)\,u(x+h)\,dh \\
&\qquad\qquad - \int_{\mathbb{R}^N} \nabla_h g(h)u(x+h)\,dh.
\end{aligned}$$

The first and third integrals are well defined and bounded by constants that only depend on $\|\nabla g\|_{L^1}$, $\|u\|_\infty$, $\|\hat{u}\|_\infty$. Let us check that the integral

$$\int_{\mathbb{R}^N} g(h)\nabla_h\hat{u}(\hat{x}+h)\,u(x+h)\,dh \tag{A.3}$$

is well defined and bounded. Assuming first that $u$ is continuous, then $\int_{\mathbb{R}^N} g(h)\nabla_h\hat{u}(\hat{x}+h)\,u(x+h)\,dh$ is well defined and bounded. Now, if $u \in L^\infty(\widetilde{O}+\Omega_p)$, we may approximate it by continuous functions $u_n$ converging to $u$ a.e. with $\|u_n\|_\infty \le \|u\|_\infty$. Then $g*(u_n(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 \to g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2$ a.e. and in $L^p$ for all $p \in [1,\infty)$. Since our statement holds for $u_n$ it also holds for $u$ and the above convergence is also uniform. The statement of the Lemma follows. □

**Remark 1.** *Notice that our mild assumption that $\hat{u} \in BV(O^c)$ forces us to consider the assumption that $\hat{u} \in L^\infty(O^c)$. Indeed this implies that $u$ is also bounded and we can give sense to the integral $\int_{\mathbb{R}^N} g(h)\nabla_h\hat{u}(\hat{x}+h)\,u(x+h)\,dh$.*

**Remark 2.** *If we assume that $u \in L^\infty(\widetilde{O})$ then the statement of Lemma A.2 holds for $x \in O$.*

**Lemma A.3.** *Under the assumptions of Proposition A.1, if $(u_n, w_n) \in \mathcal{A}_{2,T}$ is a minimizing sequence for $\mathcal{E}_{2,T}$ such that $u_n$ is uniformly bounded, then we may extract a subsequence converging to a minimum of $\mathcal{E}_{2,T}$.*

*Proof.* Let $(u_n, w_n) \in \mathcal{A}_{2,T}$ be a minimizing sequence of (A.1) such that $\{u_n\}_n$ is uniformly bounded. Since $\Omega$ is a bounded domain we have that

$$\int_{\widetilde{O}} \int_{\widetilde{O}^c} \chi_{\{w_n>1\}} w_n(x,\hat{x}) \log w_n(x,\hat{x}) d\hat{x}dx$$

is bounded. Hence $w_n(1+\log^+ w_n)$ is bounded in $L^1(\widetilde{O}\times\widetilde{O}^c)$, *i.e.* $w_n$ is bounded in $L\,Log^+L(\widetilde{O}\times\widetilde{O}^c)$. Then the sequence $w_n$ is relatively weakly compact in $L^1$ and modulo a subsequence we may assume that $w_n$ weakly converges in $L^1(\widetilde{O}\times\widetilde{O}^c)$ to some $w \in \mathcal{W}$. By Lemma A.2, the functions

$$\nabla_x \int_{\mathbb{R}^N} g(h)(u_n(x+h) - \hat{u}(\hat{x}+h))^2\, dh$$

and

$$\nabla_{\hat{x}} \int_{\mathbb{R}^N} g(h)(u_n(x+h) - \hat{u}(\hat{x}+h))^2\, dh$$

are bounded in $L^\infty(\widetilde{O}\times\widetilde{O}^c)$. Thus, modulo the extraction of a subsequence, we may assume that $u_n \to u$ weakly in all $L^p$, $1 \le p < +\infty$ and $g * (u_n(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$ converges strongly in all $L^p$ spaces and also in the dual of $L\,Log^+L$ to some function $W$. Then by passing to the limit as $n \to \infty$ we have

$$\int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) W(x,\hat{x}) d\hat{x}dx + T\mathcal{H}(w) \le \liminf_n \mathcal{E}_{2,T}(u_n, w_n).$$

Taking test functions $\psi(x,\hat{x})$, integrating in $\widetilde{O}\times\widetilde{O}^c$ and using the convexity of the square function, we have

$$\int_{\mathbb{R}^N} g(h)(u(x+h) - \hat{u}(\hat{x}+h))^2\, dh \le W(x,\hat{x}).$$

Thus

$$\mathcal{E}_{2,T}(u,w) \le \liminf_n \mathcal{E}_{2,T}(u_n, w_n).$$

□

*Proof of Proposition A.1.* The proof of existence follows from Lemma A.3 once we observe that there are minimizing sequences $(u_n, w_n) \in \mathcal{A}_{2,T}$ for $\mathcal{E}_{2,T}$ such that $u_n$ is uniformly bounded. For that, let $(u, w) \in \mathcal{A}_{2,T}$ and let us compute the equation satisfied by $u$ if it is a minimum of $\mathcal{E}_{2,T}(\cdot, w)$. Indeed, since

$$\int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x, \hat{x}) \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,2}^2 d\hat{x} dx = \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} g * (w(z - \cdot, \hat{z} - \cdot))(u(z) - \hat{u}(\hat{z}))^2 d\hat{z} dz, \quad \text{(A.4)}$$

we have

$$u(z) = \int_{\mathbb{R}^N} g * (w(z - \cdot, \hat{z} - \cdot))\hat{u}(\hat{z}) d\hat{z}, \quad z \in O, \quad \text{(A.5)}$$

where $g * (w(z - \cdot, \hat{z} - \cdot))$ are the pixel influence weights $m(z, \hat{z})$ defined in (3.5). We have used the fact that $\int_{O^c} g * (w(z - \cdot, \hat{z} - \cdot)) d\hat{z} = 1$ for any $z \in O$ and any $w' \in \mathcal{W}$.

Similarly, if $w$ is a minimum of $\mathcal{E}_{2,T}(u, \cdot)$ we have that

$$w(x, \hat{x}) = w_{2,T}(u)(x, \hat{x}) = \frac{1}{Z_{2,T}(u; x)} \exp\left(-\frac{1}{T} g * (u(x + \cdot) - \hat{u}(\hat{x} + \cdot))^2\right). \quad \text{(A.6)}$$

Notice that both equations (A.5) and (A.6) hold if $(u, w) \in \mathcal{A}_{2,T}$ is a minimum of $\mathcal{E}_{2,T}$.

To prove the existence of minima of of $\mathcal{E}_{2,T}$, let $(u'_n, w'_n) \in \mathcal{A}_{2,T}$ be a minimizing sequence for this energy. Let

$$u_n = \arg\min_u \mathcal{E}_{2,T}(u, w'_n), \quad \text{(A.7)}$$

$$w_n = \arg\min_w \mathcal{E}_{2,T}(u_n, w). \quad \text{(A.8)}$$

Since

$$\mathcal{E}_{2,T}(u_n, w_n) \leq \mathcal{E}_{2,T}(u_n, w'_n) \leq \mathcal{E}_{2,T}(u'_n, w'_n),$$

$(u_n, w_n) \in \mathcal{A}_{2,T}$ is also a minimizing sequence of (A.1). By (A.5) we have

$$u_n(z) = \int_{\mathbb{R}^N} g * (w'_n(z - \cdot, \hat{z} - \cdot))\hat{u}(\hat{z}) d\hat{z}, \quad z \in O, \quad \text{(A.9)}$$

and we deduce that $\|u_n\|_{L^\infty(O)} \leq \|\hat{u}\|_\infty$. Then, using Lemma A.3 we may extract a subsequence of $(u_n, w_n)$ converging to a minimum of $\mathcal{E}_{2,T}$.

Let us prove the regularity assertion. Let $(u,w) \in \mathcal{A}_{2,T}$ be a minimum of $\mathcal{E}_{2,T}$. Let us first prove that $\nabla_x w(x,\hat{x})$ and $\nabla_{\hat{x}} w(x,\hat{x})$ are bounded. By (A.5) we have that $\|u\|_{L^\infty(O)} \leq \|\hat{u}\|_\infty$ and

$$a \leq Z_{2,T}(u;x) \leq b, \tag{A.10}$$

for some constants $b > a > 0$ which only depend on $\|\hat{u}\|_\infty$. Thus $w$ is bounded and bounded away from zero. To abbreviate our expressions, let $U(x,\hat{x}) := g * (u(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$. Since

$$\nabla_x w(x,\hat{x}) = -\frac{\exp\left(-\frac{1}{T}U(x,\hat{x})\right)}{Z_{2,T}(u;x)}\left(\frac{1}{T}\nabla_x U(x,\hat{x}) + \frac{\nabla_x Z_{2,T}(u;x)}{Z_{2,T}(u;x)}\right),$$

$$\nabla_{\hat{x}} w(x,\hat{x}) = -\frac{\exp\left(-\frac{1}{T}U(x,\hat{x})\right)}{TZ_{2,T}(u;x)}\nabla_{\hat{x}} U(x,\hat{x}),$$

where

$$\nabla_x Z_{2,T}(u;x) = -\frac{1}{T}\int_{\widetilde{O}^c} \exp\left(-\frac{1}{T}U(x,\hat{x})\right)\nabla_x U(x,\hat{x})d\hat{x},$$

by Lemma A.2 and (A.10) we conclude that $\nabla_x w(x,\hat{x})$ and $\nabla_{\hat{x}} w(x,\hat{x})$ are bounded.

Now, by (A.5), for any $x \in O$ we have

$$\nabla_x u(x) \quad = \quad \int_{\mathbb{R}^N} \nabla_x g * (w(x-\cdot,\hat{z}-\cdot))\hat{u}(\hat{z})d\hat{z}.$$

Since

$$\begin{aligned}\nabla_x g * (w(x-\cdot,\hat{z}-\cdot)) = \\ \int_{\mathbb{R}^N} g(h)\nabla_x w(x+h,\hat{z}+h)\chi_{\widetilde{O}}(x+h)\chi_{\widetilde{O}^c}(\hat{z}+h)dh \\ + \int_{\mathbb{R}^N} g(h)w(x+h,\hat{z}+h)\nabla_x\chi_{\widetilde{O}}(x+h)\chi_{\widetilde{O}^c}(\hat{z}+h)dh\end{aligned}$$

and $\chi_{\widetilde{O}} \in BV(\mathbb{R}^N)$ we conclude that $\nabla_x g * (w(x-\cdot,\hat{z}-\cdot))$ is bounded. We deduce that $\nabla_x u(x)$ is bounded. Hence $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.

## Existence of minima for the patch NL-Poisson model

In this Section we consider the patch NL-Poisson model $\mathcal{E}_{2,\lambda,T}$. For simplicity we set $\lambda = 0$. Similar results hold for $\lambda \in [0,1)$. The energy reads

$$\mathcal{E}_{2,0,T}(u,w) := \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) \|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla} d\hat{x}dx + T\mathcal{H}(w), \quad \text{(A.11)}$$

where we denote $\|p\|^2_{g,\nabla} = \int_{\mathbb{R}^N} g(h)\|\nabla p(h)\|_2^2 dh$ for any $p \in W^{1,2}(\Omega_p)$. Recall that we assume that $u|_{O^c} = \hat{u}$.

Let $\mathcal{A}_{2,0,T} := \{(u,w) \in \mathcal{A}_{2,T} : u \in W^{1,2}(O),\ u|_{\partial O} = \hat{u}|_{\partial O^c}\}$.

Our purpose is to prove the following result stating the existence of solutions of

$$\min_{(u,w)\in\mathcal{A}_{2,0,T}} \mathcal{E}_{2,0,T}(u,w). \quad \text{(A.12)}$$

**Proposition A.4.** *Assume that $\hat{u} \in W^{2,2}(O^c) \cap L^\infty(O^c)$ and $g \in W^{1,\infty}(\mathbb{R}^N)^+$ has compact support in $\Omega_p$. There exists a solution of the variational problem (A.12). Moreover for any solution $(u,w) \in \mathcal{A}_{2,0,T}$ we have $u \in W^{1,2}(O) \cap W^{2,p}_{loc}(O) \cap L^\infty(O)$ for all $p \in [1,\infty)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

**Lemma A.5.** *Assume that $\hat{u} \in W^{1,2}(O^c)$ and $g \in L^\infty(\mathbb{R}^N)^+$ has compact support on $\Omega_p$. Let $(u,w) \in \mathcal{A}_{2,0,T}$. Assume that $\mathcal{E}_{2,0,T}(u,w) \le C$. Then*

$$\|u\|_{W^{1,2}(O)} \le C'(C, \|\nabla\hat{u}\|_{L^2(O^c)}), \quad \text{(A.13)}$$

*where $C' = C'(C, \|\nabla\hat{u}\|_{L^2(O^c)})$ denotes a constant that depends on its arguments.*

*Proof.* Since the left hand side of

$$\begin{aligned}\int_{\widetilde{O}} \int_{\widetilde{O}^c} & w(x,\hat{x}) g * |\nabla_x u(x+\cdot) - \nabla_{\hat{x}}\hat{u}(\hat{x}+\cdot)|^2 \, dxd\hat{x} \\ &= \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) \left( g * |\nabla_x u(x+\cdot)|^2 + g * |\nabla_{\hat{x}}\hat{u}(\hat{x}+\cdot)|^2 \right. \\ &\qquad \left. - 2g * (\nabla_x u(x+\cdot) \cdot \nabla_{\hat{x}}\hat{u}(\hat{x}+\cdot)) \right) dxd\hat{x}\end{aligned}$$

is upper bounded,

$$|g * (\nabla_x u(x+\cdot) \cdot \nabla_{\hat{x}}\hat{u}(\hat{x}+\cdot))| \le \frac{\epsilon}{2} g * |\nabla_x u(x+\cdot)|^2 + \frac{1}{2\epsilon} g * |\nabla_{\hat{x}}\hat{u}(\hat{x}+\cdot)|^2,$$

for any $\epsilon > 0$, and

$$\int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) g * |\nabla_x \hat{u}(\hat{x}+\cdot)|^2 \, dx d\hat{x} \le C'(C, \|\nabla \hat{u}\|_{L^2(O^c)}),$$

taking $\epsilon = \frac{1}{2}$, the result follows from

$$\begin{aligned}
&\int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) g * |\nabla_x u(x+\cdot)|^2 \, dx d\hat{x} \\
&\qquad = \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x}) \int_{\mathbb{R}^N} g(h) |\nabla_x u(x+h)|^2 \, dx d\hat{x} dh \\
&\qquad\quad = \int_{\widetilde{O}} \int_{\mathbb{R}^N} g(h) |\nabla_x u(x+h)|^2 \, dx dh \\
&\qquad\quad = \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \chi_{\widetilde{O}}(x) g(h) |\nabla_x u(x+h)|^2 \, dx \\
&\qquad\quad = \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \chi_{\widetilde{O}}(\hat{x}-h) g(h) |\nabla_x u(\hat{x})|^2 \, dh d\hat{x} \\
&\qquad\quad \ge \int_{O} \int_{\mathbb{R}^N} \chi_{\widetilde{O}}(\hat{x}-h) g(h) |\nabla_x u(\hat{x})|^2 \, dh d\hat{x} \\
&\qquad\quad\quad \ge \int_{O} \int_{\mathbb{R}^N} g(h) |\nabla_x u(\hat{x})|^2 \, dh d\hat{x} = \int_{O} |\nabla_x u(\hat{x})|^2 \, d\hat{x}.
\end{aligned}$$

□

**Lemma A.6.** *Assume that $\hat{u} \in W^{2,2}(O^c)$, $u \in W^{1,2}(O)$, $u|_{\partial O} = \hat{u}|_{\partial O^c}$ and $g \in W^{1,\infty}(\mathbb{R}^N)^+$ has compact support on $\Omega_p$. Then*

$$\nabla_x g * |\nabla_x u(x+\cdot) - \nabla_{\hat{x}} \hat{u}(\hat{x}+\cdot)|^2 \quad \text{and} \quad \nabla_{\hat{x}} g * |\nabla_x u(x+\cdot) - \nabla_{\hat{x}} \hat{u}(\hat{x}+\cdot)|^2 \tag{A.14}$$

*are bounded in $L^\infty(\widetilde{O} \times \widetilde{O}^c)$ with a bound depending on $\|\hat{u}\|_{W^{2,2}(O^c)}$, $\|g\|_{W^{1,\infty}}$ and $\|\nabla u\|_{L^2(O)}$.*

*Proof.* The bounds in (A.23) follow if we prove that

$$\nabla_{x+\hat{x}} \int_{\mathbb{R}^N} g(h) |\nabla_x u(x+h) - \nabla_{\hat{x}} \hat{u}(\hat{x}+h)|)^2 \, dh \quad \text{and}$$

$$\nabla_{x-\hat{x}} \int_{\mathbb{R}^N} g(h) |\nabla_x u(x+h) - \nabla_{\hat{x}} \hat{u}(\hat{x}+h)|^2 \, dh$$

are bounded in $L^\infty$ with a bound depending on $\|\hat{u}\|_{W^{2,2}(O^c)}$, $\|g\|_{W^{1,\infty}}$ and $\|\nabla u\|_{L^2(\tilde{O})}$. Let us write $u_i(x) = \partial_{x_i} u(x)$, $\hat{u}_i(x) = \partial_{x_i} \hat{u}(x)$, $i = 1, \dots, N$.

Let $x \in \widetilde{O}$, $\hat{x} \in \widetilde{O}^c$, $i \in \{1, \dots, N\}$. As in the proof of Lemma A.2 we have

$$\begin{aligned}
&\nabla_{x+\hat{x}} \int_{\mathbb{R}^N} g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))^2 \, dh \\
= \;& 2 \int_{\mathbb{R}^N} g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))(\nabla_x u_i(x+h) - \nabla_{\hat{x}} \hat{u}_i(\hat{x}+h)) \, dh \\
= \;& 2 \int_{\mathbb{R}^N} g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))(\nabla_h u_i(x+h) - \nabla_h \hat{u}_i(\hat{x}+h))^2 \, dh \\
= \;& -\int_{\mathbb{R}^N} \nabla_h g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))^2 \, dh.
\end{aligned}$$

Then

$$\begin{aligned}
&\left| \nabla_{x+\hat{x}} \int_{\mathbb{R}^N} g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))^2 \, dh \right| \\
&\qquad \leq 2\|\nabla_h g\|_\infty \left( \int_{\tilde{O}+\Omega_p} |\nabla u(\hat{x})|^2 d\hat{x} + \int_{\widetilde{O}^c+\Omega_p} |\nabla \hat{u}(\hat{x})|^2 d\hat{x} \right).
\end{aligned}$$

Notice that since $u \in W^{1,2}(O)$, $u|_{\partial O} = \hat{u}|_{\partial O^c}$ and $u = \hat{u}$ on $O^c$ we have that $u \in W^{1,2}(\tilde{O} + \Omega_p)$.

As above, by direct computation and after integration by parts, we have

$$\begin{aligned}
&\nabla_{x-\hat{x}} \int_{\mathbb{R}^N} g(h)(u_i(x+h) - \hat{u}_i(\hat{x}+h))^2 \, dh \\
&\qquad = -\int_{\mathbb{R}^N} \nabla_h g(h)(u_i(x+h)^2 - \hat{u}_i(\hat{x}+h)^2) \, dh \\
&\qquad\quad + 2 \int_{\mathbb{R}^N} g(h) u_i(x+h) \nabla_h \hat{u}_i(\hat{x}+h) \, dh \\
&\qquad\quad + 2 \int_{\mathbb{R}^N} \nabla_h (g(h)\hat{u}_i(\hat{x}+h)) u_i(x+h) \, dh.
\end{aligned}$$

The three terms are bounded by a constant depending on $\|\hat{u}\|_{W^{2,2}(O^c)}$, $\|g\|_{W^{1,\infty}}$ and $\|\nabla u\|_{L^2(\widetilde{O})}$. □

*Proof of Proposition A.4. Existence.* Let $(u_n, w_n)$ be a minimizing sequence of (A.12). Since $\Omega$ is a bounded domain we have that

$$\int_{\widetilde{O}} \int_{\widetilde{O}^c} \chi_{\{w_n > 1\}} w_n(x, \hat{x}) \log w_n(x, \hat{x}) d\hat{x} dx$$

is bounded. Hence $w_n(1+\log^+ w_n)$ is bounded in $L^1(\widetilde{O}\times\widetilde{O}^c)$. Then the sequence $w_n$ is relatively weakly compact in $L^1$ and modulo a subsequence we may assume that $w_n$ weakly converges in $L^1(\widetilde{O}\times\widetilde{O}^c)$ to some $w\in\mathcal{W}$.

By Lemma A.5, we have that $u_n$ is uniformly bounded in $W^{1,2}(\widetilde{O})$. By Lemma A.6, we have that

$$\nabla_x \int_{\mathbb{R}^N} g(h)|\nabla u_n(x+h)-\nabla\hat{u}_n(\hat{x}+h)|^2\,dh \quad \text{and}$$
$$\nabla_{\hat{x}} \int_{\mathbb{R}^N} g(h)|\nabla u_n(x+h)-\nabla\hat{u}_n(\hat{x}+h)|^2\,dh \quad \text{(A.15)}$$

are uniformly bounded in $L^\infty(\widetilde{O}\times\widetilde{O}^c)$. Thus, modulo the extraction of a subsequence, we may assume that $u_n\to u$ a.e. and in $L^2(\widetilde{O})$, $\nabla u_n\to\nabla u$ weakly in $L^2(\widetilde{O}+\Omega_p)$, and $g*(\nabla_x u_n(x+\cdot)-\nabla_{\hat{x}}\hat{u}_n(\hat{x}+\cdot))^2$ converges strongly in all $L^p$ spaces, $1\le p<\infty$, and also in the dual of $L\,Log^+L$ to some function $W$. Then by passing to the limit as $n\to\infty$ we have

$$\int_{\widetilde{O}}\int_{\widetilde{O}^c} w(x,\hat{x})W(x,\hat{x})d\hat{x}dx + T\mathcal{H}(w) \le \liminf_n \mathcal{E}_{2,0,T}(u_n,w_n).$$

Taking test functions $\psi(x,\hat{x})$, integrating in $\widetilde{O}\times\widetilde{O}^c$ and using the convexity of the square function, we have

$$\int g(h)(\nabla_x u(x+h)-\nabla_{\hat{x}}\hat{u}(\hat{x}+h))^2\,dh \le W(x,\hat{x}).$$

Thus

$$\mathcal{E}_{2,0,T}(u,w) \le \liminf_n \mathcal{E}_{2,0,T}(u_n,w_n).$$

*Regularity.* Observe that if $(u,w)\in\mathcal{A}_{2,0,T}$ is a minimum of (A.12), then $u\in W^{1,2}(O)$ (by Lemma A.5) and it satisfies the Euler-Lagrange equations. By fixing $w$ and computing the first variation of $\mathcal{E}_{2,0,T}$ with respect to its first variable, we have that $u$ is a solution of the Poisson equation:

$$\begin{aligned} &\Delta u(z) = \operatorname{div} v(w;z), && z\in O, \\ &u=\hat{u} && \text{in } \partial O, \end{aligned} \quad \text{(A.16)}$$

where

$$v(w;z) = \int_{\mathbb{R}^N} g*(w(z-\cdot,\hat{z}-\cdot))\nabla\hat{u}(\hat{z})d\hat{z}. \quad \text{(A.17)}$$

Notice that we used again that $k(w;z) := \int_{\mathbb{R}^N} g * (w(z-\cdot, \hat{z}-\cdot))d\hat{z} = 1$ for all $z \in O$. On the other hand, if $u$ is fixed and we compute the first variation of $\mathcal{E}_{2,0,T}$ with respect to its second variable, then

$$w(x,\hat{x}) = w_{2,0,T}(u;x,\hat{x}) = \frac{1}{Z_{2,0,T}(u;x)} \exp\left(-\frac{1}{T}\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}\right), \tag{A.18}$$

where the normalizing factor $Z_{2,0,T}(u)$ is given by

$$Z_{2,0,T}(u;x) = \int_{\widetilde{O}^c} \exp\left(-\frac{1}{T}\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}\right) d\hat{x}. \tag{A.19}$$

As in the proof of Proposition A.1 we observe that $Z_{2,0,T}(u;x)$ is bounded and bounded away from zero (thanks to Lemmas A.5 and A.6). Then, it follows that $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$. Using this and (A.17) we have that $\boldsymbol{v}(w) \in W^{1,\infty}(O)^2$. Then the solution $u$ of (A.16) is in $W^{1,2}(O) \cap W^{2,p}_{\text{loc}}(O) \cap L^\infty(O)$ for any $p \in [1,\infty)$ (Gilbarg and Trudinger (2001)).

**Remark 3.** *Notice that the regularity result holds for any* $(u^*, w^*) \in \mathcal{A}_{2,0,T}$ *satisfying* (A.16) *and* (A.18).

## Existence of minima for the patch NL-medians and patch NL-gradient medians models

We consider the patch NL-medians model

$$\mathcal{E}_{1,T}(u,w) = \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x,\hat{x})\|p_u(x) - p_{\hat{u}}(\hat{x})\|_g d\hat{x}dx + T\mathcal{H}(w). \tag{A.20}$$

As previously, we assume $\mathcal{E}_{2,T}(u,w) = +\infty$ in case that the second integral is not defined.

Let

$$\mathcal{W} := \{w \in L^1(\widetilde{O} \times \widetilde{O}^c) : \int_{\widetilde{O}^c} w(x,\hat{x})\, d\hat{x} = 1 \text{ a.e.} x \in \widetilde{O}\}.$$

Let us consider the admissible class of functions

$$\mathcal{A}_{1,T} := \{(u,w) : u \in L^\infty(\Omega),\ u = \hat{u} \text{ in } O^c, w \in \mathcal{W}\}.$$

Our purpose is to prove the following result.

**Proposition A.7.** *Assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. There exists a minimum $(u, w) \in \mathcal{A}_{1,T}$ of $\mathcal{E}_{1,T}$. For any minimum $(u, w) \in \mathcal{A}_{1,T}$ we have that $u \in W^{1,\infty}(O)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.*

To prove Proposition A.7 we need the following Lemma.

**Lemma A.8.** *Assume that $g \in C_c(\mathbb{R}^N)^+$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. Assume that $u \in L^\infty(\widetilde{O} + \Omega_p)$. Then the functions*

$$\nabla_x g * |u(x+\cdot) - \hat{u}(\hat{x}+\cdot)| \quad \text{and} \quad \nabla_{\hat{x}} g * |u(x+\cdot) - \hat{u}(\hat{x}+\cdot)| \tag{A.21}$$

*are uniformly bounded in $\widetilde{O} \times \widetilde{O}^c$ by a constant that depends on $\|\nabla g\|_{L^1}$, $\|u\|_\infty$, $\|\hat{u}\|_\infty$.*

*Proof.* Let $x \in \widetilde{O}$, $\hat{x} \in \widetilde{O}^c$. Observe that $\hat{x} + \Omega_p \subset O^c$. Let us observe that by approximating $u$ by smooth functions the following computations are easily justified. Let $\eta(r) = \operatorname{sign}_0(r)$. Using the $L^1$ integrability of $\nabla_h g$ and the boundedness of $u$, we compute

$$\begin{aligned}
& \nabla_{x+\hat{x}} \int_{\mathbb{R}^N} g(h)|u(x+h) - \hat{u}(\hat{x}+h)|\,dh \\
= & \int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_x u(x+h) - \nabla_{\hat{x}}\hat{u}(\hat{x}+h))\,dh \\
= & \int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_h u(x+h) - \nabla_h \hat{u}(\hat{x}+h))\,dh \\
= & -\int_{\mathbb{R}^N} \nabla_h g(h)|u(x+h) - \hat{u}(\hat{x}+h)|\,dh.
\end{aligned}$$

By our assumptions, the above integral is bounded by a constant that only

depends on $\|\nabla g\|_{L^1}$, $\|u\|_\infty$ and $\|\hat{u}\|_\infty$. Now,

$$\begin{aligned}
&\nabla_{x-\hat{x}} \int_{\mathbb{R}^N} g(h)|u(x+h) - \hat{u}(\hat{x}+h)|\, dh \\
&\quad = \int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_x u(x+h) + \nabla_{\hat{x}}\hat{u}(\hat{x}+h))\, dh \\
&\quad = \int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h))(\nabla_h u(x+h) + \nabla_h \hat{u}(\hat{x}+h))\, dh \\
&\qquad = \int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h)) \\
&\qquad\qquad (\nabla_h u(x+h) - \nabla_h \hat{u}(\hat{x}+h) + 2\nabla_h \hat{u}(\hat{x}+h))\, dh \\
&\qquad = \int_{\mathbb{R}^N} g(h)\nabla_h |u(x+h) - \hat{u}(\hat{x}+h)|\, dh \\
&\qquad\qquad + 2\int_{\mathbb{R}^N} g(h)\eta(u(x+h) - \hat{u}(\hat{x}+h))\nabla_h \hat{u}(\hat{x}+h)\, dh.
\end{aligned}$$

Integrating by parts, we have that the first term in the last equality is bounded. The last integral is also bounded since $\hat{u} \in BV(O^c)$. The necessary computations can be justified by smoothing the absolute value ($\sqrt{\epsilon^2+^2}$) and smoothing the functions. □

**Remark 4.** *There are minimizing sequences that satisfy the assumptions of the Lemma A.2.*

**Remark 5.** *The above proof can be extended to any patch error function given by*

$$g * |u(x+\cdot) - \hat{u}(\hat{x}+\cdot)|^p$$

*for an exponent* $p \in [1, \infty)$.

We consider now the patch NL-gradient medians energy, with $\lambda = 0$:

$$\mathcal{E}_{2,0,T}(u, w) := \int_{\widetilde{O}} \int_{\widetilde{O}^c} w(x, \hat{x}) \|p_u(x) - p_{\hat{u}}(\hat{x})\|_{g,\nabla} d\hat{x} dx + T\mathcal{H}(w), \quad \text{(A.22)}$$

In this case we only sketch a proof for existence, without stating the regularity of the solution.

Let us define the following notation:

$$HBV(\Omega) = \{u \in W^{1,1}(\Omega), \nabla u \in BV(\Omega)\}.$$

The following Lemma is analogous to Lemmas A.2, A.6 and A.8 for the case of patch NL-gradient medians.

**Lemma A.9.** *Assume that $\hat{u} \in HBV(O^c)$, $u \in BV(O)$, $u = \hat{u}$ in $O^c$ and $g \in W^{1,\infty}(\mathbb{R}^N)^+$ has compact support on $\Omega_p$. Then*

$$\nabla_x \int_{\mathbb{R}^N} g(h)|\nabla_x u(x+h) - \nabla_{\hat{x}}\hat{u}(\hat{x}+h)|\,dh \quad \text{and}$$
$$\nabla_{\hat{x}} \int_{\mathbb{R}^N} g(h)|\nabla_x u(x+h) - \nabla_{\hat{x}}\hat{u}(\hat{x}+h)|\,dh \quad \text{(A.23)}$$

*are bounded in $L^\infty(\widetilde{O}\times\widetilde{O}^c)$ with a bound depending on $\|\hat{u}\|_{HBV(O^c)}$, $\|g\|_{W^{1,\infty}}$ and $\|u\|_{BV}$.*

Then from the boundedness of the energy one gets that the assumptions are satisfied. As before, from Lemma A.9 one can deduce the existence of minima.

Appendix B

# Optimal correspondence maps

In this appendix we prove existence of solutions of the energy $\mathcal{E}_{2,0}$ and the existence of optimal correspondence maps. Then in Subsection B we prove the existence of correspondence maps which are uniform limits of bounded variation functions with finitely many values. We give the details corresponding to the patch NL-means model. Analogous results with similar proofs hold for the patch NL-Poisson model (see Remark 7).

### Existence of optimal correspondence maps

Let us first recall the notion of measurable measure-valued map.

**Definition B.1** (Measurable measure-valued map)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^N$, $\mathcal{Y} \subseteq \mathbb{R}^M$ be open sets, $\mu$ be a positive Radon measure in $\mathcal{X}$ and $x \to \nu_x$ be a function that assigns to each $x$ in $\mathcal{X}$ a Radon measure $\nu_x$ on $\mathcal{Y}$. We say that the map is $\mu$-measurable if $x \to \nu_x(B)$ is $\mu$-measurable for any Borel set $B$ in $\mathcal{Y}$.*

By the disintegration theorem, if $\nu$ is a positive Radon measure in $\mathcal{X} \times \mathcal{Y}$ such that $\nu(K \times \mathcal{Y}) < \infty$ for any compact set $K \subseteq \mathcal{X}$ and $\mu = \pi_\sharp \nu$ where $\pi : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ is the projection on the first factor (*i.e.* $\mu(B) = \nu(B \times Y)$ for any Borel set $B \subseteq X$), then there exist a measurable measure-valued map $x \to \nu_x$ such that $\nu_x(\mathcal{Y}) = 1$ $\mu$-a.e. in $\mathcal{X}$ and for any $\psi \in L^1(\mathcal{X} \times \mathcal{Y}, \nu)$

we have

$$\psi(x,\cdot) \in L^1(\mathcal{Y},\nu_x) \qquad \text{for } \mu - \text{a.e.} x \in \mathcal{X},$$
$$x \to \int_{\mathcal{Y}} \psi(x,y)\, d\nu_x(y) \in L^1(\mathcal{X},\mu),$$
$$\int_{\mathcal{X}\times\mathcal{Y}} \psi(x,y)\, d\nu(x.y) = \int_{\mathcal{X}}\int_{\mathcal{Y}} \psi(x,y)\, d\nu_x(y)\, d\mu(x).$$

Let us consider $\mathcal{MP}$ the set of measurable measure valued maps $\nu \geq 0$ in $\widetilde{O}\times \mathrm{cl}(\widetilde{O}^c)$ such that $\pi_\sharp \nu = \mathcal{L}^N|_{\widetilde{O}}$, where $\mathcal{L}^N|_{\widetilde{O}}$ denotes the Lebesgue measure restricted to $\widetilde{O}$. We assume that $g \in C_c(\mathbb{R}^N)$ has support contained in $\Omega_p$, $\nabla g \in L^1(\mathbb{R}^N)$ and $\hat{u} \in BV(O^c) \cap L^\infty(O^c)$. Let

$$\mathcal{A}_{2,0} := \{(u,\nu) : u \in L^\infty(\Omega), u = \hat{u} \text{ in } O^c, \nu \in \mathcal{MP}\}.$$

For $(u,\nu) \in \mathcal{A}_{2,0}$, define

$$\mathcal{E}_{2,0}(u,\nu) := \int_{\widetilde{O}}\int_{\widetilde{O}^c} g * (u(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2 d\nu(x,\hat{x}). \tag{B.1}$$

Notice that, by Lemma A.2, the above integral is well defined. We are now ready to state the existence result.

**Proposition B.2.** *There exists a minimum $(u,\nu) \in \mathcal{A}_{2,0}$ of $\mathcal{E}_{2,0}$.*

*Proof.* Let $(u_n,\nu_n) \in \mathcal{A}_{2,0}$ be a minimizing sequence of $\mathcal{E}_{2,0}$. Let us observe that we may take $u_n$ uniformly bounded. Indeed, let $\kappa = 2\|\hat{u}\|_\infty$ and $u_{\kappa,n} = u_n \chi_{|u_n|\leq\kappa}$. Then

$$|u_{\kappa,n}(x) - \hat{u}(\hat{x})| \leq |u_n(x) - \hat{u}(\hat{x})|.$$

This is clearly true if $|u_n(x)| \leq \kappa$. Now if $|u_n(x)| > \kappa$ we have

$$|u_n(x) - \hat{u}(\hat{x})| > \kappa - \hat{u}(\hat{x}) \geq \|\hat{u}\|_\infty \geq |u_{\kappa,n}(x) - \hat{u}(\hat{x})|.$$

Thus $g * (u_{\kappa,n}(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2 \leq g * (u_n(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$ and $(u_{\kappa,n},\nu_n)$ is also a minimizing sequence. Thus, we may assume that $u_n$ is uniformly bounded. By extracting a subsequence, if necessary, we may assume that $u_n$ converges to some $u \in L^\infty(\widetilde{O}+\Omega_p)$ with $u|_{O^c} = \hat{u}$. By Lemma A.2, $g * (u_n(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$ is uniformly Lipschitz in $x,\hat{x}$ and we may assume

that it converges uniformly to a function $W(x,\hat{x})$. On the other hand, we may also assume that $\nu_n \to \nu$ weakly$^*$ as measures. Hence

$$\int_{\widetilde{O}}\int_{\widetilde{O}^c} W(x,\hat{x})d\nu(x,\hat{x}) \leqslant \liminf_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} g*(u_n(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 d\nu(x,\hat{x}).$$

Notice that $\pi_\sharp\nu = \mathcal{L}^N|_{\widetilde{O}}$. Clearly from the weak$^*$ convergence, we have that $\nu(V\times \mathrm{cl}(\widetilde{O}^c)) \leq \mathcal{L}^N(V)$ for any open set $V \subseteq \widetilde{O}$ and $\nu(K\times \mathrm{cl}(\widetilde{O}^c)) \geq \mathcal{L}^N(K)$ for any compact set $K \subseteq \widetilde{O}$. By regularity of the measure $\nu$ we have that $\nu(B\times \mathrm{cl}(\widetilde{O}^c)) = |B|$ for any Borel set $B \subseteq \widetilde{O}$. Thus $\pi_\sharp\nu = \mathcal{L}^N|_{\widetilde{O}}$.

Now, as in the proof of Proposition A.1, we have that $g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 \leq W(x,\hat{x})$. Hence

$$\int_{\widetilde{O}}\int_{\widetilde{O}^c} g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 d\nu(x,\hat{x}) \leq \liminf_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} g*(u_n(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 d\nu(x,\hat{x}).$$

Thus $(u,\nu) \in \mathcal{A}_{2,0}$ is a minimum of $\mathcal{E}_{2,0}$. □

**Remark 6.** *If $\hat{u} \geq 0$ we may also take $u_{\kappa,n} = \tau_\kappa(u_n)$ where $\kappa \geq \|\hat{u}\|_\infty$ and $\tau_\kappa(r) = \mathrm{sign}(r)\inf(|r|,\kappa)$.*

Let $\varphi : \widetilde{O} \to \widetilde{O}^c$ be a measurable map. Then $x \in \widetilde{O} \to \nu_x = \delta_{\varphi(x)}(\hat{x})$ is measurable. Similarly if the map $x \in \widetilde{O} \to \nu_x = \delta_{\varphi(x)}(\hat{x})$ is measurable then $\varphi$ is measurable. Let us denote by $\nu^\varphi$ the measure determined by $\varphi$.

**Proposition B.3.** *There exists a minimum $(u^*,\nu^*) \in \mathcal{A}_{2,0}$ of $\mathcal{E}_{2,0}$ such that $\nu^* = \nu^\varphi$ where $\varphi : \widetilde{O} \to \mathrm{cl}(\widetilde{O}^c)$ is a measurable map.*

*Proof.* Let $(u^*,\nu^*) \in \mathcal{A}_{2,0}$ be a minimum of $\mathcal{E}_{2,0}$. Then

$$\nu^* \in \operatorname*{arg\,min}_{\nu\in\mathcal{MP}} \mathcal{E}_{2,0}(u^*,\nu).$$

Let us prove that there is a measurable map $\varphi$ such that $\nu^\varphi$ is a minimum of $\mathcal{A}_{2,0}$. Indeed, this is a consequence of the Kuratowski-Ryll-Nardzewski Theorem (Srivastava (1998), Theorem 5.2.1) or (Aliprantis and Border (2006),

Theorem 14.86). Let us consider $U(x,\hat{x}) = g * (u(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$, $x \in \widetilde{O}$, $\hat{x} \in \widetilde{O}^c$. Let

$$m(x) = \min_{\hat{x}\in\widetilde{O}^c} U(x,\hat{x}),$$
$$M(x) = \{\hat{x} \in \widetilde{O}^c : U(x,\hat{x}) = m(x)\}.$$

Then, by Berge's maximum theorem applied to $-U(x,\hat{x})$ (Aliprantis and Border (2006), Theorem 14.30) we have that $m(x)$ is continuous and $M(x)$ is an upper hemicontinuous correspondence with compact values. Thus it has a closed graph (Aliprantis and Border (2006), Theorem 14.12), hence it is measurable (Aliprantis and Border (2006), Theorem 14.68). By the Kuratowski-Ryll-Nardzewski Theorem, we know that $x \to M(x)$ admits a measurable selector, that is there is a measurable map $x \to \varphi(x) \in M(x)$. The measure $\nu^\varphi$ is a minimum of $\mathcal{A}_{2,0}$ since we may write

$$U(x,\varphi(x)) = \min_{\hat{x}\in\widetilde{O}^c} U(x,\hat{x}) \leq \int_{\widetilde{O}^c} U(x,\hat{x}) d\nu_x^*(\hat{x}).$$

Integrating in $x$ we deduce that

$$\mathcal{E}_{2,0}(u^*,\nu^\varphi) = \int_{\widetilde{O}}\int_{\widetilde{O}^c} U(x,\hat{x}) d\nu_x^\varphi(\hat{x})\mathrm{d}x \leq \int_{\widetilde{O}}\int_{\widetilde{O}^c} U(x,\hat{x}) d\nu_x^*(\hat{x})\mathrm{d}x = \mathcal{E}_{2,0}(u^*,\nu^*).$$

□

Let us sketch a second proof which gives a different point of view. Clearly, since the energy function $\nu \to \mathcal{E}_{2,0}(u^*,\nu)$ is linear in $\nu$, there are minima that are attained on the set of extreme points of the convex set $\mathcal{MP}$. Thus, the proposition is a consequence of the following Lemma whose proof can be found in (Arias et al. (2012)).

**Lemma B.4.** *The set of extreme points of the convex set $\mathcal{MP}$ coincides with the set of measures $\{\nu^\varphi : \varphi$ is a measurable map$\}$.*

We now address the relation between the patch NL-means functional for $T > 0$ and $\mathcal{E}_{2,0}$.

**Proposition B.5.** *The energies $\mathcal{E}_{2,T}$ Gamma-converge to the energy $\mathcal{E}_{2,0}$. In particular, the minima of $\mathcal{E}_{2,T}$ converge to minima of $\mathcal{E}_{2,0}$.*

*Proof.* Let $(u,\nu) \in \mathcal{A}_{2,0}$ and $(u_n, w_n) \in \mathcal{A}_2$ be such that $u_n \to u$ weakly$^*$ in $L^\infty$ and $w_n \to \nu$ weakly$^*$ as measures. The fact that $u_n \to u$ weakly$^*$ in $L^\infty$ includes that $u_n$ is uniformly bounded. By Lemma A.2, $g*(u_n(x+\cdot) - \hat{u}(\hat{x}+\cdot))^2$ is uniformly Lipschitz and converges uniformly to some function $W(x,\hat{x})$. Then

$$\int_{\widetilde{O}}\int_{\widetilde{O}^c} W(x,\hat{x})d\nu(x,\hat{x}) \leq \liminf_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} w_n(x,\hat{x}) g*(u_n(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 dxd\hat{x}$$

$$\leq \liminf_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} w_n(x,\hat{x}) g*(u_n(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 dxd\hat{x} + T_n\mathcal{H}(w_n).$$

Since $g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2 \leq W(x,\hat{x})$, we have that

$$\mathcal{E}_{2,0}(u,\nu) \leq \liminf_n \mathcal{E}_{2,T_n}(u_n,w_n).$$

Now, let $(u,\nu) \in \mathcal{A}_{2,0}$. Let $\nu_x$ be the probability measures obtained by disintegrating $\nu$ with respect to $\mathcal{L}^N|_{\widetilde{O}}$. Let us take $w_n(x,\hat{x}) = g_{T_n} * \nu_x(\hat{x})$ where $g_T(\hat{x}) = \frac{1}{T^2} g(\frac{\hat{x}}{T})$ for any $T>0$. Thus $w_n \in \mathcal{W}$, $w_n \to \nu$ weakly$^*$ as measures and $w_n \leq \frac{1}{T_n^N}\|g\|_\infty$. This implies that $(u,w_n) \in \mathcal{A}_2$ and

$$\lim_n T_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} w_n(x,\hat{x}) \log\, w_n(x,\hat{x})\, dxd\hat{x} = 0.$$

On the other hand

$$\lim_n \int_{\widetilde{O}}\int_{\widetilde{O}^c} w_n(x,\hat{x}) g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2\, dxd\hat{x} = \int_{\widetilde{O}}\int_{\widetilde{O}^c} g*(u(x+\cdot)-\hat{u}(\hat{x}+\cdot))^2\, d\nu(x,\hat{x}).$$

Thus $\limsup_n \mathcal{E}_{2,T_n}(u,w_n) \leq \mathcal{E}_{2,0}(u,\nu)$. □

**Remark 7.** *Statements analogous to Propositions B.2, B.3 and B.5 also hold for the non-local Poisson model with $\lambda = 0$, where the limit energy is now*

$$\mathcal{E}_{2,0,0}(u,\nu) := \int_{\widetilde{O}}\int_{\widetilde{O}^c} g*(\nabla u(x+\cdot) - \nabla\hat{u}(\hat{x}+\cdot))^2\, d\nu(x,\hat{x}).$$

*In this case, we assume that $\hat{u} \in W^{2,2}(O^c)$ and $g \in W^{1,\infty}(\mathbb{R}^N)$ has compact support in $\Omega_p$ and we use Lemmas A.5 and A.6.*

## Regularity of optimal correspondence maps

The following result gives us some information on the the existence of correspondences with some regularity.

**Theorem B.6.** *Let $X$ be an open bounded subset of $\mathbb{R}^N$ with Lipschitz boundary and $Y$ be a compact subset of $\mathbb{R}^m$. Let $U : X \times Y \to \mathbb{R}$ be a Lipschitz continuous function. For each $x \in X$, let $M(x) := \{\hat{x} \in Y : U(x,\hat{x}) = \min_{\bar{x}\in Y} U(x,\bar{x})\}$. Then there exists a selection of the multifunction $x \in X \to M(x) \subseteq Y$, i.e., a function $S : X \to Y$ such that $S(x) \in M(x)\ \forall x \in X$, which is a uniform limit of functions in $BV(X)^m$.*

We included a brief summary of the properties of $BV$ functions after the proof of this theorem. Notice that we defined $X$ as an open set since we define the $BV$ space on open sets. The same statement holds true if we replace $X$ by its closure $\overline{X}$. In that case, since $X$ has a Lipschitz boundary (hence of Lebesgue null measure) functions in $BV(X)$ uniquely determine its extension to $\overline{X}$ (the trace on $\partial X$ is well defined) Ambrosio et al. (2000).

Its proof is a simple adaptation of the Kuratowski-Ryll-Nardzewski Theorem (Srivastava (1998), Theorem 5.2.1) or (Aliprantis and Border (2006), Theorem 14.86). For that we need the following simple Lemma.

**Lemma B.7.** *Let $m(x) = \inf_{\hat{x}\in Y} U(x,\hat{x})$, $x \in X$. The function $m(x)$ is Lipschitz. Also are the functions $x \in X \to \inf_{\bar{x}\in B} U(x,\bar{x})$ for any $B \subseteq Y$ and $\epsilon > 0$.*

*Proof.* Let $x, \bar{x} \in X$. Since $U$ is Lipschitz in both variables we have

$$m(x) \leq U(x,\hat{x}) \leq U(\bar{x},\hat{x}) \leq L|x-\bar{x}|.$$

Taking infima with respect to $\hat{x}$ in the right hand side we have $m(x) \leq m(\bar{x}) + L|x-\bar{x}|$. By symmetry we have that $m$ is Lipschitz. The more general case follows in the same way. □

*Proof of Theorem B.6.* Let

$$M_\epsilon(x) = \{\hat{x} \in Y : U(x,\hat{x}) \leq \min_{\bar{x}\in Y} U(x,\bar{x}) + \epsilon\}, \quad x \in X.$$

Let $\epsilon_n \downarrow 0$ and $M_n(x) := M_{\epsilon_n}(x)$. Let $d$ denote the euclidian distance in $Y$. Without loss of generality, assume that $d < 1$.

We define inductively a sequence $S_n : X \to Y$ of functions in $BV(X)^m$ taking finitely many values such that for every $x \in X$ and every $n \in \mathbb{N}$,

(i) $d(S_n(x), M(x)) \le 2^{-n}$, and

(ii) $d(S_{n-1}(x), S_n(x)) \le 2^{-n+2}$.

For that, let $\{r_k\}$ be a countable dense set in $Y$. Define $S_0(x) = r_0$ for all $x \in X$. Let $n > 0$ and assume that we have constructed $S_i$ satisfying $(i)$, $(ii)$ for all $i < n$. For each $k \in \mathbb{N}$, let

$$E_k^n = \{x \in X : |S_{n-1}(x) - r_k| \le 2^{-n+2},\ d(r_k, M_n(x)) \le 2^{-n}\}.$$

Let us prove that for a suitable selection of $\epsilon_n$ the sets $E_k^n$ are sets of finite perimeter in $X$. Since $S_{n-1}$ is in $BV(X)^m$ and takes finitely many values, the set $\{x \in X : |S_{n-1}(x) - r_k| \le 2^{-n+2}\}$ is of finite perimeter in $X$. Observe

$$\begin{aligned}
&\{x \in X : d(r_k, M_n(x)) \le 2^{-n}\} \\
&= \{x \in X : \exists \hat{x} \in B(r_k, 2^{-n}) \text{ s. t. } U(x, \hat{x}) \le m(x) + \epsilon_n\} \\
&= \{x \in X : \inf_{\hat{x} \in B(r_k, 2^{-n})} U(x, \hat{x}) - m(x) \le \epsilon_n\}.
\end{aligned}$$

Since by Lemma B.7 the functions $\inf_{\hat{x} \in B(r_k, 2^{-n})} U(x, \hat{x})$ and $m(x)$ are Lipschitz functions of $x$, by excluding a set of null measure we may select a sequence $\epsilon_n \to 0+$ such that all sets $E_k^n$ are sets of finite perimeter in $X$.

Let us consider a finite set $R_n \subset \{r_k\}_k$, which is a $2^{-n}$ net in $Y$, that is, any point $\hat{x} \in Y$ is at distance at most $2^{-n}$ from $R_n$. Let us prove that $X = \cup_{k : r_k \in R_n} E_k^n$. Let $x \in X$. Since, by $(i)$, $d(S_{n-1}(x), M(x)) \le 2^{-n+1}$, there exists $\hat{x} \in M(x)$ such that $|S_{n-1}(x) - \hat{x}| \le 2^{-n+1}$. Let $r_k \in R_n$ be such that

$$|\hat{x} - r_k| \le 2^{-n}. \tag{B.2}$$

Then

$$|S_{n-1}(x) - r_k| \le |S_{n-1}(x) - \hat{x}| + |\hat{x} - r_k| \le 2^{-n+1} + 2^{-n} < 2^{-n+2}. \tag{B.3}$$

Thus $x \in E_k^n$.

Then there exist pairwise disjoint sets $D_k^n \subseteq E_k^n$ such that $\cup_{k : r_k \in R_n} D_k^n = \cup_{k : r_k \in R_n} E_k^n = X$ and $D_k^n$ are sets of finite perimeter in $X$. It suffices to take $D_k^n = E_k^n \setminus \cup_{j<k} E_j^n$.

Let $S_n(x) = \sum_{k:r_k \in R_n} r_k \chi_{D_k^n}(x)$. Clearly $S_n \in BV(X)^m$ and takes finitely many values in $Y$. By (B.2), we have that $d(S_n(x), M(x)) \leq 2^{-n}$. By (B.3), we have $d(S_{n-1}(x), S_n(x)) \leq 2^{-n+2}$.

Now, by $(ii)$, the sequence $S_n$ converges uniformly to some function $S : X \to Y$. By $(i)$, $S(x) \in M(x)$ for all $x \in X$. Thus $S$ is a uniform limit of functions in $BV(X)^m$.

**Remark 8.** *Notice that the above proof also shows that given $\epsilon > 0$ there is a function $S_\epsilon \in BV(X)^m$ with finitely many values such that $U(x, S_\epsilon(x)) \leq m(x) + \epsilon$.*

## Some properties of functions with bounded variation

Let us point out some of the properties of $S$ as a uniform limit of $BV(X)^m$ functions. For a detailed account of the properties of functions of bounded variation, we refer to Ambrosio et al. (2000).

Let $Q$ be an open subset of $\mathbb{R}^N$. Let $u \in L^1_{\text{loc}}(Q)$. The total variation of $u$ in $Q$ is defined by

$$V(u, Q) := \sup \left\{ \int_Q u \ \text{div}\, \sigma \ dx : \sigma \in C_c^\infty(Q; \mathbb{R}^N), |\sigma(x)| \leq 1 \ \forall x \in Q \right\}, \tag{B.4}$$

where $C_c^\infty(Q; \mathbb{R}^N)$ denotes the vector fields with values in $\mathbb{R}^N$ which are infinitely differentiable and have compact support in $Q$. For a vector $v = (v_1, \ldots, v_N) \in \mathbb{R}^N$ we denoted $|v|^2 := \sum_{i=1}^N v_i^2$. Following the usual notation, we will denote $V(u, Q)$ by $|Du|(Q)$.

Let $u \in L^1(Q)$. We say that $u$ is a *function of bounded variation* in $Q$ if $V(u, Q) < \infty$. The vector space of functions of bounded variation in $Q$ will be denoted by $BV(Q)$. Recall that $BV(Q)$ is a Banach space when endowed with the norm $\|u\| := \int_Q |u| \ dx + |Du|(Q)$.

A measurable set $E \subseteq Q$ is said to be of *finite perimeter* in $Q$ if $\chi_E \in BV(Q)$. The perimeter of $E$ in $Q$ is defined as $P(E, Q) := |D\chi_E|(Q)$. Recall that almost all level sets of a bounded variation function are sets of finite perimeter.

Let us denote by $\mathcal{L}^N$ and $\mathcal{H}^{N-1}$, respectively, the $N$-dimensional Lebesgue measure and the $(N-1)$-dimensional Hausdorff measure in $\mathbb{R}^N$.

Let $u \in [L^1_{\mathrm{loc}}(Q)]^m$ $(m \geq 1)$. We say that $u$ has an approximate limit at $x \in Q$ if there exists $\xi \in \mathbb{R}^m$ such that

$$\lim_{\rho \downarrow 0} \frac{1}{|B(x,\rho)|} \int_{B(x,\rho)} |u(y) - \xi| dy = 0. \tag{B.5}$$

The set of points where this does not hold is called the approximate discontinuity set of $u$, and is denoted by $S_u$. Using Lebesgue's differentiation theorem, one can show that the approximate limit $\xi$ exists at $\mathcal{L}^N$-a.e. $x \in Q$, and is equal to $u(x)$: in particular, $|S_u| = 0$. If $x \in Q \setminus S_u$, the vector $\xi$ is uniquely determined by (B.5) and we denote it by $\tilde{u}(x)$.

We say that $x \in Q$ is an *approximate jump point of* $u$ if there exist $u^+(x) \neq u^-(x) \in \mathbb{R}^m$ and $|\nu_u(x)| = 1$ such that

$$\lim_{\rho \downarrow 0} \frac{1}{|B^{\pm}_{\rho}(x, \nu_u(x))|} \int_{B^{\pm}_{\rho}(x,\nu_u(x))} |u(y) - u^{\pm}(x)| \, dy = 0$$

where $B^{\pm}_{\rho}, \nu_u(x)) = \{y \in B(x,\rho) \ : \ \pm\langle y - x, \nu_u(x)\rangle > 0\}$. We denote by $J_u$ the set of approximate jump points of $u$. If $u \in BV(Q)^m$, the set $S_u$ is countably $\mathcal{H}^{N-1}$ rectifiable, $J_u$ is a Borel subset of $S_u$ and $\mathcal{H}^{N-1}(S_u \backslash J_u) = 0$ Ambrosio et al. (2000). In particular, we have that $\mathcal{H}^{N-1}$-a.e. $x \in Q$ is either a point of approximate continuity of $\tilde{u}$, or a jump point with two limits in the above sense.

Appendix C

# Convergence of the alternating optimization scheme

In this chapter we prove the convergence of the alternating optimization scheme (Algorithm 2) such a scheme to a critical point of the energy. We consider the models of patch NL-means and patch NL-Poisson models with $\lambda = 0$.

Let $\mathcal{E}_{E,T}$ be one of the energies $\mathcal{E}_{2,T}$ or $\mathcal{E}_{2,0,T}$. Similarly, $\mathcal{A}_{E,T}$ denotes $\mathcal{A}_{2,T}$ or $\mathcal{A}_{2,0,T}$.

**Proposition C.1.** *The iterated optimization algorithm converges (modulo a subsequence) to a critical point $(u^*, w^*) \in \mathcal{A}_{E,T}$ of $\mathcal{E}_{E,T}$. For the energy $\mathcal{E}_{2,T}$ (resp. $\mathcal{E}_{2,0,T}$) the solution obtained has the smoothness described in Proposition A.1 (resp. A.4), that is $u^* \in W^{1,\infty}(O)$ and $w^* \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$ (resp. $u^* \in W^{1,2}(O) \cap W^{2,p}_{loc}(O) \cap L^\infty(O)$ for any $p \in [1, \infty)$ and $w \in W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$).*

*Proof.* Being similar, we give the details only for the case of the patch NL-Poisson energy.

*Step 1. Basic estimates.* Let us prove that

$$\kappa \sum_{k=0}^{N} \|w^{k+1} - w^k\|^2 + \mathcal{E}_{2,0,T}(u^{N+1}, w^{N+1}) \leq \mathcal{E}_{2,0,T}(u^0, w^0) \tag{C.1}$$

for some $\kappa > 0$ and $\{w^k(x,\hat{x})\}_k$ is uniformly bounded in $W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.

Let $h(u,w;x,\hat{x}) = w(x,\hat{x})\|p_u(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla} + Tw(x,\hat{x})\log w(x,\hat{x})$. We may write $\mathcal{E}_{2,0,T}(u,w) = \int_{\widetilde{O}}\int_{\widetilde{O}^c} h(u,w;x,\hat{x})d\hat{x}dx$.

To prove (C.1), let us observe that

$$\begin{aligned} h(u^k,w^k;x,\hat{x}) - h(u^k,w^{k+1};x,\hat{x}) &\\ = \frac{\partial h}{\partial w}(u^k,w^{k+1};x,\hat{x})(w^k(x,\hat{x}) - w^{k+1}(x,\hat{x})) &\\ + \frac{\partial^2 h}{\partial w^2}(u^k,\overline{w};x,\hat{x})(w^k(x,\hat{x}) - w^{k+1}(x,\hat{x}))^2 & \end{aligned}$$

for an intermediate value $\overline{w}(x,\hat{x}) \in [w^k(x,\hat{x}), w^{k+1}(x,\hat{x})]$. Since

$$\frac{\partial h}{\partial w}(u^k,w^{k+1}) = 0 \quad \text{and} \quad \frac{\partial^2 h}{\partial w^2}(u^k,\overline{w}) = \frac{T}{\overline{w}},$$

it suffices to prove that the sequence $w^k$ is bounded independently of $k$, because this implies that also is $\overline{w}$. In that case we have

$$h(u^k,w^k;x,\hat{x}) - h(u^k,w^{k+1};x,\hat{x}) \geq \kappa(w^k(x,\hat{x}) - w^{k+1}(x,\hat{x}))^2 \qquad \text{(C.2)}$$

for some $\kappa > 0$.

To prove that $w^k$ is bounded independently of $k$ it suffices to observe that

$$w^k(x,\hat{x}) = \frac{1}{Z_{2,0,T}(u^{k-1};x)} \exp\left(-\frac{1}{T}\|p_{u^{k-1}}(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}\right), \qquad \text{(C.3)}$$

where the normalizing factor $Z_{2,0,T}(u^{k-1};x)$ is given by

$$Z_{2,0,T}(u^{k-1};x) = \int_{\widetilde{O}^c} \exp\left(-\frac{1}{T}\|p_{u^{k-1}}(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}\right) d\hat{x}. \qquad \text{(C.4)}$$

Now, we observe that by Lemma A.5, $u^k$ is uniformly bounded in $W^{1,2}(O)$, and by Lemma A.6 $\|p_{u^{k-1}}(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}$ is uniformly bounded. This implies that there exist $b > a > 0$ independent of $k$ such that

$$a \leq Z_{2,0,T}(u^{k-1};x) \leq b.$$

By the results of Section A this implies that $\{w^k(x,\hat{x})\}_k$ is uniformly bounded in $W^{1,\infty}(\widetilde{O} \times \widetilde{O}^c)$.

Now, using (C.2) we obtain

$$\begin{aligned}\kappa\|w^k - w^{k+1}\|_2^2 &\leq \mathcal{E}_{2,0,T}(u^k, w^k) - \mathcal{E}_{2,0,T}(u^k, w^{k+1}) \\ &= \mathcal{E}_{2,0,T}(u^k, w^k) - \mathcal{E}_{2,0,T}(u^{k+1}, w^{k+1}) \\ &+ \mathcal{E}_{2,0,T}(u^{k+1}, w^{k+1}) - \mathcal{E}_{2,0,T}(u^k, w^{k+1}) \\ &\leq \mathcal{E}_{2,0,T}(u^k, w^k) - \mathcal{E}_{2,0,T}(u^{k+1}, w^{k+1}),\end{aligned}$$

since $\mathcal{E}_{2,0,T}(u^{k+1}, w^{k+1}) - \mathcal{E}_{2,0,T}(u^k, w^{k+1}) \leq 0$ because $u^{k+1}$ is given by (4.2). By adding from $k = 0, \ldots, N$, we get (C.1).

*Step 2. Convergence to a critical point of $\mathcal{E}_{2,0,T}$ and regularity.* By Step 1 we may extract a subsequence $k_j$ such that $w^{k_j}$ weakly converges to some $w^*$ in $L^p$ for all $p \in [1, \infty]$ and $u^{k_j}$ converges to some $u^* \in W^{1,2}(O)$. By (C.1) also $w^{k_j+1}$ converges to $w^*$ in $L^p$ for all $p \in [1, \infty]$.

The equations satisfied by $u^{k+1}, w^{k+1}$ are

$$\begin{aligned}\Delta u^{k+1}(z) &= \operatorname{div} v^{k+1}(z) && z \in O \\ u^{k+1}(z) &= \hat{u}(z) && z \in \partial O,\end{aligned} \tag{C.5}$$

where

$$v^{k+1}(z) = \int_{\Omega_p} g(h) \int_{\widetilde{O}^c} w^{k+1}(z - h, \hat{x}) \nabla \hat{u}(\hat{x} + h) dy dh \tag{C.6}$$

and $w^{k+1}(x, \hat{x})$ is given by (C.3) and (C.4) with $k$ replaced by $k + 1$.

Notice that

$$v^{k_j}(z), v^{k_j+1}(z) \to v^*(z) := \int_{\Omega_p} g(h) \int_{\widetilde{O}^c} w^*(z - h, \hat{x}) \nabla \hat{u}(\hat{x} + h) dy dh$$

as $j \to \infty$. The convergence is also strong, since $v^k$ is uniformly bounded. Then, using (C.5) we have that

$$\begin{aligned}\|\nabla u^k - \nabla u^{k+1}\|_2^2 &= \int_{\mathbb{R}^N} (v^k - v^{k+1})(\nabla u^k - \nabla u^{k+1})\, dz \\ &\leq \|v^k - v^{k+1}\|_2 \|\nabla u^k - \nabla u^{k+1}\|_2.\end{aligned}$$

Since both $u^k$ and $u^{k+1}$ have the same boundary values, $u^k - u^{k+1}$ converges to 0 in $W^{1,2}(O)$. Thus $u^{k_j}, u^{k_j+1}$ both converge to $u^*$ in $L^2(O)$. We have

$$w^*(x, \hat{x}) = \frac{1}{Z_{2,0,T}(u^*; x)} \exp\left(-\frac{1}{T} \|p_{u^*}(x) - p_{\hat{u}}(\hat{x})\|_{g,\nabla}^2\right), \tag{C.7}$$

where the normalizing factor $Z_{2,0,T}(u^*; x)$ is given by

$$Z_{2,0,T}(u^*; x) = \int_{\widetilde{O}^c} \exp\left(-\frac{1}{T}\|p_{u^*}(x) - p_{\hat{u}}(\hat{x})\|^2_{g,\nabla}\right) d\hat{x}. \tag{C.8}$$

Notice that $u^*$ satisfies the boundary condition in (C.5). Thus $(u^*, w^*) \in \mathcal{A}_{2,0,T}$ is a critical point of $\mathcal{E}_{2,0,T}(u, w)$. Since $\mathcal{E}_{2,0,T}(u^*, w)$ is a strictly convex function of $w$, then $w^*$ is a minimum of $\mathcal{E}_{2,0,T}(u^*, w)$. Since $\mathcal{E}_{2,0,T}(u, w^*)$ is a strictly convex function of $u$, then $u^*$ is a minimum of $\mathcal{E}_{2,0,T}(u, w^*)$. By Remark 3, $(u^*, w^*)$ we have the regularity stated in the statement. □

# Part II

# Gradient-based video editing

Chapter 9

# Introduction

## 9.1 Problem definition

Digital editing of a captured video footage is becoming more and more common, mainly due to advances in computer graphics and computer vision techniques. Video editing tasks vary from basic operations such as trimming, cutting, splitting and resizing video segments to more elaborate ones such as editing objects' textures or, more generally, removing and adding objects in a video segment.

In this work, we consider the following video editing problem. We are given an image sequence where an object's surface has been edited in one or two reference frames, and a certain editing domain. We also require the knowledge of the motion in the editing domain. The objective is then to propagate, along motion trajectories, the edited information in the reference frame (or frames) throughout the editing domain. The resulting editing needs to be *spatially* and *temporally consistent*. This problem arises in other more complex video editing tasks, such as video inpainting and object touch-up as in Bhat et al. (2007). In what follows we discuss the concepts of temporal and spatial consistency.

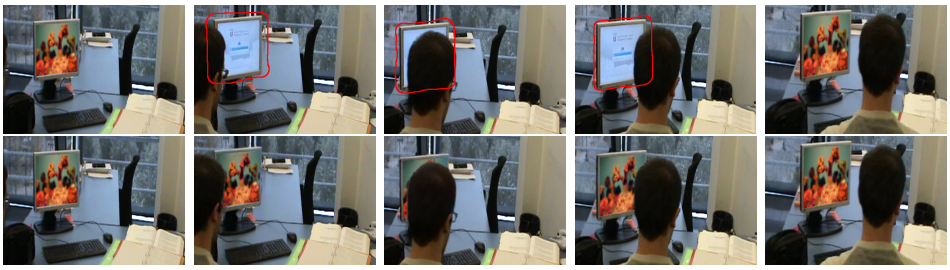


**Figure 9.1:** An example of an input and output of our model. The sequence has 20 frames in total. In the first row the user edits the first and last frames of the sequence. These are shown by the left most and right most images of the first row. In the remaining frames, we would like to propagate the edited information inside the area marked in red (the editing domain). In the second row, we show the output obtained using the proposed model.

## 9.2 Previous work

Temporal consistency refers to a smooth transition between successive frames, coherent with the motion in the sequence. Due to this constraint, the editing of a video cannot be reduced to a series of independent image editing problems. The temporal interdependence imposed by the motion in the sequence has to be taken into account. Since we desire our propagation to be temporally consistent, we are led to propagate the given information along the motion trajectories of the video.

When it comes to modelling the motion of a given sequence, one can distinguish between parametric and non-parametric models.

Parametric models work under assumptions made on the geometry of the scene. For example, the background is assumed to be static and piecewise planar Zhang et al. (2005); Jia et al. (2006). This model permits the computation of a closed form mapping between any pair of frames which can then be used to propagate information from one frame to another.

On the other hand, non-parametric models do not make assumptions on the geometry of the scene. These models usually estimate the motion in the sequence by the optical flow. There has been in recent years a considerable progress in optical flow computation. For example, state-of-the-art optical flow algorithms are able to deal with some large displacements and allow for sharp discontinuities in the movement. This is the case for Sun et al. (2010); Chambolle and Pock (2011); Brox and Malik (2011); Ayvaci et al. (2012)

to name a few. These methods still suffer from the "aperture" problem: the component of the motion vector tangent to the image level line cannot be estimated. In practice, to alleviate this problem a smoothness term is incorporated. The smoothness term causes a filling-in effect leading to dense flow fields, even if the aperture problem is present. Since we do not wish to restrict the scene geometry nor its motion, we will be using non-parametric models for the motion in the sequence.

Several approaches have been proposed in the literature to address the problem of propagating information along the optical flow to ensure temporal consistency. In video inpainting for example, some works inpaint first the optical flow and then propagate information along the inpainted flow to fill-in the inpainting domain. An optical flow with subpixel resolution requires a frame interpolation scheme. As it has been observed by Kokaram et al. (2005), using a bi-linear interpolation scheme to propagate the information along the optical flow presents a considerable blur in the results after just a few frames. In Kokaram et al. (2005), the problem is alleviated by using a higher order interpolation scheme. However, in Shiratori et al. (2006), the authors noted that though higher order interpolation schemes behave better than the bi-linear one, the blur artifact still persists. In this work we face the same problem. We address it by extending a scheme introduced in Facciolo et al. (2011) that allows for propagation of information along the optical flow through a large number of frames maintaining the sharpness of the result. This scheme gives good results while using bi-linear interpolation. It can also be used, providing similar or better results, with higher order interpolation schemes at the expense of extra computational costs.

An interesting related approach is followed in Bugeau et al. (2010). They integrate the optical flow, computing a set of motion trajectories that roughly covers the editing domain. These trajectories are then used to propagate the known color information inside the editing domain. For the computation of these trajectories, the optical flow itself has to be interpolated at subpixel positions. Since the optical flow is generally smoother than the frames, the interpolation errors committed are smaller and less noticeable in the resulting video.

Finally, let us mention the *unwrap mosaics* approach by Rav-Acha et al. (2008), which is interesting because it avoids estimating frame-to-frame motion. Instead, the authors propose to compute a static unwrapped texture, a sequence of occlusion masks, and a sequence of transformations from the unwrapped texture to each one of the frames in the video. The editing

is then performed directly on the unwrapped texture, and the changes are mapped back into the video sequence using the estimated transformations. The technique of the unwrap mosaics permits to handle a wide range of situations including zooms, geometric deformations and occlusions. The method relies however on a substantial algorithmic machinery including accurate video segmentation, keypoints tracking and nonlinear optimization for computing the texture and mappings. Also, since the mosaics are fixed, the illumination changes must be managed in a post-processing step.

The above mentioned works deal with propagating color information assuming that the color remains constant along trajectories. This assumption is often referred to in the literature as the *brightness constancy assumption*. However, due to shadows, reflections and other illumination changes, the color may change along trajectories. As a consequence the propagation of color might cause spatial inconsistencies between the editing domain and its surrounding.

Spatial consistency refers to a seamless integration of the editing with its spatial surrounding in each frame. In the image editing literature, spatial consistency is usually addressed using gradient-domain methods. These have been applied extensively for tasks such as seamless cloning and compositing (Pérez et al. (2003); Georgiev (2005)), shadow removal (Finlayson et al. (2006)), HDR compression (Fattal et al. (2002)), image inpainting (Arias et al. (2011); Komodakis and Tziritas (2007)), texture synthesis (Kwatra et al. (2005)), among others. Essentially, gradient-domain image editing is based on the manipulation of the image gradients instead of its grey levels. The modified gradients are then integrated to recover the resulting image. Typically, this is achieved by solving a Poisson equation with suitable boundary conditions. This procedure prevents the appearance of seams at the boundaries of the edited region. For a more detailed introduction to gradient-domain methods, the reader is referred to Agrawal and Raskar (2007). Since we want our propagation to be spatially consistent, we will work in the gradient-domain.

Some authors have already tackled the problem of extending gradient-domain image editing techniques to video. In Wang et al. (2005), it has been proposed to apply Poisson editing (Pérez et al. (2003)) directly to video by considering a video to be a three-dimensional volume and using the 3D-gradient to perform editing operations. Though this eliminates artifact such as flickering, it does not take motion into account.

In Bhat et al. (2007), it has been noted that using the 3D gradient leads

to severe ghosting artifact for videos with camera motion. For this reason, they use a 3D gradient where the temporal component is given by the convective derivative, *i.e.* the derivative along motion trajectories. That work deals with several video editing tasks, from which the most related to our application is the "object touch-up". They proceed in two steps: first they propagate the color information (using *Structure from Motion*). The result is temporally consistent in the editing domain, but may have spatial seams. To remedy this, a second step is performed. Using the spatial gradient of the propagated information, an energy functional is proposed with two terms. A term performing a Poisson image editing in each frame imposing spatial consistency; and, a term filtering along motion trajectories to further ensure temporal consistency. These two terms are balanced by a positive parameter. The resulting video is spatially and temporally consistent. This work has been further elaborated into a full-framework in Bhat et al. (2010) for image and video filtering.

A similar two step procedure was applied in Facciolo et al. (2011). The difference with Bhat et al. (2007) is that the first step consisting of propagating colors is done using the convective derivative (instead of using Structure from Motion). In Facciolo et al. (2011) a numerical scheme is used, the *de-blurring scheme for the convective derivative* (DSCD), which makes the propagation possible through a large number of frames without the blurring artifacts noted in Shiratori et al. (2006); Kokaram et al. (2005). Then, a similar two term energy to the one in Bhat et al. (2010) is used to remove spatial seams in a temporally consistent manner.

In Bhat et al. (2007, 2010) and Facciolo et al. (2011), the model to impose temporal consistency is based on the brightness constancy assumption. This makes it hard for the system to handle fast illumination changes along time. Furthermore, the propagation in Bhat et al. (2007, 2010) and Facciolo et al. (2011) needs to be divided in two steps: the first one is to obtain a temporally consistent gradient field, which is then integrated in the second step to achieve spatial consistency.

## 9.3 Our contribution

In this work we first discuss a mathematical model for temporal consistency, by generalizing the brightness-constancy assumption to allow for global additive illumination changes. This is achieved by working in the gradient domain. Based on this generalization, we propose an energy functional that

can be used to propagate gradient-domain information along motion trajectories. Its minimizers are temporally and spatially consistent, being able to handle sudden illumination changes. Although it is based on a model for temporal consistency that considers only global illumination changes, the variational formulation allows for some spatial variation of the illumination change.

The proposed formulation is simple, in the sense that it amounts to a quadratic energy whose minimization is performed by solving a system of linear equations. In addition to that, the only requirement is the knowledge of the optical flow (or any other dense estimate of the motion field).

As a use case, we consider the application where a user edits a frame by changing the texture of an objects' surface. In this application, we propagate the edited information throughout the rest of a given editing domain.

The proposed energy is motivated in Chapter 10, in a continuous setting. The discrete energy is presented in Chapter 11, together with the discretized operators for the spatial gradient and the convective derivative. The handling of occlusions in the model is also explained in that Section. In Chapter 12 we present the *de-blurring scheme for the convective derivative* (DSCD), introduced in Facciolo et al. (2011). Experimental results on real sequences and discussions about the method's limitations are presented in Chapter 13. Finally, Chapter 14 gives some concluding remarks.

Chapter 10

# Deriving the model

In this chapter we discuss some mathematical models for temporal consistency of a video. This helps motivating our energy. Throughout the text we will use symbols with **boldface** to indicate vector valued quantities and matrices. Non-boldface symbols will indicate scalar valued quantities. Let us note that we will not make that distinction when we discuss examples of 1D videos, and in these cases we will use non-boldface symbols.

## 10.1 Models for temporal consistency

We consider a spatio-temporal domain $\Omega \times [0,T]$, where $\Omega \subset \mathbb{R}^2$ is a rectangular domain, and $T > 0$. In some places in the text, to avoid cluttered equations, we will use $\Omega^T$ as a notational shorthand for the video domain $\Omega \times [0,T]$. Let $u : \Omega \times [0,T] \to \mathbb{R}$ be a given scalar video and let $\boldsymbol{v} : \Omega \times [0,T] \to \mathbb{R}^2$ be the corresponding motion field. The value of the motion field at $(\boldsymbol{x},t) \in \Omega \times [0,T]$, $\boldsymbol{v}(\boldsymbol{x},t)$, represents the velocity of the projection of a particle in the 3D scene onto the image plane (Horn (1986)). The trajectory of the particle can be obtained by solving the following ODE:

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}(t) = \boldsymbol{v}\left(\boldsymbol{x}(t),t\right) , \tag{10.1}$$

where $t \in [0,T]$. For simplicity we assume in this chapter that the functions we consider can be differentiated as many times as needed.

In what follows we review the brightness-constancy assumption, widely used in the literature to compute the optical flow. Then we discuss a generalization of this model to account for spatial global illumination changes. Henceforth we refer to this as the *Global Brightness Change* assumption (GBC). Our variational formulation will be based on this generalization. Finally we discuss the differences between the GBC and the gradient-constancy assumption, also used in the context of optical flow computation.

### Brightness-constancy assumption

For a *Lambertian* object under uniform constant illumination, the brightness of an object's particle does not change in time. This implies that $u(\boldsymbol{x}, t)$ is constant along trajectories, leading to the following brightness-constancy equation (Horn and Schunck (1981)):

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} u(\boldsymbol{x}(t), t) = \boldsymbol{\nabla} u(\boldsymbol{x}(t), t) \cdot \boldsymbol{v}(\boldsymbol{x}(t), t) + \frac{\partial u}{\partial t}(\boldsymbol{x}(t), t)\,, \tag{10.2}$$

where $\frac{\mathrm{d}}{\mathrm{d}t} u$ is the derivative of $u$ along trajectories $\boldsymbol{x}(t)$ and $\boldsymbol{\nabla} u$ refers to the spatial gradient of $u$. Let us define the *convective derivative* as

$$\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) := \boldsymbol{\nabla} u(\boldsymbol{x}, t) \cdot \boldsymbol{v}(\boldsymbol{x}, t) + \frac{\partial u}{\partial t}(\boldsymbol{x}, t)\,. \tag{10.3}$$

Notice that the convective derivative permits to express the derivative along trajectories. We use this notation to shorten our expressions and to make explicit its connection with the field $\boldsymbol{v}$.

The brightness constancy assumption has been used extensively for the computation of optical flow (Weickert et al. (2006); Baker et al. (2011)), and recently for video interpolation given an optical flow (Kokaram et al. (2005); Shiratori et al. (2006); Bugeau et al. (2010); Facciolo et al. (2011)).

### Global brightness change assumption (GBC)

Under illumination changes, the brightness-constancy assumption does not hold. In this section we generalize this assumption to account for spatially constant, additive illumination changes. In that case, if we follow the trajectories of two particles, the difference of their colors remains constant. We base our model on this basic observation.

Let us consider two particles that at time $t$ are in positions $\boldsymbol{x}_0 \in \Omega$ and $\boldsymbol{y}_0 \in \Omega$. We denote their trajectories by $\boldsymbol{\varphi}(\boldsymbol{x}_0, s)$ and $\boldsymbol{\varphi}(\boldsymbol{y}_0, s)$, with $s \in [0, T]$. Then for $k > 0$,

$$u(\boldsymbol{\varphi}(\boldsymbol{y}_0, t+k), t+k) - u(\boldsymbol{\varphi}(\boldsymbol{x}_0, t+k), t+k) = u(\boldsymbol{y}_0, t) - u(\boldsymbol{x}_0, t)\,. \quad (10.4)$$

This is represented by Figure 10.1c. After rearranging terms, dividing by $k$, and taking $k \to 0$ we obtain $\partial_{\boldsymbol{v}} u(\boldsymbol{y}_0, t) = \partial_{\boldsymbol{v}} u(\boldsymbol{x}_0, t)$. Since this holds for all $\boldsymbol{x}_0, \boldsymbol{y}_0 \in \Omega$, we have that $\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t)$ is a function only of $t$, *i.e.*

$$\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = g(t)\,. \quad (10.5)$$

Here $g(t)$ expresses the global illumination change rate. Thus, (10.5) generalizes the brightness-constancy model taking into consideration global changes in illumination.

Taking the spatial gradient on both sides of Eq. (10.5), we get a differential version of (10.4)

$$\boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = \boldsymbol{0}\,. \quad (10.6)$$

Our variational model is based on this equation. A Taylor expansion of (10.6) leads to

$$u(\boldsymbol{y}_0 + k\boldsymbol{v}(\boldsymbol{y}_0, t), t+k) - u(\boldsymbol{x}_0 + k\boldsymbol{v}(\boldsymbol{x}_0, t), t+k) \approx u(\boldsymbol{y}_0, t) - u(\boldsymbol{x}_0, t)\,, \quad (10.7)$$

which is an linearized version of (10.4).

**Remark.** The GBC can be regarded as a particular case of the *Generalized Dynamic Image Model* (GDIM) proposed by Negahdaripour Negahdaripour (1998), in the context of optical flow computation. GDIM is a more general model for temporal consistency, accounting for additive and multiplicative illumination changes, along with their spatial variations. There, the assumption is that the convective derivative fulfills $\partial_v u(x,t) = m(x,t)u(x,t) + g(x,t)$, where $m$ and $g$ are referred to as the *multiplier* and *offset* fields respectively. If we assume a spatially constant offset field (*i.e.* a global additive illumination change), and a zero multiplier field, the model reduces to the GBC model. In the next Section where we propose an energy based on the GBC model, the restriction of a global illumination change will be somewhat relaxed allowing some spatial variation on the illumination change rate.

### Comparison with the gradient-constancy assumption

It is interesting to compare our assumption with the related assumption where

$$u(\boldsymbol{\varphi}(\boldsymbol{x}_0, t+k) + \boldsymbol{h}, t+k) - u(\boldsymbol{\varphi}(\boldsymbol{x}_0, t+k), t+k) \\ = u(\boldsymbol{x}_0 + \boldsymbol{h}, t) - u(\boldsymbol{x}_0, t)\,.$$

Note that in this model $\boldsymbol{y}_0 = \boldsymbol{x}_0 + \boldsymbol{h}$ is mapped to the next frame using the mapping of $\boldsymbol{x}_0$ and not its own. In this case, the underlying differential equation is

$$\partial_{\boldsymbol{v}} \boldsymbol{\nabla} u(\boldsymbol{x}, t) = \mathbf{0}\,, \tag{10.8}$$

which in the literature is referred to as the gradient-constancy assumption Uras et al. (1988); Weickert et al. (2006); Papenberg et al. (2006). Clearly, in a captured video sequence the gradient needs not to remain constant along trajectories, except for the particular case in which the motion field $\boldsymbol{v}(\cdot, t)$ is constant (corresponding to a translational movement).

For the purpose of optical flow computation, as discussed in Uras et al. (1988), even when the movement is not translational the gradient-constancy assumption is a good approximation, since the change in the gradient along trajectories between two consecutive frames is minor. However, in our application where we propagate information along a large number of frames, small changes in the gradient accumulate along a trajectory becoming significant. Figure 10.1 illustrates the three discussed models and shows their differences.

## 10.2 Proposed energy in the continuous setting

In this section we present a continuous variational model for video editing imposing (10.6). We then derive the natural boundary conditions of the model, and we present two application scenarios differing on the choice of the boundary conditions.

Throughout this section we consider a continuous scalar video $u_0 : \Omega \times [0, T] \to \mathbb{R}$, and an editing domain $O \subset \Omega \times [0, T]$ with a Lipschitz boundary Adams (1975) (to simplify, we can consider that $O$ has a smooth boundary). As in the previous section, we will sometimes use $\Omega^T$ as a notational shorthand for the video domain $\Omega \times [0, T]$. We denote temporal "slices" of

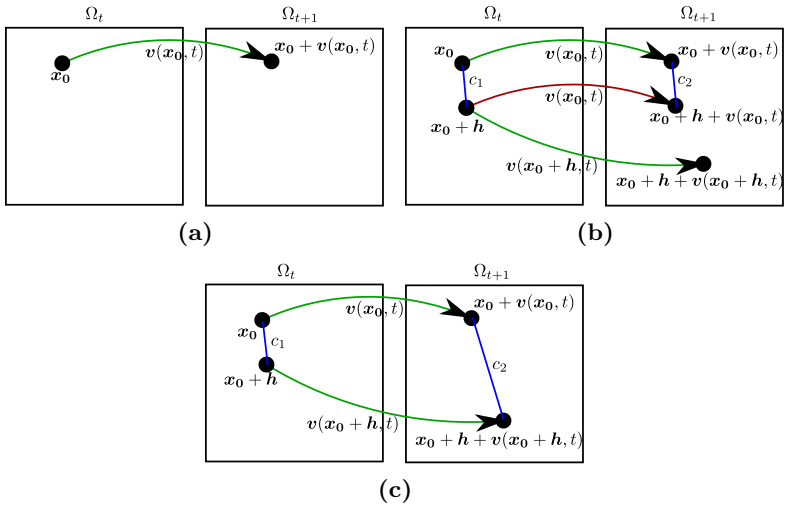



**Figure 10.1:** An illustration of the different models for temporal consistency. $\Omega_t$ and $\Omega_{t+1}$ refer to temporal "slices" of $\Omega$ at time $t$ and $t+1$ respectively. (a) Shows the brightness constancy assumption. A point and its projection by the optical flow maintain the same color. (b) Illustrates the gradient constancy assumption. Here $c_1 = u(\boldsymbol{x}_0+\boldsymbol{h},t)-u(\boldsymbol{x}_0,t)$ and $c_2 = u(\boldsymbol{x}_0+\boldsymbol{h}+\boldsymbol{v}(\boldsymbol{x}_0,t),t+1)-u(\boldsymbol{x}_0+\boldsymbol{v}(\boldsymbol{x}_0,t),t+1)$ and it is assumed that $c_2-c_1=0$. (c) Depicts the global brightness change model. Here $c_2 = u(\boldsymbol{x}_0+\boldsymbol{h}+\boldsymbol{v}(\boldsymbol{x}_0+\boldsymbol{h},t),t+1)-u(\boldsymbol{x}_0+\boldsymbol{v}(\boldsymbol{x}_0,t),t+1)$ and it is assumed that $c_2-c_1=0$. This model differs from the gradient-constancy assumption by projecting $\boldsymbol{x}_0+\boldsymbol{h}$ to the next frame using its own optical flow and not the one of $\boldsymbol{x}_0$.

$O$ by $O_t = \{\boldsymbol{x}\in\Omega : (\boldsymbol{x},t)\in O\}$. Similarly, temporal slices of $\Omega\times[0,T]$ are denoted by $\Omega_t : t\in[0,T]$ representing the frames of the continuous video. An illustration of these domains can be seen in Figure 10.2.

The proposed energy imposes the global brightness change model by penalizing deviations from condition (10.6):

$$E(u) = \int_0^T \int_{O_t} \|\boldsymbol{\nabla}\partial_{\boldsymbol{v}} u(\boldsymbol{x},t)\|^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\,. \tag{10.9}$$

While Eq. (10.6) implies a spatially constant illumination change, the variational model allows some spatial variation on $\partial_{\boldsymbol{v}} u$. This is a useful feature in practical applications since it accounts for localized light sources, shadows and reflections, as long as they manifest at the boundary of the editing domain.

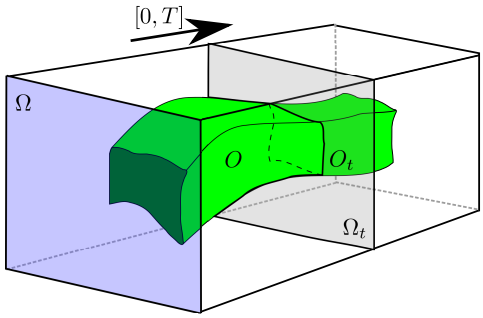



**Figure 10.2:** Illustration of an editing domain $O$ inside of the video domain $\Omega \times [0,T]$. $O_t$ and $\Omega_t$ are temporal slices at time $t$ of $O$ and $\Omega \times [0,T]$ respectively. To simplify the figure we do not show all complexities that a general editing domain may have (see Fig.10.3).

In the energy of Eq. (10.9) the motion field $\boldsymbol{v}$ is assumed to be known, therefore we only work with editing tasks that do not modify the motion field of the original video. Such tasks consist typically in changing the texture or the appearance of an object's surface visible in the sequence. The editing domain $O$ should contain the spatio-temporal volume described by the surface to be edited. The energy (10.9) could also be applied to other video editing tasks, such as video inpainting. In that case, the motion field needs to be inpainted first (Kokaram et al. (2005); Shiratori et al. (2006)).

Since this is a gradient-based energy, the choice of the boundary conditions plays an essential role in the application of the model. Before presenting them, let us discuss the types of video editing applications we tackle using (10.9). We consider two application settings.

In the first one, the user provides a first frame $\Omega_0$, edited with some image editing tool (automatically or with some user intervention). Minimizing (10.9) with the edited first frame set as a Dirichlet boundary condition on $u$, propagates the edited information to the rest of the sequence. We call this the *one-lid* setting where the *lid* refers to the first frame $\Omega_0$, containing the information to be propagated. The other temporal end of the video at $t = T$ is left free. Note that one could edit the last frame $\Omega_T$ and set it as Dirichlet boundary condition, while leaving the first frame $\Omega_0$ free. Henceforth, when we refer to the one-lid setting we always consider $\Omega_0$ to be the lid.

For the second setting, both the first and last frames are edited and provided

by the user. Both of them are set as Dirichlet boundary conditions. We refer to this as the *two-lid* setting. In this case, minimizing (10.9) yields an interpolation along trajectories between both lids. If the editing on both lids is not consistent with the surface's motion in the sequence, the solution will present a temporal blending between the lids. The two-lid setting allows to treat cases in which the modified surface is occluded and then dis-occluded by another object in the sequence.

For both the *one-lid* and the *two-lid* settings, we assume that any point of the spatio-temporal volume described by the edited surface is reachable by at least one trajectory originating from a lid (see Figure 10.3).

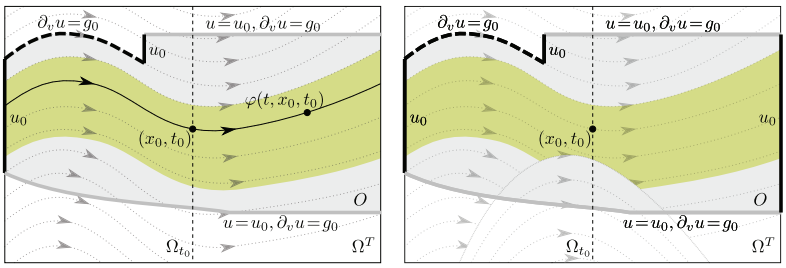
We assume that the editing domain $O$ is given. In a practical editing application, an editing domain can be computed by tracking the portion of the surface that the user wants to edit. An accurate subpixel tracking of the edited surface in a general situation is a problem in itself. However, many available tracking algorithms provide a good approximate tracking Yilmaz et al. (2006). Thus we consider a general editing domain $O$ without requiring it to be a precise tracking of the actually edited surface (as long as this surface is contained in $O$). As a consequence, some parts of the editing domain may not belong to the edited surface. In these places, the original video should be restored. This is shown in Figure 10.3. The trajectory of the edited surface is shown in green. The gray regions of $O$ are not intended to be modified.

## 10.3 Boundary conditions

Let us first introduce some notation. Let us denote by $\partial O$ the boundary of $O$. Let $\partial O_t$ denote the boundary of the slice $O_t$, $t \in [0, T]$. Let us denote by $\boldsymbol{\nu}^O = (\boldsymbol{\nu}^O_{\boldsymbol{x}}, \nu^O_t) \in \mathbb{R}^3$ the outer unit normal to $\partial O$ (a vector in the unit sphere of $\mathbb{R}^3$) and let $\boldsymbol{\nu}^{O_t}$ be the outer unit normal to $\partial O_t$ (a vector in the unit circle of $\mathbb{R}^2$). Notice that the normal $\boldsymbol{\nu}^O$ exists at any point if $\partial O$ is smooth, and at almost any point with respect to the Hausdorff (surface) measure $\mathcal{H}^2$ on $\partial O$ if we assume $\partial O$ to be Lipschitz Ambrosio et al. (2000); Adams (1975).

Let us introduce the following decomposition of the boundary of $O$. We consider the *lateral* boundary of $O$ as the set

$$\partial O_{\text{lat}} := \{(\boldsymbol{x}, t) \in \partial O : t \in (0, T)\}.$$




**(a)** *One-lid* problem. **(b)** *Two-lid* problem.

**Figure 10.3:** Domains and Dirichlet boundary conditions for a one-lid (a), and a two-lid problem (b) on a video with one spatial dimension. Trajectories of the edited object are marked in green (middle area). Note that (b) exhibits an occlusion and dis-occlusion of the edited object.

It corresponds to excluding the temporal ends of $O$, that is $O_0$ and $O_T$, from $\partial O$. The lateral boundary of $O$ can be further classified into three parts. First, the *tangential* boundary, given by

$$\partial O_{\text{tang}} = \{(\boldsymbol{x}, t) \in \partial O : \nu_t^O + \boldsymbol{v} \cdot \boldsymbol{\nu}_{\boldsymbol{x}}^O = 0\},$$

which corresponds to the segments of $\partial O$ that are tangential to the motion trajectories. In the example of Fig. 10.3a the tangential boundary is the dashed part of $\partial O$. Second, the *vertical* boundary, $\partial O_{\text{vert}}$, consisting of the segments of the lateral boundary parallel to $\Omega_0$, *i.e.*

$$\partial O_{\text{vert}} := \{(\boldsymbol{x}, t) \in \partial O_{\text{lat}} : |\boldsymbol{\nu}^O(\boldsymbol{x}, t) \cdot \boldsymbol{e}_t| = 1\},$$

where $\boldsymbol{e}_t = (0, 0, 1)$. It is formed by the vertical walls of the boundary except the initial and final slices $O_0$ and $O_T$, respectively. This is shown by the black vertical segments in Fig. 10.3a. Third, the *oblique* boundary, denoted by $\partial O_{\text{obli}}$, is the remaining non-tangential and non-vertical boundary (gray segments of $\partial O$ in Fig. 10.3a). That is

$$\partial O_{\text{obli}} := \partial O_{\text{lat}} \setminus (\partial O_{\text{tang}} \cup \partial O_{\text{vert}}).$$

Both $O_{\text{vert}}$ and $O_{\text{obli}}$ are non-tangential. However, they have to be distinguished because they admit different boundary conditions, as will be seen below.

Let us mention that a more detailed study of boundary conditions will be given in Appendix E. For the moment we introduce and motivate them based on more intuitive arguments.

In what follows we use $u_0$ to denote the original image sequence where the editing on the lid(s) has been performed.

### Boundary conditions for the *one-lid* setting

We impose the following boundary conditions for $(\boldsymbol{x}, t) \in (O_0 \times \{0\}) \cup \partial O_{\text{lat}}$, excluding the points $(\boldsymbol{x}, t)$ of the lateral boundary with $\boldsymbol{x} \in \partial\Omega$:

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}, 0), \qquad \boldsymbol{x} \in O_0 \tag{10.10}$$

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \partial O_{\text{vert}}, \tag{10.11}$$

$$\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = g_0(\boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \partial O_{\text{tang}} \setminus \partial\Omega^T, \tag{10.12}$$

$$\begin{array}{l} u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t) \\ \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = g_0(\boldsymbol{x}, t) \end{array}, \qquad (\boldsymbol{x}, t) \in \partial O_{\text{obli}} \setminus \partial\Omega^T, \tag{10.13}$$

where $g_0$ is a given video. We have used the shorthand $\Omega^T = \Omega \times [0, T]$.

Eqs. (10.10) and (10.11) correspond to the boundary conditions on vertical segments of the boundary. Note that they could be merged together into a single condition as

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t), \qquad t \in [0, T), \ |\boldsymbol{\nu}^O(\boldsymbol{x}, t) \cdot \boldsymbol{e}_t| = 1.$$

We write them separately to highlight the Dirichlet boundary condition that fixes the first *lid* containing the editing provided by the user.

On the lateral boundary, there are boundary conditions on $u$ and on its convective derivative $\partial_{\boldsymbol{v}} u$. The latter specifies the rate of illumination change at $\partial O$. The illumination change rate in the interior of $O$ corresponds to a smooth spatial interpolation of $g_0$ given at the boundary. In a typical editing application we set $g_0 = \partial_{\boldsymbol{v}} u_0$. In this way we impose in the editing domain the ambient illumination change of the original sequence.

The boundary conditions on $u$ apply only when trajectories cross the boundary, leaving or entering $O$. Note that if $O$ corresponds to a precise tracking of the edited surface (the green domain in Fig. 10.3a), trajectories will only cross the boundary at the temporal ends, $O_0$ and $O_T$. Therefore, all the lateral boundary will be tangential and only Eq. (10.12) will apply, specifying the illumination change rate. In this case, the only Dirichlet boundary conditions on $u$ are the ones at the lid. The solution is then obtained by propagating (interpolating) the data at $O_0$ ($O_0$ and $O_T$) along trajectories,

while accommodating for the illumination changes specified at the boundary.

In a more general case, we do not require $O$ to be a precise tracking of the surface to be edited, as long as it contains the surface. In such cases, other trajectories, not belonging to the target surface, are included in $O$, and may leave or enter the domain through non-tangential segments of the boundary (either at the temporal ends $O_0, O_T$ or through the lateral boundary). The Dirichlet boundary conditions of Eqs. (10.11) and (10.13) apply in these cases.

Note that in the previous boundary conditions, none was specified on $\partial\Omega$, apart from the lids. These will be specified later when we derive the Euler-Lagrange equation in Section 10.4.

**Remark.** Notice that the vertical and oblique parts of the boundary admit different boundary conditions, although both of them are non-tangential. The rigorous derivation of possible boundary conditions in the different parts of the boundary will be given in Appendix E. But some intuition on the reasons can be gained by looking at the cases in which the minima are attained with zero energy. This happens only for some choices of the boundary conditions, referred to as *compatible boundary conditions*. In such cases $u$ comes as the solution of the PDE

$$\nabla \partial_v u(x,t) = 0.$$

To solve this PDE, we proceed in two steps. In the first step we integrate spatially the equation, to get $\partial_v u(x,t) = g(t)$. This integration is independent for each temporal slice. It requires the value of $g(t)$ to be specified at the spatial boundary $\partial O_t$ of the slice $O_t$. Note that, modulo sets of zero surface measure, $\bigcup_{t\in[0,T]} \partial O_t = \partial O_{\text{tang}} \cup \partial O_{\text{obli}}$. This justifies why $g$ has to be specified at $\partial O_{\text{tang}} \cup \partial O_{\text{obli}}$, but not on $\partial O_{\text{vert}}$. In the second step, we integrate $\partial_v u(x,t) = g(t)$ along optical flow trajectories. To perform this integration we need the value of $u$ whenever a trajectory crosses the editing domain, *i.e.* on the non-tangential components of $\partial O$. As an illustration, in Appendix F we provide two analytical examples for non-zero energy solutions, showing how the solution can be computed from these boundary conditions.

### Boundary conditions for the *two-lid* setting

Here, in addition to the boundary conditions of the one-lid case, the frame $O_T$ is specified as a Dirichlet boundary condition:

$$u(\boldsymbol{x}, T) = u_0(\boldsymbol{x}, T), \quad \boldsymbol{x} \in O_T. \tag{10.14}$$

This setting is relevant to handle, for example, the case in which part of the editing domain is occluded and then dis-occluded (see example in Figure 10.3b).

**Remark.** As proved in Sadek et al. (2012), the boundary conditions (10.10), (10.11), (10.12) and (10.13) permit to prove that there is a unique minimizer of the energy. At an intuitive level, one can say that

1. Each trajectory needs to have at least one Dirichlet boundary condition on $u$.
2. At each temporal slice we need that a Dirichlet condition on $\partial_{\boldsymbol{v}} u$ is specified on a set of positive length of the boundary of $O_t$.

Analogous conditions are needed in the discrete case.

As a consequence, the model cannot handle cases in which a point is dis-occluded and occluded again, since its corresponding trajectory will not reach a Dirichlet boundary condition on the boundary of $O$. To solve this, one could partition the problem into several smaller ones fulfilling conditions 1 and 2. An example of this is given later in Section 13, Figure 13.11.

## 10.4 Euler-Lagrange equation

The Euler-Lagrange equation of energy (10.9) is given by the following fourth order PDE

$$\partial^*_{\boldsymbol{v}} \mathrm{div} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = 0, \qquad (\boldsymbol{x}, t) \in O, \tag{10.15}$$

where div is the spatial divergence adjoint to $-\boldsymbol{\nabla}$ and $\partial^*_{\boldsymbol{v}}$ denotes the adjoint operator of the convective derivative, given by $\partial^*_{\boldsymbol{v}} f = -\frac{\partial f}{\partial t} - \mathrm{div}(vf)$.

For the one-lid setting, in addition to the Dirichlet boundary condition discussed above, the following Neumann type boundary conditions apply on $\partial O \cap \partial \Omega^T$:

$$\operatorname{div} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = 0, \qquad t = T, \tag{10.16}$$

$$\boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0, \qquad (\boldsymbol{x}, t) \in \partial O_{\text{tang}} \cap \partial \Omega^T, \tag{10.17}$$

$$\begin{array}{l} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0 \\ \operatorname{div} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = 0 \end{array}, \qquad (\boldsymbol{x}, t) \in \partial O_{\text{obli}} \cap \partial \Omega^T. \tag{10.18}$$

In the two-lid setting, condition (10.16) does not apply, since $O_T$ has a Dirichlet boundary condition on $u$.

We refer the reader to Appendix E for a derivation of the Euler-Lagrange equation and its boundary conditions.

Chapter 11

# Discrete model

In this chapter we derive our discrete energy as a discretization of (10.9). We start by deriving a discrete version of operator $\boldsymbol{\nabla}\partial_{\boldsymbol{v}}$.

## 11.1 Discretization of operator $\boldsymbol{\nabla}\partial_{\boldsymbol{v}}$

For simplicity we consider now that $\Omega \subset \mathbb{R}$, *i.e.* $u$ is a one dimensional video; the resulting discretization can be easily extended to higher spatial dimensions (see Section 11.3). We consider a discrete video obtained by regularly sampling the continuous one with a spatial step $h$ and a temporal step $k$. Let us approximate operator (10.6) at $(x_0, t_0)$.

Using a forward difference scheme for the spatial derivative, we have

$$\frac{\partial}{\partial x}\partial_v u(x_0, t_0) \approx \frac{1}{h}\left[\partial_v u(x_0 + h, t_0) - \partial_v u(x_0, t_0)\right].$$

The convective derivatives can be approximated with a forward difference scheme as well:

$$\partial_v u(x_0 + h, t_0) \approx \frac{1}{k}[u(\varphi(x_0 + h, t_0 + k), t_0 + k) - u(x_0 + h, t_0)],$$

where in the last term we used that $\varphi(x, t_0) = x$. Similarly we have that

$$\partial_v u(x_0, t_0) \approx \frac{1}{k}[u(\varphi(x_0, t_0 + k), t_0 + k) - u(x_0, t_0)].$$

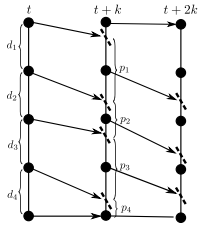



**Figure 11.1:** Illustration of the proposed discrete operator. The operator computes the difference between two points on a grid and their projection onto the next frame by the optical flow. In the figure, the operator computes $d_1 - p_1$ , $d_2 - p_2$, etc... Our energy is based on this operator, imposing the squared $L_2$ norm of these differences to be constant along the video.

The value of $\varphi(x, t + k)$ can be approximated as follows:

$$\varphi(x, t_0 + k) \approx x + kv(x, t_0)\,.$$

For subpixel motions, $x+kv(x, t_0)$ will fall outside of the sampling grid, and $u(x + kv(x, t_0), t_0 + k)$ has to be interpolated from the available samples. We will denote by $\hat{u}(x + kv(x, t_0), t_0)$ the interpolated value. When using bi-linear interpolation, this results in an *upwind scheme with an adaptive stencil* for the convective derivative, as the one in Zhou et al. (1998).

Putting everything together, we have the following operator

$$\begin{aligned}\frac{\partial}{\partial x}\partial_v u(x_0, t_0) \approx \\ \frac{1}{kh}\left[\hat{u}(x_0 + h + kv(x_0 + h, t_0), t_0 + k) - u(x_0 + h, t_0)\right] - \\ \frac{1}{kh}\left[\hat{u}(x_0 + kvf(x_0, t_0), t_0 + k) - u(x_0, t_0)\right]. \quad (11.1)\end{aligned}$$

Figure 11.1 illustrates the discrete operator (11.1).

## 11.2 The discrete energy

In this section we propose our variational model in the discrete setting. Let us first define the following notation. We now consider $u$ to be a scalar discrete video defined as a function $u : \Omega \times \{0, 1, \cdots, T\} \to \mathbb{R}$. Here $\Omega \times \{0, 1, \cdots, T\}$ is the discrete spatio-temporal domain, and $\Omega \subset \mathbb{Z}^2$ is

a rectangular discrete domain (the spatial domain of each frame). As in the continuous setting, we will use $\Omega^T$ as a notational shorthand for the video domain $\Omega \times \{0, 1, \ldots, T\}$, to avoid cluttered equations at some locations in the text. Let us also denote by $O \subset \Omega \times \{0, 1, \cdots, T\}$ the editing domain. Notice that we are using the same notations both for continuous and discrete domains. Each case will be clear from the context.

For a discrete video we use the optical flow computed on the original video $u_0$, as a computable approximation of the motion field. The *forward optical flow* $v^f : \Omega \times \{0, 1, \cdots, T\} \to \mathbb{R}^2$ establishes a correspondence between $(x, t)$ and $(x + v^f(x, t), t + k)$. Similarly, we define the *backward optical flow* as the vector field $v^b : \Omega \times \{0, 1, \cdots, T\} \to \mathbb{R}^2$. At frame $t$, $v^b(\cdot, t)$ establishes a correspondence with frame at time $t - k$.

We consider the discrete energy:

$$E_{\boldsymbol{\kappa}}(u) = \sum_{(\boldsymbol{x},t)\in\widetilde{O}} \|\boldsymbol{\kappa}(\boldsymbol{x}, t)\boldsymbol{\nabla}\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t)\|^2. \tag{11.2}$$

The discrete gradient $\boldsymbol{\nabla}$ and convective derivative $\partial_{\boldsymbol{v}}$ are presented in Section 11.3. $\boldsymbol{\kappa} : \Omega \times \{0, 1, \cdots, T\} \to \{0, 1\}^{2\times 2}$ is an *occlusion tensor* which will be defined in Section 11.4. The domain $\widetilde{O} \subseteq \Omega \times \{0, 1, \cdots, T\}$ is defined as the set of video pixels whose $\boldsymbol{\nabla}\partial_{\boldsymbol{v}}$ stencil intersects $O$. Our implementation of the boundary conditions specified in Section 10.2 is based on this domain, together with the operators presented next.

## 11.3 Definition of the operators

In what follows we define the spatial gradient and the convective derivative operators as a generalization to two spatial dimensions of the discretizations presented in Section 11.1, considering $k = h = 1$.

Our definition of the operators will integrate the different boundary conditions discussed in Section 10.2. We define the operators over $\Omega \times \{0, 1, \cdots, T\}$. In doing so, the Dirichlet boundary conditions on $\partial O$ are straightforwardly implemented by assuming that $u = u_0$ in $\widetilde{O} \setminus O$. That is, if some of the values needed by the operators fall in $\widetilde{O} \setminus O$, the values of $u_0$ are used. The Neumann type boundary conditions on $\partial\Omega$ will be incorporated in the definitions of the operators.

Throughout this section we will use a running example to illustrate the definitions of the operators. Figure 11.2 shows a discrete sequence with

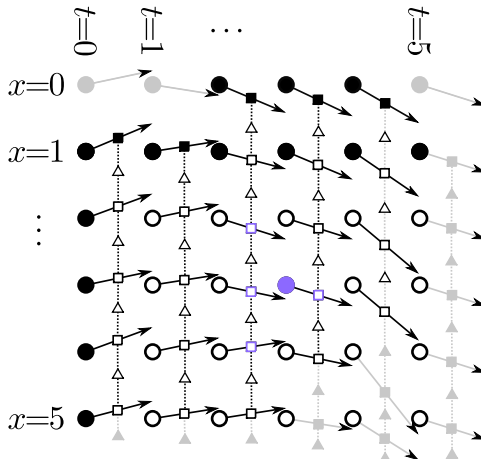



**Figure 11.2:** Graphical representation of the discrete operators for a 1D video sequence. The circles represent pixels of the video. Each arrow originating from a circle at $(x,t)$ represents the forward optical flow $v^f(x,t)$. The square attached to the optical flow $v^f(x,t)$ represents the forward convective derivative $\partial_v^f u(x,t)$, computed as the difference between the values at both ends of the arrow, that is $\hat{u}(x+v^f(x,t),t+1)-u(x,t)$. The spatial gradient of the convective derivative at $(x,t)$, $\nabla^f \partial_v^f u(x,t)$, is represented by the triangle connecting the squares that represent $\partial_v^f u(x+1,t)$ and $\partial_v^f u(x,t)$. See text for more details. Note that the domains of $\partial_v^f u$ and $\nabla^f \partial_v^f u$ are the same as the video domain, namely $\Omega \times \{0,1,\dots,T\}$. Let us stress that the squares and triangles are just a graphical representation of the operators. The fact that they are drawn at subpixel positions does not imply that the operators are defined on a subpixel grid.

one spatial dimension: $\Omega \times \{0,1,\dots,T\} = \{0,1,\dots,5\} \times \{0,1,\dots,5\}$. The circles represent the pixels in the video. Pixels in the editing domain $O$ are depicted in white, whereas the black pixels correspond to $\widetilde{O} \setminus O$. The rest of the pixels in $\Omega$ that are not involved in the energy are shown as gray circles. There is a forward optical flow $v^f(x,t)$ at each pixel, represented by the vector $(v^f(x,t),1) \in \mathbb{R}^2$.

### Convective derivative operator

Let us introduce some useful notation before presenting the definition. For each video pixel $(\boldsymbol{x},t)$ we define its forward interpolation neighborhood $N^f(\boldsymbol{x},t)$ as the set containing the grid positions surrounding $(\boldsymbol{x}+\boldsymbol{v}^f(\boldsymbol{x},t),t+1)$. Analogously, $N^b(\boldsymbol{x},t)$ contains those surrounding $(\boldsymbol{x}+\boldsymbol{v}^b(\boldsymbol{x},t),t-1)$.

The values at these pixels are used to bi-linearly interpolate the subpixel values at $(\boldsymbol{x} \pm \boldsymbol{v}^{f/b}(\boldsymbol{x},t), t \pm 1)$.

Note that, depending on the flow vector, it might happen that for a pixel $(\boldsymbol{x},t)$, the forward interpolation neighborhood $N^f(\boldsymbol{x},t)$ falls partly (or completely) ouside $\Omega_{t+1}$ (pixels with a gray arrow in Figure 11.2). We will distinguish the set of pixels which have their full interpolation neighborhood $N^f(\boldsymbol{x},t)$ in $\Omega_{t+1}$ by defining the set $S_t^f = \{\boldsymbol{x} \in \Omega_t : N^f(\boldsymbol{x},t) \subset \Omega_{t+1}\}$. Analogously, for the backward optical flow, we define the set $S_t^b = \{\boldsymbol{x} \in \Omega_t : N^b(\boldsymbol{x},t) \subset \Omega_{t-1}\}$.

The discrete *forward* convective derivative using the forward optical flow is then defined at a point $(\boldsymbol{x},t) \in \Omega \times \{0,1,\ldots,T\}$ as follows

$$\partial_{\boldsymbol{v}}^f u(\boldsymbol{x},t) = \begin{cases} \hat{u}\left(\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x},t), t+1\right) - u(\boldsymbol{x},t), & \boldsymbol{x} \in S_t^f, \\ 0, & \boldsymbol{x} \notin S_t^f, \end{cases} \tag{11.3}$$

where $\hat{u}(\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x},t), t+1)$ is the bi-linear interpolation of $u(\,\cdot\,, t+1)$ at $\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x},t)$. We will call equation (11.3) as the $\boldsymbol{v}^f$-*scheme*. Let us also define the $\boldsymbol{v}^b$-*scheme* using the backward optical flow:

$$\partial_{\boldsymbol{v}}^b u(\boldsymbol{x},t) = \begin{cases} u(\boldsymbol{x},t) - \hat{u}\left(\boldsymbol{x} + \boldsymbol{v}^b(\boldsymbol{x},t), t-1\right), & \boldsymbol{x} \in S_t^b, \\ 0, & \boldsymbol{x} \notin S_t^b, \end{cases} \tag{11.4}$$

Note that when the interpolation neighborhood is not completely inside $\Omega$, the convective derivative is zero. This amounts to a Neumann boundary condition on $\partial\Omega$, and as will be discussed in Section 11.5, it causes a boundary condition of the type $\operatorname{div}\boldsymbol{\nabla}\partial_{\boldsymbol{v}} u(\boldsymbol{x},t) = 0$ on the Euler-Lagrange equation (at certain parts of $\partial\Omega$).

In the example of the Figure 11.2, each convective derivative $\partial_v^f u(x,t)$ is represented by a square node attached to the optical flow vector $(v^f(x,t), 1)$. It computes the difference between the values of $u$ at both ends of the arrow, the value at the tip being given by the bi-linear interpolation (linear in this one-dimensional example). The black squares represent known convective derivatives (their stencil is contained in $\widetilde{O} \setminus O$). These act as Dirichlet boundary condition setting the value of $\partial_v u$. Gray squares represent the convective derivatives of pixels $x \notin S_t^f$ which are set to zero in the definition.

### Adjoint of the convective derivative

To define the adjoint of the convective derivative operator, let us first introduce the following expression for the bi-linear interpolation:

$$\hat{u}(\boldsymbol{x}+\boldsymbol{v}^f(\boldsymbol{x},t),t+1)=\sum_{\boldsymbol{y}\in N^f(\boldsymbol{x},t)} w^f(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y},t+1),$$

where $w^f(\boldsymbol{x},\boldsymbol{y})$ are the bi-linear weights for $\boldsymbol{x}+\boldsymbol{v}^f(\boldsymbol{x},t)$. Note that this expression is not limited to the bi-linear interpolation. Any other linear interpolation scheme, *e.g.* bi-cubic interpolation, fits this model with a suitable definition of the interpolation neighborhood and weights.

The adjoint of the forward convective derivative is then given by

$$\partial_{\boldsymbol{v}}^{f*} g(\boldsymbol{x},t)=\mathbf{1}_{S_t^f}(\boldsymbol{x})g(\boldsymbol{x},t)- \sum_{\boldsymbol{y}:\boldsymbol{x}\in N^f(\boldsymbol{y},t-1)} \mathbf{1}_{S_{t-1}^f}(\boldsymbol{y})w^f(\boldsymbol{x},\boldsymbol{y})g(\boldsymbol{y},t-1)\,, \quad (11.5)$$

where for any set $A\subset\Omega$, $\mathbf{1}_A(\boldsymbol{x})$ is the indicator function of $A$, returning the value 1 if $\boldsymbol{x}\in A$, and 0 otherwise.

Using the graphical representation of the operators in Figure 11.2, the convective derivative operator can be thought of as acting on a video (represented by the round nodes) and returning a function represented by the square nodes. Conversely, its adjoint operator, $\partial_v^{f*}$, acts on functions $g$ represented by the square nodes, and returns a function on the round nodes. Its value at a node $(x,t)$ can be thought of as the net outgoing flow through the node. The outgoing flow is given by $g(x,t)$, whereas the incoming flow is given by the summation in (11.5), computed taking into account the values of $g$ at the previous frame, on the pixels whose convective derivative stencil includes $(x,t)$. This is illustrated in Figure 11.2: the blue squares depict the values of $g$ needed to compute $\partial_v^{f*} g(3,3)$, the round node in blue.

### Spatial gradient operator

Since the gradient operates on convective derivatives, we will define two gradient operators, $\boldsymbol{\nabla}^f,\boldsymbol{\nabla}^b : \mathbb{R}^{\Omega^T}\to\mathbb{R}^{2\times\Omega^T}$ in correspondence with the $\boldsymbol{v}^f$ and $\boldsymbol{v}^b$-schemes for the convective derivative (recall that $\Omega^T=\Omega\times\{0,1,\dots,T\}$). Both gradients are implemented using a forward difference scheme (spatially).

Let us consider first the gradient for the $\boldsymbol{v}^f$-scheme. Some care must be taken with its definition. The spatial partial derivative in the direction $\boldsymbol{e}_1 = (1,0)$ of the convective derivative $\partial^f_{\boldsymbol{v}} u$ is given at a point $(\boldsymbol{x},t)$ by $[\boldsymbol{\nabla}^f \partial^f_{\boldsymbol{v}} u(\boldsymbol{x},t)]_1 = \partial^f_{\boldsymbol{v}} u(\boldsymbol{x}+\boldsymbol{e}_1,t) - \partial^f_{\boldsymbol{v}} u(\boldsymbol{x},t)$. Thus, two convective derivatives are needed, $\partial^f_{\boldsymbol{v}} u(\boldsymbol{x},t)$ and $\partial^f_{\boldsymbol{v}} u(\boldsymbol{x}+\boldsymbol{e}_1,t)$. Therefore we will require that both $\boldsymbol{x}$ and $\boldsymbol{x}+\boldsymbol{e}_1$ lie in $S^f_t$. This motivates the definition of the following sets

$$\widetilde{S}^f_{\boldsymbol{e}_1,t} = \{\boldsymbol{x} \in S^f_t : \boldsymbol{x} + \boldsymbol{e}_1 \in S^f_t\}, \tag{11.6}$$

$$\widetilde{S}^f_{\boldsymbol{e}_2,t} = \{\boldsymbol{x} \in S^f_t : \boldsymbol{x} + \boldsymbol{e}_2 \in S^f_t\}, \tag{11.7}$$

where $\boldsymbol{e}_2 = (0,1)$.

For an arbitrary video $q : \Omega \times \{0,1,\dots,T\} \to \mathbb{R}$, the $i$th component of its spatial gradient $\boldsymbol{\nabla}^f q$ is then defined at a point $(\boldsymbol{x},t) \in \Omega \times \{0,1,\dots,T\}$ as

$$[\boldsymbol{\nabla}^f q(\boldsymbol{x},t)]_i = \begin{cases} q(\boldsymbol{x}+\boldsymbol{e}_i,t) - q(\boldsymbol{x},t), & \boldsymbol{x} \in \widetilde{S}^f_{\boldsymbol{e}_i,t}, \\ 0, & \text{otherwise}, \end{cases} \tag{11.8}$$

for $i = 1,2$. This definition of the gradient implements a boundary condition of Neumann type at the spatial boundary of $S^f_t$.

The adjoint operator is given by the negative spatial divergence implemented with backward spatial differenteces. For a vector-valued video $\boldsymbol{g} \in \mathbb{R}^{2\times\Omega^T}$, we will define the spatial divergence by

$$\begin{aligned} \operatorname{div}^f \boldsymbol{g}(\boldsymbol{x},t) = {} & \mathbf{1}_{\widetilde{S}^f_{\boldsymbol{e}_1,t}}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x},t)_1 \\ & - \mathbf{1}_{\widetilde{S}^f_{\boldsymbol{e}_1,t}}(\boldsymbol{x}-\boldsymbol{e}_1) \boldsymbol{g}(\boldsymbol{x}-\boldsymbol{e}_1,t)_1 \\ & + \mathbf{1}_{\widetilde{S}^f_{\boldsymbol{e}_2,t}}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x},t)_2 \\ & - \mathbf{1}_{\widetilde{S}^f_{\boldsymbol{e}_2,t}}(\boldsymbol{x}-\boldsymbol{e}_2) \boldsymbol{g}(\boldsymbol{x}-\boldsymbol{e}_2,t)_2. \end{aligned}$$

Analogously, we define a spatial gradient operator for the $\boldsymbol{v}^b$-scheme, $\boldsymbol{\nabla}^b$, with its corresponding definition domains $\widetilde{S}^b_{\boldsymbol{e}_1,t}$ and $\widetilde{S}^b_{\boldsymbol{e}_2,t}$. $\boldsymbol{\nabla}^b$ is implemented with a forward difference scheme as in Eq. (11.8). The difference between $\boldsymbol{\nabla}^f$ and $\boldsymbol{\nabla}^b$ lies only in their definition domains, $\widetilde{S}^f_{\boldsymbol{e}_i,t}$ and $\widetilde{S}^b_{\boldsymbol{e}_i,t}$ .

In the example of Figure 11.2, the spatial gradient reduces to a forward spatial derivative. It acts on convective derivatives (or functions represented by the square nodes attached to the optical flow vectors) and returns a function represented by the triangular nodes on the spatial edges between convective

derivatives. Each triangle in Figure (11.2) represents a value of $\nabla^f \partial_v^f u$, and therefore a term in the energy. The gray triangles correspond to the spatial gradients that are set to zero in the definition of the operator. They can not be computed since they require a convective derivative not in $S_t^f$. The white triangles depict the set $\widetilde{S}_{x,t}^f$. Conversely the spatial divergence can be thought of as acting on the triangular nodes, and returning a function represented by the squares.

## 11.4 Treatment of occluded pixels

Most optical flow algorithms assign for all pixels an optical flow vector, even if they are occluded in the adjacent frame. These "false correspondences" can be detrimental to the performance of the method, and have to be removed from the energy.

Occlusions are intrinsic to the problem of the optical flow computation. In fact, some optical flow algorithms detect occlusions as part of the estimation of the movement (Alvarez et al. (2007); Ayvaci et al. (2012)). Such algorithms output an occlusion mask together with the optical flow. Many optical flow algorithms however, do not provide occlusion masks. To be able to work with such optical flows, we describe in Appendix D a simple method to detect occlusions. Other occlusion detection methods could be used instead.

In the following we assume that occluded pixels have been detected, either as part of the optical flow algorithm or by a post-processing detection step. We denote by $K_t^f, K_t^b \subset \Omega_t$ the sets of forward and backward occluded pixels.

Let us consider the forward displacement from frame $t$ to frame $t+1$. At a pixel $\boldsymbol{x} \in K_t^f$, the forward optical flow establishes a correspondence between $\boldsymbol{x}$, visible at $t$, and $\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x}, t)$ which is not visible at $t+1$. The convective derivative associated to this correspondence needs to be removed from our energy. We achieve this by setting to zero any spatial partial derivative involving $\partial_{\boldsymbol{v}}^f u(\boldsymbol{x}, t)$. To that aim, we introduce a *forward occlusion tensor* $\boldsymbol{\kappa}^f : \Omega \times \{0, 1, \dots, T\} \to \{0, 1\}^{2\times 2}$ as the following diagonal matrix

$$\boldsymbol{\kappa}^f(\boldsymbol{x}, t) = \begin{bmatrix} \boldsymbol{\kappa}_{\boldsymbol{e}_1}^f(\boldsymbol{x}, t) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\kappa}_{\boldsymbol{e}_2}^f(\boldsymbol{x}, t) \end{bmatrix}.$$

**Figure 11.3:** Effect of the occlusion tensor. The image on the left shows an output without using the occlusion tensor. On the right, the output at the same frame with occlusion handling.

When applying $\boldsymbol{\kappa}^f(\boldsymbol{x},t)$ to $\boldsymbol{\nabla}^f\partial^f_{\boldsymbol{v}}u(\boldsymbol{x},t)$, $\boldsymbol{\kappa}^f_{\boldsymbol{e}_1} : \Omega \times \{0,1,\dots,T\} \to \{0,1\}$ acts on the spatial partial derivatives in the $\boldsymbol{e}_1 = (1,0)$ direction, whereas $\boldsymbol{\kappa}^f_{\boldsymbol{e}_2} : \Omega \times \{0,1,\dots,T\} \to \{0,1\}$ applies to the partial derivative in the direction $\boldsymbol{e}_2 = (0,1)$.

Given $K^f_t$ we define $\boldsymbol{\kappa}^f_{\boldsymbol{e}_1}$ as

$$\boldsymbol{\kappa}^f_{\boldsymbol{e}_1}(\boldsymbol{x},t) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in K^f_t \text{ or } \boldsymbol{x}+\boldsymbol{e}_1 \in K^f_t, \\ 1 & \text{otherwise.} \end{cases} \tag{11.9}$$

A similar definition is given to $\boldsymbol{\kappa}^f_{\boldsymbol{e}_2}$ and the same applies for the *backward occlusion tensor* $\boldsymbol{\kappa}^b$.

Figure 11.3 gives and example of a result obtained with and without the occlusion tensor.

**Remark.** The removal of the convective derivatives associated to occluded pixels could be equally achieved by modifying the sets $S^f_t$ and $S^b_t$, without introducing the occlusion tensors $\boldsymbol{\kappa}^{f/b}$. Recall that $S^f_t$ refers to the domain of the convective derivative. In Section 11 a pixel $(\boldsymbol{x},t)$ is excluded from $S^f_t$ whenever its forward mapping falls out of the frame domain. This situation can indeed be regarded as an occlusion: $(\boldsymbol{x}+\boldsymbol{v}^f(\boldsymbol{x},t),t+1)$ is not visible. Thus, the same treatment could be given for the pixels in $K^{f/b}_t$. In doing so, the treatment of occluded pixels becomes implicit in the definition of the operators $\partial^{f/b}_{\boldsymbol{v}}$ and $\boldsymbol{\nabla}^{f/b}$. For the sake of clarity, we use the occlusion tensors making the occlusion handling explicit in the energy (12.5).

## 11.5 The discrete Euler-Lagrange equation

Energy (11.2) is quadratic, thus the Euler-Lagrange equation is given by the following linear system:

$$\partial_{\boldsymbol{v}}^{*}\mathrm{div}(\boldsymbol{\kappa}(\boldsymbol{x},t)\boldsymbol{\nabla}\partial_{\boldsymbol{v}}u(\boldsymbol{x},t))=0, \qquad (\boldsymbol{x},t)\in O, \tag{11.10}$$

with the boundary conditions specified above (and writing $u = u_0$ at $\widetilde{O}\setminus O$). We solve this equation using the conjugate gradient method.

Let us briefly discuss the stencil that implements the Euler-Lagrange equation and its boundary conditions. For simplicity, we base this discussion on the one dimensional sequence of Figure 11.4, and we assume that there are no occlusions, that is $\kappa(x,t) = 1$ for all $(x,t) \in \Omega$. Figure 11.4 shows the stencil of the Euler-Lagrange equation at $(x = 3, t = 3)$, the pixel shown in blue. Determining $\partial_v^{*}\mathrm{div}\nabla\partial_v u(x,t)$, requires the values of $\mathrm{div}\nabla\partial_v u$ at the four blue squares. For computing these spatial divergences, $\nabla\partial_v u$ is needed at the triangular nodes highlighted in blue. Now, these require the convective derivative to be computed at the red square nodes, resulting the stencil shown by the red circles (in addition to the original $(x,t)$).

If at any of these steps, one of the required quantities falls out of $\Omega$, it is assumed to be zero. Depending on the optical flow, this amounts to setting to zero either $(a)$ $\mathrm{div}\nabla\partial_v u$ and $\nabla\partial_v u$, or $(b)$ only $\nabla\partial_v u$, or $(c)$ only $\mathrm{div}\nabla\partial_v u$. This comes as a consequence of the definition of the adjoint operators $\partial_v^{*}$ and div.

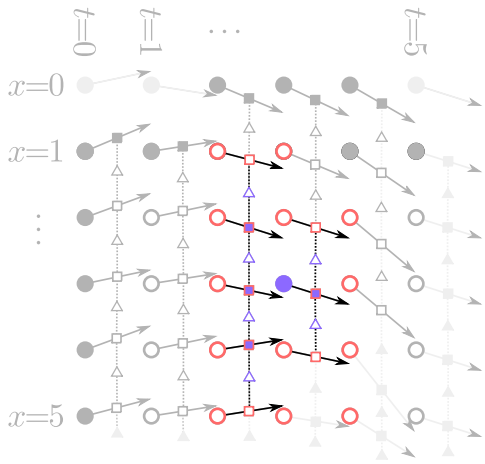



**Figure 11.4:** Example of the stencil of the Euler-Lagrange equation at at ($x = 3, t = 3$), the pixel shown in blue. To compute $\partial_v^* \mathrm{div} \nabla \partial_v u(x,t)$, the values of $\mathrm{div} \nabla \partial_v u$ at the four blue squares are used. These spatial divergences require $\nabla \partial_v u$ at the triangular nodes highlighted in blue. These require the convective derivative to be computed at the red square nodes, resulting the stencil shown by the red circles (in addition to the original $(x,t)$).

Chapter 12

# A deblurring scheme for the convective derivative

In Chapter 11, we presented two different schemes for the discretization of the convective derivative, namely the $\boldsymbol{v}^f$- and $\boldsymbol{v}^b$-schemes. In the present section we comment on the behaviour of these schemes. This discussion will lead us to the derivation of a hybrid scheme that exploits the intrinsic properties of the $\boldsymbol{v}^f$- and $\boldsymbol{v}^b$-schemes. The resulting hybrid scheme allows to handle a much larger number of frames. This scheme is based on the work presented in Facciolo et al. (2011).

## 12.1 A motivating example

For the sake of this discussion, we consider minimizing (11.2) with $\boldsymbol{\kappa}(\boldsymbol{x}, t) = \boldsymbol{I}$ for all $(\boldsymbol{x}, t) \in \widetilde{O}$ (*i.e.* no occlusions):

$$E(u) = \sum_{(\boldsymbol{x},t)\in\widetilde{O}} \|\boldsymbol{\nabla}\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t)\|^2.$$

In a general case, the minimum energy may be non-zero due to incompatible boundary conditions. Let us assume that the boundary conditions are compatible and let $u$ be a minimizer of $E$ with zero energy, *i.e.* $E(u) = 0$.

In this case, we have that

$$\begin{aligned} \partial_{\boldsymbol{v}} u(\boldsymbol{x},t) &= g(t), && \forall (\boldsymbol{x},t) \in \widetilde{O}\,, \\ u(\boldsymbol{x},t) &= u_0(\boldsymbol{x},t), && \forall (\boldsymbol{x},t) \in \widetilde{O} \setminus O\,, \end{aligned} \tag{12.1}$$

where $g(t)$ is the constant illumination change rate at frame $t$ obtained from the boundary conditions. Additionally, we will assume that there is no illumination change and therefore $g(t) = 0$ for all $t$.

Let us now consider a concrete example of a one-lid problem where we fix the first frame $t = 0$ and set it as a Dirichlet boundary condition. In this case Eq. (12.1) can be solved by propagating forward the information at the lid sequentially from one frame to the next. In this context the $\boldsymbol{v}^b$-scheme is an explicit scheme, whereas the $\boldsymbol{v}^f$-scheme is implicit. To see this, let us consider that the problem is one-dimensional with constant optical flow $v^f(x,t) = v_0 \in (0,1)$, and correspondingly $v^b(x,t) = -v_0$.

Let us discuss the effect of using the $\boldsymbol{v}^b$-scheme for solving (12.1). In this case, the following recursive relation between two adjacent frames holds for $0 \le t < T$, $x \in O$:

$$\begin{aligned} u(x,t+1) - \hat{u}(x - v_0, t) = \\ u(x,t) - [v_0 u(x-1,t) + (1-v_0)u(x,t)] = 0. \end{aligned} \tag{12.2}$$

Thus, the values of $u(\cdot, t+1)$ are explicitly determined by applying the averaging operator with coefficients $[1 - v_0, v_0]$ to frame $u(\cdot,t)$. Denoting this operator as $M^b$, we can describe this relation as: $u(\cdot, t+1) = M^b u(\cdot,t)$. It is interesting to look at the frequency response of the averaging filter $M^b$, given by

$$M^b(\omega) = (1 - v_0) + v_0 e^{i\omega}.$$

This frequency response is shown in Fig. 12.1. It has an approximately linear phase for low frequencies with a slope of $v_0$. Thus, the filter shifts $v_0$ pixels the low frequency components of the signal. Note also that there is a significant attenuation of medium and high frequencies. By recursing equation (12.2) we can express the solution at frame $t$ in terms of the first frame (the lid) as $u(\cdot,t) = (M^b)^t u(\cdot,0)$. Therefore, the solution while beeing shifted according to the constant optical flow, becomes increasingly blurry with $t$.

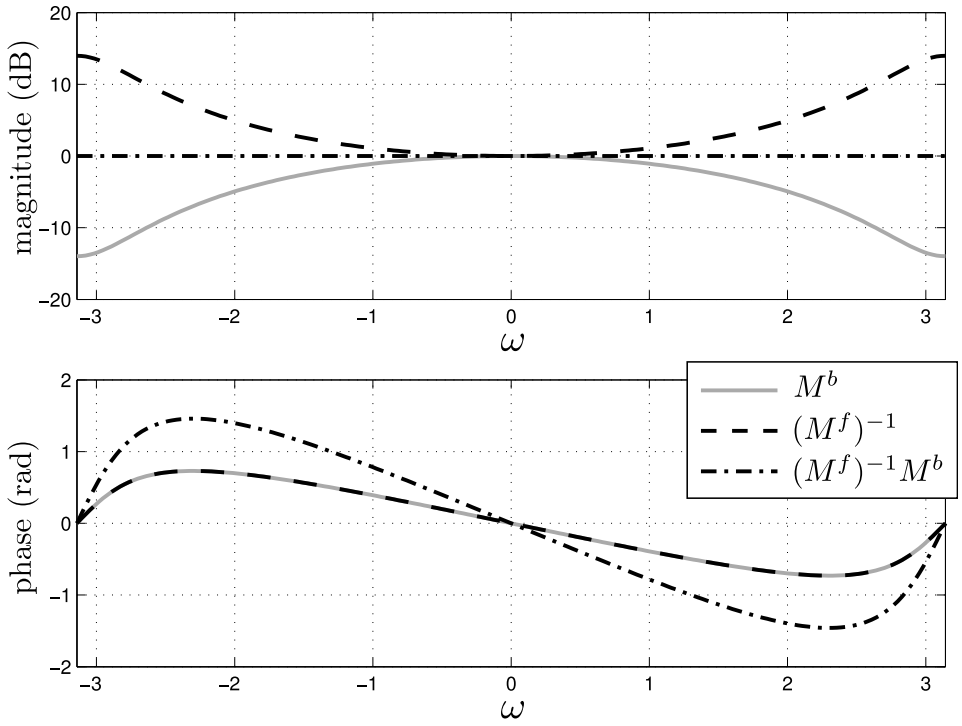



**Figure 12.1:** Analysis of a zero-energy solution for a one-lid problem, with a purely translational motion. Frequency responses of the $M^b$ and $(M^f)^{-1}$ filters, and their composition $(M^f)^{-1}M^b$.

A similar argument for the implicit $\boldsymbol{v}^f$-scheme reveals that for $0 \leq t < T$, $x \in O$ we have

$$0 = \hat{u}(x, t+1) - u(x,t) = \\ [(1-v_0)u(x,t+1) + v_0 u(x+1, t+1)] - u(x,t) \quad (12.3)$$

which can be written as $M^f u(\cdot, t+1) = u(\cdot, t)$, where $M^f$ applies the averaging operator $[v_0, 1-v_0]$ to $u(\cdot, t+1)$. $u(\cdot, t+1)$ is given by the pseudo-inverse of $M^f$ applied to $u(\cdot, t)$. The frequency response of the inverse operator is given by

$$(M^f)^{-1}(w) = \frac{1}{(1-v_0) + v_0 e^{i\omega}} = \frac{1}{\overline{M^b(w)}},$$

where $\overline{M^b(w)}$ denotes the convex conjugate of $M^b(w)$. Figure 12.1 shows the modulus and phase of $(M^f)^{-1}$. The phase is the same as for $M^b$, cor-

responding to a shift of $v_0$ of the low frequency components. However, the modulus is inverted: high frequencies are amplified. Therefore, the repeated application of the pseudo-inverse acts as an inverse smoothing, which enhances the high frequencies in the solution (sharpening) introducing numerical artifacts which accumulate along time.

It is interesting to note that the effects of the $\boldsymbol{v}^b$- and $\boldsymbol{v}^f$-schemes for discretizing the convective derivative are opposite. The $\boldsymbol{v}^b$-scheme introduces blurring in the solution, while the $\boldsymbol{v}^f$-scheme sharpens the solution but also introduces oscillations. This suggests that they can be combined into a hybrid scheme with the hope that their negative effects cancel out.

## 12.2 The DSCD: a mixed scheme

As in Facciolo et al. (2011), we propose a mixed scheme, which we call the *deblurring scheme for the convective derivative* (DSCD for short). The idea of the DSCD is to attain this objective by alternating between the $\boldsymbol{v}^b$- and the $\boldsymbol{v}^f$-schemes. Shortly, if from $t = 0$ to $t = 1$ we apply the $\boldsymbol{v}^f$-scheme, then from frame $t = 1$ to $t = 2$ we apply $\boldsymbol{v}^b$-scheme and so on.

There are two ways to implement the DSCD, depending on whether it starts at frame $t = 0$ with the $\boldsymbol{v}^b$ or the $\boldsymbol{v}^f$-scheme. This determines the way in which the data given at the lid is related to subsequent frames: Either with an explicit averaging filter in the case of starting with the $\boldsymbol{v}^b$-scheme, or with an implicit sharpening filter when starting with the $\boldsymbol{v}^f$-scheme. The one that starts with the $\boldsymbol{v}^f$-scheme does not use the optical flows at odd frames: It will use a $\boldsymbol{v}^f$ step at $t = 0$ with the forward optical flow from $t = 0$ to $t = 1$; then it will use a $\boldsymbol{v}^b$ step at frame $t = 2$ with the backward optical flow from $t = 2$ to $t = 1$, and so on. We call this scheme the *even assignation* of the DSCD (or even DSCD). Alternatively, the *odd assignation* of the DSCD (or odd DSCD) starts with the $\boldsymbol{v}^b$-scheme and only uses the forward and backward optical flows at odd frames.

Let us discuss the resulting schemes when applied to the 1D example with the constant translational optical flow given above. Suppose $t$ is even, and consider the odd DSCD. Then, between frames $t$ and $t + 1$ we use the $\boldsymbol{v}^b$-scheme, and the $\boldsymbol{v}^f$-scheme between $t+1$ and $t+2$. Following the previous discussion, we have that, $u(\cdot, t + 2) = (M^f)^{-1}M^b u(\cdot, t)$. The frequency response of the compound filter $(M^f)^{-1}M^b$ has a flat magnitude: the blurring effect of $M^b$ and sharpening effect of $(M^f)^{-1}$ cancel out. The phase is ap-

proximately linear for low and medium frequencies, but now the slope is $2v_0$, corresponding to the shift between two frames. The same holds for the even DSCD, since two linear filters commute. Thus, if we apply both DSCDs to the one-lid problem at hand, the result obtained should coincide at even frames, but will differ at odd frames. For the odd DSCD, odd frames are computed by a blurring $M^b$ filter of the previous even frame, whereas the even DSCD applies a sharpening $(M^f)^{-1}$ filter.

This behaviour can be appreciated in Figure 12.2. The Figure compares the results of the $\boldsymbol{v}^b$- and $\boldsymbol{v}^f$-schemes together with both DSCDs in a one-lid problem with a purely translational optical flow. The experiment was artificially generated by translating an image with a constant horizontal displacement of $v_0 = [-0.426, 0]$ px/frame. An editing domain is also generated by translating a binary mask with the same optical flow. To test the different schemes we consider a one-lid problem on the given editing domain. In this way we can qualitatively and quantitatively evaluate the ability of the schemes to propagate the first frame. A good propagation should recreate the original sequence inside the editing domain.

As expected, the $\boldsymbol{v}^b$-scheme incrementally blurs the result in the horizontal direction. The results are not shown for the $\boldsymbol{v}^f$-scheme, since it rapidly amplifies high horizontal frequencies, destroying the signal within just a couple of frames. The results for both DSCD schemes perform a better propagation throughout the whole sequence (41 frames). To better appreciate the differences between the even and the odd DSCD, we show the result for three pairs of frames formed by an even frame and the subsequent odd frame. Both DSCDs yield very similar results at even frames, while they differ at odd frames. In particular the even DSCD shows high frequency artifacts due to the application of the sharpening filter $(M^f)^{-1}$.

The plots in Figure 12.3 show the root mean square error (RMSE) between the sequences obtained by each of the four propagation schemes and the original sequence. The RMSE rapidly grows with the iterations for both the $\boldsymbol{v}^f$- and $\boldsymbol{v}^b$-schemes. The RMSE curves for both DSCD schemes grow considerably slower. The even DSCD shows a high RMSE at odd frames due to high frequency artifacts. Notice that both DSCDs present a very similar RMSE at even frames. The lowest RMSE is attained by the odd DSCD, in accordance to the results shown in Figure 12.2.

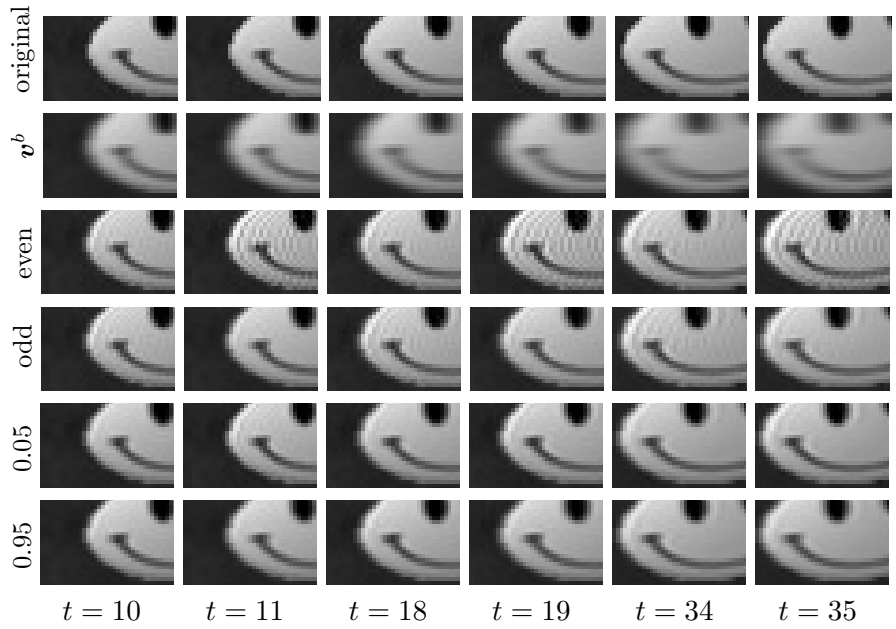



**Figure 12.2:** Results obtained for a synthetic one-lid problem with a constant translation of $\boldsymbol{v_0} = [-0.425, 0]$ px/frame, shown at frames $t = 10, 11, 18, 19, 34, 35$ (columns from left to right). First four rows from top to bottom: original sequence (ground truth), $\boldsymbol{v}^b$-scheme, even DSCD, odd DSCD. The $\boldsymbol{v}^f$-scheme is not shown since it destroys the signal after just a couple of frames. The last two rows show results obtained with a combination of the even and odd DSCD explained in Section 12.4. Fifth row: $\beta = 0.05$ in Eq. (12.6), *i.e.* a combination of 95% of even DSCD with 5% of odd DSCD in the energy. Sixth row: same as fifth row but with $\beta = 0.95$. These correspond to the way we propose for the DSCD to be used in an energy. Note that the high frequency artifacts are greatly diminished, specially with $\beta = 0.95$.

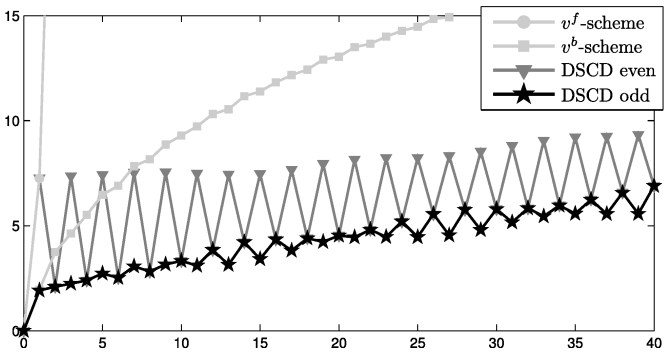



**Figure 12.3:** Evolution of the root mean square error w.r.t. the ground truth corresponding to the synthetic problem with a constant translation of $\boldsymbol{v_0} = [-0.425, 0]$ px/frame. The sequence has 40 frames. Some frames of the result are shown in Figure 12.2. The RMSE curves corresponding to the results shown in the last two rows of Figure 12.2, have been omitted to avoid cluttering in the graphs. They are shown later in 12.7.

## 12.3 Scope and limitations of the previous analysis

The preceding discussion holds only for zero-energy solutions with a purely translational flow. However, it provides some insights on the behaviour of the DSCD on more general cases.

### One-lid problems

In a real one-lid problem, the boundary conditions may not be compatible, and the Euler-Lagrange equation cannot be reduced to solving the PDE $\boldsymbol{\nabla}\partial_{\boldsymbol{v}}u(\boldsymbol{x},t)=0$. Furthermore the cancelation of the blurring and sharpening is exact only when the flow is translational. Still, a similar behavior of the schemes can be observed for one-lid problems with an approximately fronto-parallel movement, even if they do not correspond to a zero energy solution with a pure translational flow. Both DSCDs are able to propagate the information for a much larger number of frames, with much less noticeable artifacts.

This is illustrated in Figure 12.4, for an example with a real sequence. The editing surface is mainly translated, but also suffers mild tilts and zooms. The optical flow was computed using the algorithm of Brox and Malik (2011). The results shown were obtained with the energy in Eq. (11.2) for a one-lid problem using the $\boldsymbol{v}^b$- and $\boldsymbol{v}^f$-schemes, as well as for the DSCD. The results of the DSCD were obtained using an energy which combines the even and odd DSCDs (Eq. (12.6), setting $\beta = 0.02$, which correspond to a 98% of even DSCD and a 2% of odd) and will be explained in Section 12.4. Let us just say for the moment that the variational combination of even and odd DSCD schemes greatly attenuates the formation of high frequency artifacts in the solution. The solution obtained with the $\boldsymbol{v}^b$-scheme progressively blurs the data on the lid (second row in Figure 12.4). On the other hand, the $\boldsymbol{v}^f$-scheme gradually amplifies high spatial frequencies (third row in Figure 12.4). The DSCD allows to propagate the logo throughout the sequence without appreciable artifacts (fourth row in Figure 12.4). The last two rows in Figure 12.4 will be explained below.

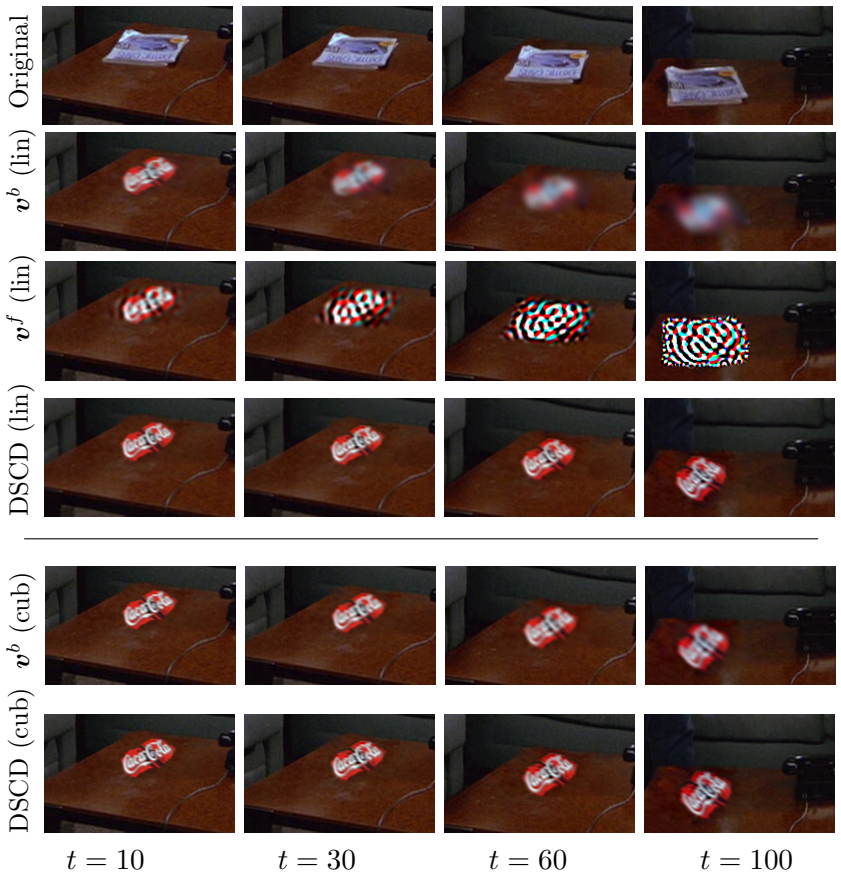



**Figure 12.4:** Results obtained for a real one-lid problem. The lid is placed at the first frame $t = 0$. The first row shows the original sequence before the editing. The following three rows show the results obtained using bi-linear interpolation to implement the convective derivative schemes. From top to bottom show: Results obtained with $\boldsymbol{v}^b$-scheme (explicit), $\boldsymbol{v}^f$-scheme (implicit), and DSCD hybrid scheme. The last two rows were computed using bi-cubic interpolation. From top to bottom: $\boldsymbol{v}^b$-scheme (explicit) and DSCD hybrid scheme. Both DSCD results were obtained with the variational combination of the even and odd DSCD of Eq. (12.6) with $\beta = 0.02$ (thus, mainly even DSCD), explained in Section 12.4.

### Two-lid problems

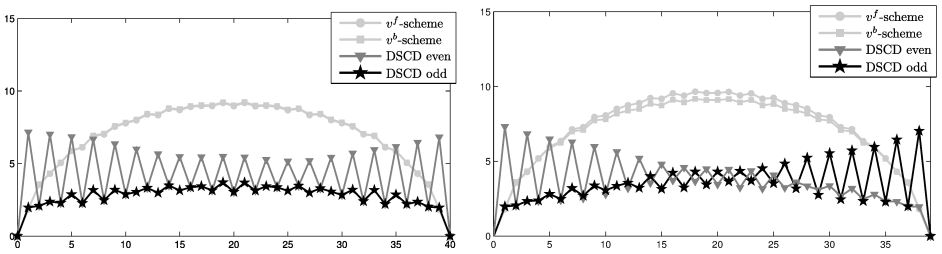
The situation for a two-lid setting is different, specially for the $\boldsymbol{v}^f$- and $\boldsymbol{v}^b$-schemes. For these schemes, the boundary conditions at both lids are not likely to be compatible. Consider for instance the $\boldsymbol{v}^b$-scheme. To have compatible lids, the second lid should not only be a transformed version of the first lid according to the motion, but also would have to account for the blurring caused scheme itself. Of course in a practical editing application, both lids are non-blurred, thus the minimum is not a zero energy solution and the analysis of the preceding section does not apply. Intuitively speaking, there are two opposing effects competing: the first lid should be blurred as time increases, whereas the second lid at $t = T$ should be sharpened backwards in time (and vice versa for the $\boldsymbol{v}^f$-scheme). We have observed empirically that the averaging effect dominates, with results presenting considerable blur at intermediate frames away from the lids. This has been observed in the literature as well Kokaram et al. (2005); Shiratori et al. (2006).

Figure 12.5 depicts RMSE curves for the synthetic translational example using two lids, resulting in non-compatible boundary data. We have to distinguish two cases depending on the whether the number of frames in the sequence is even or odd, because the behaviour of both DSCDs depend on the parity of the number of frames. Figure 12.5a shows the RMSE curves obtained with 41 frames (the first lid is frame $t = 0$ whereas the second lid is frame $t = 40$). In Figure 12.5b we remove one frame and place the second lid at frame $t = 39$.

The behaviour of the $\boldsymbol{v}^b$ and $\boldsymbol{v}^f$ schemes is roughly the same, and this is independent on the number of frames: their RMSE curves are symmetrical, with increasing error as we go away from the lids.

However, it can also be seen that there is a change in the behaviour of the even and odd DSCDs. For the sequence with 41 frames, the even and the odd DSCDs coincide at even frames, and at odd frames the even DSCD has a higher RMSE, associated with high frequency artifacts introduced by the sharpening step.

On the other hand, when the total number or frames is even (Figure 12.5b), for the first 10 frames in the sequence, the behaviour of even and odd DSCDs is similar to the one shown for an odd number of frames. But the situation is inverted by the end of the sequence towards the second lid: the DSCDs coincide on odd frames, and it is odd DSCD the one with high RSME at even frames.




**(a)** Two-lids case, odd number of frames. **(b)** Two-lids case, even number of frames.

**Figure 12.5:** Evolution of the root mean square error w.r.t. the ground truth corresponding to two synthetic problems with a constant translation. In (a) the sequence has 41 frames with an optical flow of $v_0 = [-0.425, 0]$ px/frame. In (b) the sequence has 40 frames with an optical flow of $v_0 = [-0.436, 0]$ px/frame.

The reasons for this become clear when we write the DSCD energies in terms of the $M^b$ and $M^f$ interpolation filters:

$$E_{\boldsymbol{\kappa}}^{\text{odd}}(u) = \|M^b u_0(\cdot\,,0) - u(\cdot\,,1)\|^2 + \|u(\cdot\,,1) - M^f u(\cdot\,,2)\|^2 + \ldots + \begin{cases} \|M^b u(\cdot\,,T-1) - u_0(\cdot\,,T)\|^2 & \text{if } T+1 \text{ is odd,} \\ \|u(\cdot\,,T-1) - M^f u_0(\cdot\,,T)\|^2 & \text{if } T+1 \text{ is even.} \end{cases}$$

$$E_{\boldsymbol{\kappa}}^{\text{even}}(u) = \|u_0(\cdot\,,0) - M^f u(\cdot\,,1)\|^2 + \|M^b u(\cdot\,,1) - u(\cdot\,,2)\|^2 + \ldots + \begin{cases} \|u(\cdot\,,T-1) - M^f u_0(\cdot\,,T)\|^2 & \text{if } T+1 \text{ is odd,} \\ \|M^b u(\cdot\,,T-1) - u_0(\cdot\,,T)\|^2 & \text{if } T+1 \text{ is even.} \end{cases}$$

The first term determines the nature of the connection with the first lid. The odd DSCD enforces an explicit (averaging) link between the lid and $u(\cdot\,,1)$. For the even DSCD, on the other hand, the link is implicit, responsible for the high RMSE erros at odd frames in close to the first lid.

The nature of the connection to the last lid depends on the parity of the total number of frames $T+1$. For an odd number of frames the connection to the last lid is of the same type as the connection to the first lid. The odd DSCD is linked to the second lid via a $\boldsymbol{v}^f$ step, which when seen in the backwards direction of propagation, is an explicit (and thus averaging) step. The even DSCD, is linked to the last lid through a term enforcing an implicit sharpening relation between $u(\cdot\,,T-1)$ and the lid, when seen in the backwards direction. This the reason for the symmetric behaviour of both DSCD curves in Figure 12.5a.

However, when the $T+1$ is even, the situation is reversed: the odd DSCD establishes an implicit link to the second lid, whereas the even DSCD is linked explicitly. This explains the exchange in the behaviour of even and odd DSCDs in Figure 12.5b.

### Zooms

Let us discuss the case of sequences with significant zooms. Consider for example a sequence consisting of a close-up on the edited object: the resolution of the editing should increase with each frame. In a one-lid the scheme will propagate whatever information is available at the lid. Thus if the lid is placed on the first frame (when the object is farther away), the scheme will propagate a low resolution version of it. Indeed, in this case the problem becomes one of super-resolution and the model will do as good as the interpolation scheme used. One can avoid this problem by performing the editing on the last high resolution frame and setting it as the lid. In Section 13.3 we discuss further this issue in a more complex example.

### Higher order interpolation

The different schemes for the convective derivative can be implemented using higher orders of interpolation to estimate $u$ at subpixel positions in Eqs. (11.3) and (11.4) (and similarly for both DSCD schemes). As an example, we have computed the outputs for the $\boldsymbol{v}^b$-scheme and the DSCD using bi-cubic interpolation. The results are shown in Figure 12.4 for the one-lid setting. As before, the results of the DSCD correspond to energy (12.6), which combines the even and odd DSCD (we set $\beta = 0.02$, thus a predominantly even DSCD). The higher order interpolation reduces the rate at which the $\boldsymbol{v}^b$-scheme blurs the frames, but eventually the blurring becomes apparent (as also noted in Kokaram et al. (2005); Shiratori et al. (2006)). The result obtained is considerably better than the one for the $\boldsymbol{v}^b$-scheme with bi-linear interpolation, but it is still blurrier than the results obtained with the bi-linear DSCD schemes. As with the bi-linear interpolation, the bi-cubic $\boldsymbol{v}^f$-scheme completely destroys the signal and its result is omitted. The bi-cubic DSCD behaves very similar to the bi-linear one. The reason for this is that the motion is mostly fronto-parallel. Thus, by alternating between the $\boldsymbol{v}^b$- and $\boldsymbol{v}^f$-schemes, the DSCD approximately compensates for the blurring caused by the low-order bi-linear interpolation. For sequences with significant zooms in which the lid is placed at a low resolution frame

(as discussed previously), a higher order interpolation scheme yields better results. An example is given in Chapter 13, Figure 13.12. This improvement comes at the expense of a greater computation cost (*e.g.* the bi-cubic interpolation uses a 16 point stencil). For most of our experiments we found good results using the bi-linear DSCD, without the need to consider higher order interpolators.

## 12.4 The discrete energy with the DSCD

In this section we present our final discrete energy. For that aim, we first introduce the operators implementing the DSCD schemes for the convective derivative.

Let us consider first the odd assignation and define the following hybrid operator for the convective derivative:

$$
\begin{aligned}
h_{\boldsymbol{v}}^{\text{odd}} u(\boldsymbol{x},t) &= \begin{cases} \partial_{\boldsymbol{v}}^{f} u(\boldsymbol{x},t), & t \text{ odd}, \\ \partial_{\boldsymbol{v}}^{b} u(\boldsymbol{x},t+1), & t \text{ even}, \end{cases} \\
&= \begin{cases} \hat{u}(\boldsymbol{x}+\boldsymbol{v}^{f}(\boldsymbol{x},t),t+1) - u(\boldsymbol{x},t), & t \text{ odd and} \\ & \boldsymbol{x} \in S_t^f, \\ u(\boldsymbol{x},t+1) - \hat{u}(\boldsymbol{x}+\boldsymbol{v}^{b}(\boldsymbol{x},t+1),t), & t \text{ even and} \\ & \boldsymbol{x} \in S_{t+1}^b, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned} \tag{12.4}
$$

This operator computes the convective derivatives corresponding to the forward and backward optical flows of the odd frames. Note that the backward derivatives are shifted: $\partial_{\boldsymbol{v}}^{b} u(\boldsymbol{x},t+1)$ is assigned to location $(\boldsymbol{x},t)$. For this reason we do not consider $h_{\boldsymbol{v}}^{\text{odd}}$ as a discretization of the convective derivative, and use the notation $h_{\boldsymbol{v}}$ instead of $\partial_{\boldsymbol{v}}$. The $h$ here stands for *hybrid*.

As in Section 11.3, we define a corresponding spatial gradient $\boldsymbol{\nabla}^{\text{odd}}$ which takes into account the domain where $h_{\boldsymbol{v}}^{\text{odd}}$ can be computed. For the definition of the gradient, we define the sets $\widetilde{S}_{\boldsymbol{e}_i,t}^{\text{odd}}$ for each frame $t$. Recall that these sets contain the locations $\boldsymbol{x}$ where both convective derivatives needed to compute the partial derivative in the direction $\boldsymbol{e}_i$ are computable. Due to the definition of $h_{\boldsymbol{v}}^{\text{odd}}$ it can be seen that if $t$ is odd, $\widetilde{S}_{\boldsymbol{e}_i,t}^{\text{odd}} = \widetilde{S}_t^f$, and if $t$ even, $\widetilde{S}_{\boldsymbol{e}_i,t}^{\text{odd}} = \widetilde{S}_{t+1}^b$. The diagram in Figure 12.6 show how the convective derivatives and their gradients are taken.

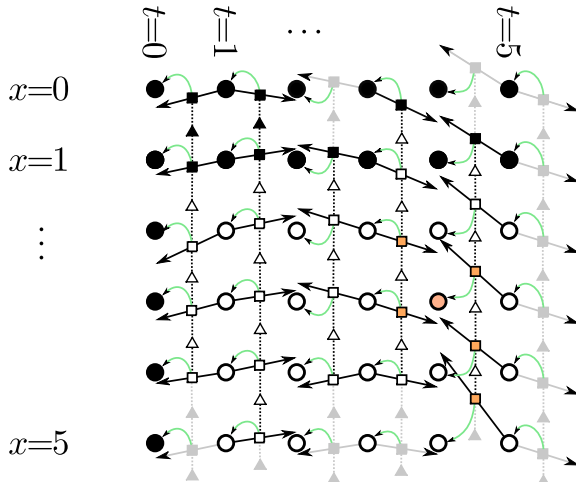



**Figure 12.6:** The hybrid scheme $h_{\boldsymbol{v}}^{\text{odd}}$ for the convective derivative using the forward and backward optical flow of odd frames, for the same one dimensional image sequence shown in Fig. 11.2. The same graphical conventions are used. The green arrows specify to which pixel we assign each convective derivative.

Analogously, we define the corresponding $h_{\boldsymbol{v}}^{\text{even}}$ implementing the even-assignation DSCD (using the forward and backward optical flows at even frames), and its associated spatial gradient $\boldsymbol{\nabla}^{\text{even}}$.

### The energy with the DSCD

Based on the operator implementing the odd DSCD we define the following energy

$$E_{\boldsymbol{\kappa}}^{\text{odd}}(u) = \sum_{(\boldsymbol{x},t)\in\widetilde{O}} \|\boldsymbol{\kappa}^{\text{odd}}(\boldsymbol{x},t)\boldsymbol{\nabla}^{\text{odd}} h_{\boldsymbol{v}}^{\text{odd}} u(\boldsymbol{x},t)\|^2. \tag{12.5}$$

Note that we consider an occlusion tensor $\boldsymbol{\kappa}^{\text{odd}} : \Omega \times \{0,1,\ldots,T\} \to \{0,1\}^{2\times 2}$ for the forward and backward optical flows at odd frames, defined in an analogous way to the forward and backward occlusion tensors in Section 11.4. Similarly we define an energy $E_{\boldsymbol{\kappa}}^{\text{even}}$, corresponding to the hybrid operator $h_{\boldsymbol{v}}^{\text{even}}$ implementing the even-assignation DSCD (using the forward and backward optical flows at even frames).

The proposed energy for using the DSCD considers both the odd and even

assignations, and reads

$$E_\beta(u) = \beta\, E_{\boldsymbol{\kappa}}^{\text{odd}}(u) + (1-\beta)\, E_{\boldsymbol{\kappa}}^{\text{even}}(u), \tag{12.6}$$

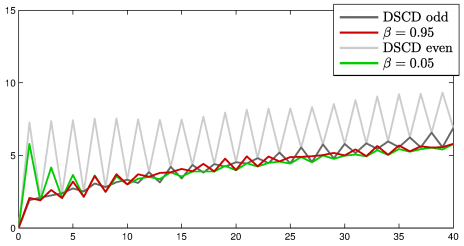
where $\beta \in [0,1]$ is a weighting coefficient. A value of $\beta = 1$ yields the odd-assignation of the DSCD, whereas $\beta = 0$ corresponds to the even-assignation.

Both versions of the DSCD exhibit a comparable behaviour. In general, a $\boldsymbol{v}^f$ step permits to recover the frequencies smoothed by the previous $\boldsymbol{v}^b$ step. However, as shown in Figure 12.2, it may also introduce other high frequencies, which on the long term will build up as high frequency artifacts (particularly for the even DSCD). These can be attenuated by adding to the used version of the DSCD a small component of the other one, which corresponds to values of $\beta \in (0,1)$, either close to 0 or 1. This can be understood in the context of the previous example: we add a slight amount of averaging to the implicit steps, and sharpening to the explicit ones. In practice we found that for the odd-assignation setting a value of $\beta \approx 0.95$ turns out to alleviate the DSCD from high frequency artifacts and without introducing much blurring (correspondingly, $\beta \approx 0.05$ for the even-assignation).

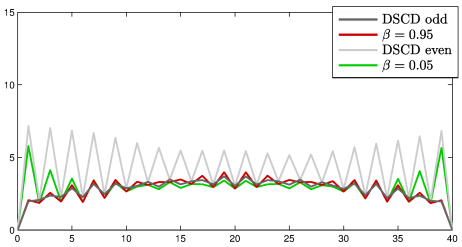
This can be appreciated in the last two rows of Figure 12.2. These results show that the high frequency artifacts of the "pure" DSCDs are removed (at the expense of a mild blurring). This is also confirmed by the corresponding RMSE curves, shown in Figure 12.7. We consider a one-lid and two two-lids problems, with even and odd total number of frames. Note that most of the peaks associated with the sharpening steps are highly reduced (except for the first pair of peaks after a lid).

The use of both DSCD schemes has another advantage. Let us consider, for example, that $\beta = 0$, *i.e.* only the even-assignation DSCD is used. Let us assume as well that $\boldsymbol{\kappa}^{\text{even}}(\boldsymbol{x},t) = \boldsymbol{I}$ for all $(\boldsymbol{x},t) \in \widetilde{O}$. Pixels on odd frames are included in the energy only if they form part of the interpolation stencil of a pixel in an adjacent even frame. Depending on the optical flow, it may occur that some pixels in $O$ only appear in the energy with a small or even zero weight. This causes the problem to be ill-conditioned. On the other hand, with $\beta \in (0,1)$ and assuming that $\boldsymbol{\kappa}^{\text{odd}}(\boldsymbol{x},t) = \boldsymbol{I}$, every pixel in $O$ is "connected" to the rest by its own forward and backward optical flows, which results in a better conditioned system.

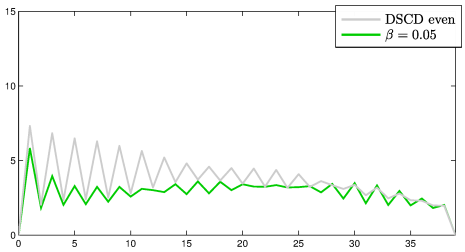
In the case of occlusions, the occlusion tensors remove terms from the energy by setting them to zero (see Section 11.4). This might cause the system to




**(a)** One-lid case.




**(b)** Two-lids case, odd number of frames.




**(c)** Two-lids case, even number of frames.

**Figure 12.7:** Evolution of the root mean square error w.r.t. the ground truth corresponding to two synthetic problems with a constant translation. In this plots we compare the "pure" even and odd DSCDs with the combination of both according to energy $E_\beta$ in Eq. 12.6. We consider $\beta = 0.05$ (95% of even with 5% of odd) and $\beta = 0.95$ (5% of even with 95% of odd). In plots (a) and (b) the sequence has 41 frames with an optical flow of $v_0 = [-0.425, 0]$ px/frame. In (c) the sequence has 40 frames with an optical flow of $v_0 = [-0.436, 0]$ px/frame. Here we only show the graphs for the even DSCD and to $\beta = 0.05$. The ones corresponding to the odd DSCD and to $\beta = 0.95$ correspond to a specular symmetry of the ones showed.

become ill-conditioned, even with $\beta \in (0, 1)$. For this reason we add a spatial regularization term. The resulting energy reads

$$E_{\beta,\lambda}(u) = \frac{1}{2} E_\beta(u) + \frac{\lambda}{p} \sum_{(\boldsymbol{x},t)\in\widetilde{O}} \|\boldsymbol{\nabla} u(\boldsymbol{x}, t)\|^p, \tag{12.7}$$

where $\lambda \geqslant 0$ and $p = 2$ (the model could be formulated with a total variation regularizer by setting $p = 1$, see Appendix E). If there are no occlusions, no elements in the energy are removed and $\lambda$ can be set to zero. Otherwise, $\lambda$ is set to a small value, so that the smoothing effect of the spatial regularization is only noticeable on "weakly connected" pixels. With the addition of this spatial regularization term, if $p = 2$, the solution is unique as long as there exist at least one Dirichlet boundary condition on $u$ in each frame (assuming

that $O_t \neq \Omega$ for all $t \in \{0, 1, \cdots, T\}$). The problem of existence and uniqueness of solutions both in the continuous and discrete frameworks, and for $p \in \{1, 2\}$, is considered in Sadek et al. (2012) under some assumptions on the optical flow that amount to say that trajectories of points are well defined. A more thorough analysis on the conditioning of the resulting system of equations is an important issue and further study is required.

Other regularization terms could also be used. In particular, in our experiments, we sometimes found better results with the addition of a temporal regularization term given by $\gamma \sum_{(\boldsymbol{x},t)\in\widetilde{O}} \|\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t)\|^p$, as in Bhat et al. (2010, 2007); Facciolo et al. (2011). This term, together with the spatial regularizer, forms a weighted 3D gradient $(\boldsymbol{\nabla}, \left(\frac{\gamma}{\lambda}\right)^{1/p} \partial_{\boldsymbol{v}})$ with the temporal component in the direction of the optical flow. Note that this term enforces brightness constancy. Therefore, if used in a sequence with significant illumination changes, it should be given a very small weight.

### Implementation

In practice we set $p = 2$ in the regularization terms. The resulting energy is quadratic, and its minimizer is given by the solution of a linear system which we solve using a conjugate gradient solver. In our implementation we used sparse matrices to store the discrete operators described previously. For example, $\partial_v^f$ can be stored as an $N \times N$ sparse matrix, where $N$ is the number of pixels in $\widetilde{O}$. For each row in the matrix, five values have to be stored: the four bi-linear interpolation weights and $-1$ at the diagonal (see Eq. (11.3)). Storing the operators as sparse matrices, greatly simplifies the computation of the adjoint operators, which can be computed by a simple matrix transpose operation (*e.g.* the matrix associated to $\partial_v^{f^*}$ is the transpose of $\partial_v^f$). This is important since a naive computation of $\partial_v^{f^*}$ following Eq. (11.5) is quadratic in the number of pixels of a frame in the video.

On the other hand, storing the whole operators, is quite demanding in terms of memory. It would be interesting to use an ad-hoc solver exploiting the structure of the problem to lower the memory requirements, for example by working on a frame by frame basis.

Chapter 13

# Experimental results

In this section we present some experimental results showing the behaviour of the presented model in practice. We also discuss the limitations of the method and how to address them. We consider the application of replacing the texture of an objects' surface throughout a video sequence. We distinguish the two application settings discussed in Section 10.2: the *one-lid* and *two-lid* settings. These differ only in the choice of the boundary conditions (see Section 10.2). Let us first describe some elements of the experimental setup.

**Color videos.** In our experiments, a color video $\mathbf{u} : \Omega \times \{0, 1, \dots, T\} \to \mathbb{R}^3$ is treated channel by channel, each as an independent scalar video. This amounts to the minimization of the following energy

$$E^{\text{color}}_{\beta,\lambda}(\mathbf{u}) = \sum_{i=1}^{3} E_{\beta,\lambda}(u_i), \tag{13.1}$$

where $E_{\beta,\lambda}$ is defined for a scalar video in Eq. (12.7) and $u_i$, $i = 1, 2, 3$ are the color channels. Note that, although the processing of each channel is done independently, the same optical flow (and thus the same operators) is used for all channels. We use the *RGB* color space, but any other color space could be used as well.

**Optical flow.** In all the sequences used, we impose no restriction on the movement of the camera nor the movement of the objects in the scene. For

the computation of the optical flow, except when otherwise specified, we use the algorithm described in Sun et al. (2010) and we use the code provided by the authors through the webpage Sun (2010). We also use the default parameters provided with the code.

**Parameters of the model.** The results are obtained by minimizing the energy in Eq. (13.1). Except when otherwise stated, we use $\beta = 0.05$, $\lambda = 0$ when no occlusions occur and $\lambda = 0.02$ otherwise. The minimization is done with the conjugate gradient algorithm. combination of odd and even DSCDs with $\beta = 0.05$ removes most high frequency artifacts caused by the sharpening steps. However, for some frames in certain sequences, we have noticed that some high frequency artifacts remain mildly apparent. For that, we apply a linear filter to the output sequence removing very high frequencies.

**Editing domain.** In a practical editing application, it is important to automate the computation of the editing domain. In the present context, this amounts to an approximate tracking of the portion of the surface that the user wants to edit. In the experiments shown in this section we used different approaches for computing the editing domain, to highlight the flexibility of the model on this issue. For the experiments shown in Figures 13.2, 13.10, 13.12 the editing domain was determined by tracking the edited surface. In Figures 13.1, 13.7 and 13.8, the edited surface has been tracked as well, then manually distorted to simulate big tracking errors (although the errors simulated never miss a portion of the edited surface). Finally, for the rest of the experiments we used a big rectangular domain.

For tracking the edited surface any tracking algorithm can be used. Since we are given an optical flow, the problem of tracking a certain object amounts to propagating a binary mask of the object, specified at a lid (or the lids), along optical flow trajectories. In our experiments we performed this propagation using the proposed model. The output corresponds to a mask tracking the edited surface along the video, which gives the editing domain. Note that for the particular case of a one-lid propagation of a binary mask, the minimum of the proposed energy can be computed efficiently by a frame by frame propagation. The reason is that this problem has compatible boundary conditions, thus the minimum is attained with zero energy. In a two-lid setting (for instance in the presence of an occlusion) the solution can be approximated by two frame by frame propagations, the first propagating

the first lid forward, and the second, the last lid backward. These can be combined by a point-wise maximum.

In what follows we first present four experiments for the one-lid setting, then we present three more experiments for the two-lid setting. We then comment briefly on the behaviour of different optical flow algorithms in the context of this work by experimenting with several of them proposed in the literature. Finally we discuss some practical limitations of the proposed model and we offer ways to overcome some of them. For every experiment below we show the original video sequence, the input to our model where the editing domain has been painted in red indicating the absence of data, and the output video after minimizing (13.1). Due to the difficulty of actually showing all frames processed, we only show a few snapshots from the processed sequence. For the full video sequences used and the results obtained, we refer the reader to the following webpage Sadek et al. (2012).

## 13.1 One-lid setting

In this setting the editing is performed on the first frame. The edited first frame is then set as a Dirichlet boundary condition and, by minimizing energy (13.1), we obtain the output video with the editing propagated along the remaining frames. We present four experiments.

In the first one, the video contains a newspaper placed on a table. After several frames, a moving light starts illuminating the newspaper and a shadow is cast by a moving person. We replace the newspaper in the first frame of the sequence by a logo and we minimize (13.1). The result shown in Figure 13.1 demonstrates how our model handles this complex illumination change. The total length of the sequence is 106 frames with around $5 \cdot 10^5$ variables inside the editing domain.

The second experiment involves a video of a newspaper placed on a table and the light in the room gets dimmed until its off. This causes a considerable large and fast global illumination change. In particular, notice that due to the change in the illumination in the room, there is a change in the dominant color: The colors shift towards blue as the light gets dimmed. We place a poster on top of the newspaper and minimize energy (13.1). Figure 13.2 shows the result. It can be seen that this large and sudden illumination change is handled by our model. The total number of frames

**Figure 13.1:** The first row shows the original sequence. A person is moving casting a shadow on the table while a light reflection is also cast on the table. The second row shows the input sequence where the editing domain has been marked in red. The third row shows the output of our method. From left to right, the frames shown correspond to $t = 0, 25, 50, 75$ and 100 respectively.

of the sequence processed in this experiment is 30 with approximately $9 \cdot 10^5$ variables inside the editing domain.

For comparison purposes, we show the result obtained with the model presented in the preliminary conference version of this work Facciolo et al. (2011), which uses a temporal consistency model based on the brightness constancy assumption. Therefore, to account for illumination changes, the authors in Facciolo et al. (2011) proceed in two steps (similar to Bhat et al. (2007)). The first step propagates the information from the lid using the brightness constancy model and the second step solves a Poisson editing problem for each frame which takes care of the spatial consistency of the editing. For the first step, we show the sequence obtained in the fourth row of Figure 13.2. The inability of the brightness constancy model to adapt to the illumination change in the scene causes spatial discontinuities around the editing domain. To remove them the second step is performed. The gradient of the first step's result is used as guiding vector field for the Poisson problems. This second step is aimed to adapt each edited frame to its spatial context by means of the boundary conditions of the Poisson equation, integrating the gradients of the propagated first lid. Thus, we can interpret this two-step procedure as an implementation of the gradient-constancy assumption. Solving a Poisson problem for each frame independently generates a flickering artifact in the resulting sequence. To avoid it, the brightness constancy model is used as a temporal regularizer with a low

**Figure 13.2:** The first row shows the original sequence. A newspaper is filmed while the camera moves and the light in the room is being dimmed. The second row shows the input sequence after editing the first frame and where the editing domain has been marked in red. The third row shows the output of our method. Note how the resulting video accommodates this fast and sudden illumination change along time. For comparison we show results obtained using the two-step procedure from Facciolo et al. (2011). In the fourth row the output of the first step: it is temporally consistent, but not spatially consistent. The fifth row shows the final result after the second step. The spatial discontinuity around the editing domain has been removed, but note how the new GBC model integrates the illumination change better. From left to right, the frames shown correspond to $t = 0, 11, 17, 24$ and 27 respectively.

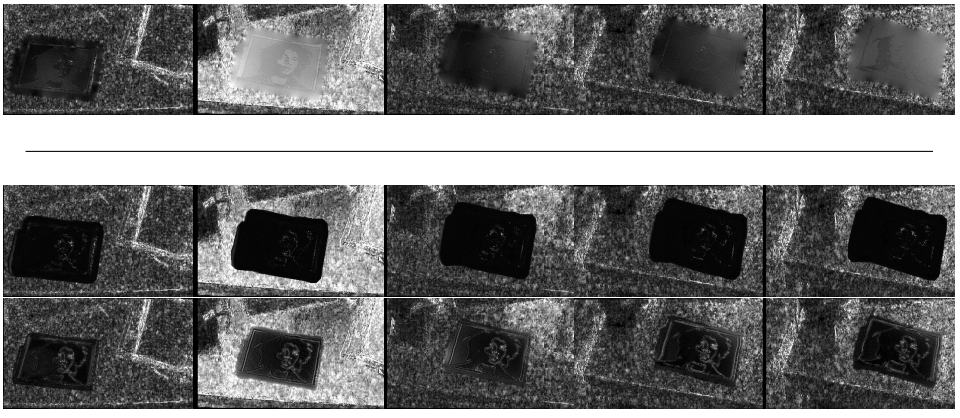


**Figure 13.3:** Magnitude of the illumination change rate (measured as the norm of the convective derivative) for the results shown in Figure 13.2. The first row shows the norm of the convective derivative for the GBC model (third row in 13.2). Last two rows show the norm of the convective derivative for the two-step procedure of Facciolo et al. (2011). The second row corresponds to the output of the first step (fourth row in 13.2), and the third row to the result after the second step (fifth row in 13.2). From left to right, the frames shown correspond to $t = 1, 11, 17, 24$ and 27 respectively (the frames differ from those shown in Figure 13.2).

weight. The resulting sequence of the second step is shown in the fifth row of Figure 13.2.

The two-step procedure in Facciolo et al. (2011) achieves a spatially and temporally consistent editing. However, notice that the result of the GBC model integrates much better the illumination change into the editing (in particular notice that the colors in the editing domain shift gradually towards blue, in accordance with the change in the dominant color of the scene). The main reason for this, is the brightness-constancy-based temporal regularization term added to the second step in Facciolo et al. (2011). Even if it has a low weight, this regularizer causes a slow reaction to fast illumination changes. This can be better appreciated in Figure 13.3, where we show the norm of the convective derivative for the results of Facciolo et al. (2011) and the present model. The norm of the convective derivative measures the illumination change rate. High values (shown in white in Figure 13.3) denote that an illumination change is happening. The result of the GBC model smoothly interpolates the illumination change rate at the boundary of the editing domain, resulting in a better integration of the editing with the surrounding. The result after the first step of Facciolo et al. (2011) has an almost zero illumination change rate, as expected from the brightness constancy model. This is corrected to some extent after the second step, but still the result has a limited capability to adapt for high illumination change rates.

In the third experiment we show a piece of cloth which exhibits a "wave" like movement. We edit the first frame and we minimize (13.1). Figure 13.4 shows the result. The total number of frames processed in this sequence is 20 with approximately $10 \cdot 10^5$ variables inside the editing domain. Note how the deformation of the inserted image follows the deformation of the cloth. For this sequence the optical flow was computed using a multi-scale Horn-Schunck optical flow algorithm Meinhardt-Llopis and Sánchez (2012).

In the last experiment we show for the one-lid setting, we consider a sequence taken from Liu et al. (2008a) which is available through the webpage Liu et al. (2008b). A video of a box and of a cylindrical can is shot. Throughout the sequence, the box and the can are occluding the background before finally interacting when the can occludes the box. We edit the first frame by modifying the textures on both objects and minimize (13.1). Figure 13.5 shows the result. Note that the editing domain includes large portions that do not belong to the edited surfaces. These places should keep the texture they had in the original video. The result shows that the

**Figure 13.4:** The first row shows the original sequence of a cloth exhibiting a "wave" like motion. The second row shows the input sequence after editing the first frame and where the editing domain has been marked in red. The third row shows the output after minimizing (13.1). Note how the editing done accommodates to the movement of the cloth in the resulting video. From left to right, the frames shown correspond to $t = 0, 7, 8, 9, 10$ and 20 respectively.

model has been able to reconstruct the original textures seamlessly. This demonstrates that a precise tracking of the edited surfaces is not required. The total number of frames in the sequence is 13 with $9 \cdot 10^5$ variables inside the editing domain. The results shown are snapshots taken every two or three frames.

## 13.2 Two-lid setting

In this setting the editing is performed on the first and last frames. These are set as Dirichlet boundary conditions and, by minimizing energy (13.1), we obtain the output video with the editing propagated into the remaining frames. Basically, the result is a smooth interpolation between the two edited frames along the trajectories of the optical flow.

The first experiment is shown in Figure 13.6, and uses the same sequence of Figure 13.5. We edit the first frame as in Figure 13.5, by introducing the yellow text on the blue box, and the formula on the can. For the last frame, we only change the color of the formula on the can, from yellow to red. No text is added on the box. In the resulting sequence, the formula and the text move coherently with the can and the box respectively. While moving, the formula changes its color smoothly from yellow to red, and the text in the box gradually vanishes. This demonstrates the interpolation between both lids along the optical flow trajectories. Note that the second "i" in

**Figure 13.5:** The first row shows the original sequence. A box and a cylindrical can are being filmed while the camera moves. The second row shows the input sequence after editing the first frame and where the editing domain has been marked in red. The third row shows the output after minimizing (13.1). From left to right, the frames shown correspond to $t = 0, 2, 5, 9$ and 12 respectively.

the text "rishi" does not vanish as expected. The reason is that this part of the box is occluded in the last frame. In that specific location, the solution behaves as in a one-lid setting, propagating the information from the first frame only.

Let us note that this type of editing, in which both lids are non-consistent, is not the main purpose of this paper. However, the inconsistent editing of the lids in this experiment gives a good insight on the working of the model. It puts in evidence the differences in the behaviour of trajectories reaching both lids and those reaching only one lid. For trajectories that reach both lids the result is a smooth interpolation between the information present at those lids. This can be seen clearly from the smooth transition of the formula's color from yellow to red and the smooth vanishing of the "rish" text. On the other hand, for trajectories that only reach a single lid (for instance due to an occlusion) the problem becomes a one-lid problem and the information will be transported from that lid only. This is what actually happens with the second "i" in the text "rishi": These trajectories do not

**Figure 13.6:** The first row shows the editing done in the first and last frames and the domain has been marked in red. The second row shows the output after minimizing (13.1). Note how the result is a smooth interpolation between the two lids. Let us mention that the original sequence of the experiment is the first row of Figure 13.5. From left to right, the frames shown correspond to $t = 0, 2, 5, 9$ and 12 respectively.

reach the second lid, and therefore the "i" is being transported from the first lid. As a consequence, if the purpose of the editing is to perform a blending between two lids edited in a non-consistent manner, it is imperative that every trajectory in the editing domain reaches both lids.

When the application is to edit an object's surface with a non-changing texture, the editing in both lids has to be consistent. This way, there will be no appreciable differences between one-lid trajectories propagating data from one of the lids, and two-lid trajectories blending data from both lids. In this context consistent editing means consistency with the motion in the scene, and consistency with the overall change in illumination from the first lid to the last. The motion consistency implies that the editing in the second lid corresponds to the warping of the editing in the first lid according to its motion. The consistency with the illumination change implies that the editing in the second lid suffers approximately the same (additive) illumination change as its surrounding with respect to the first lid. In practice, inconsistencies in the editing of both lids are tolerable.

The second experiment, shown in Figure 13.7 depicts a computer screen which gets almost completely occluded, and then dis-occluded, by a person moving in front of it. We replace the image on the screen by an image of corals, both in the first and the last frames. The sequence consists of 20

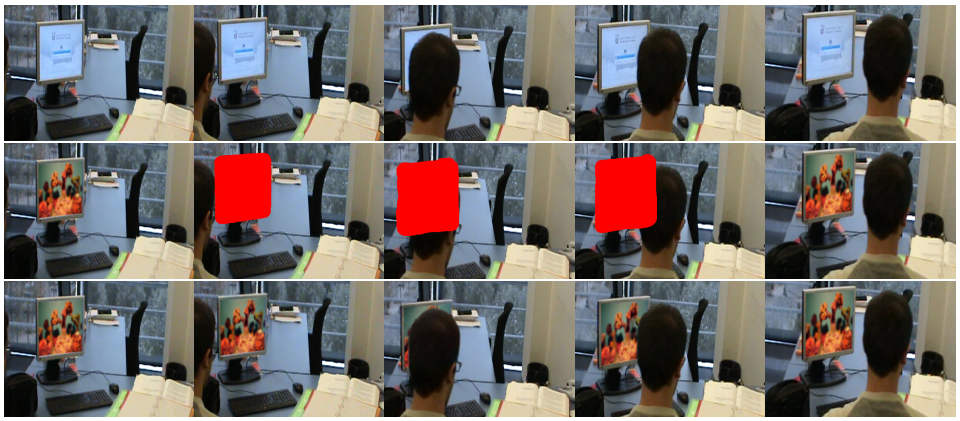


**Figure 13.7:** The first row shows the original sequence. A screen is filmed when a person sitting on a chair moves in front of it. The second row shows the input sequence after editing the first and last frames and where the editing domain has been marked in red. The third row shows the output. From left to right, the frames shown correspond to $t = 0, 4, 12, 16$ and 19 respectively.

frames with more than $4 \cdot 10^5$ variables in the editing domain.

Finally, in the third experiment, we edit a pen-holder lying on a table. The pen-holder gets partially occluded then dis-occluded by a black bag. We edit the curved surface of the pen-holder in both the first and last frames and place the image of the corals on it. The total number of frames of this sequence is 43 with around $10^6$ variables in the editing domain. Figure 13.8 shows the result obtained after minimizing (13.1). For this experiment we found better results with the addition of a temporal regularization term as mentioned at the end of Section 12.4 with a weight $\gamma = 0.02$.

## 13.3 Limitations of the proposed method

In this Section we would like to discuss, with some practical details, the limitations of the proposed method and propose ways to overcome some of them.

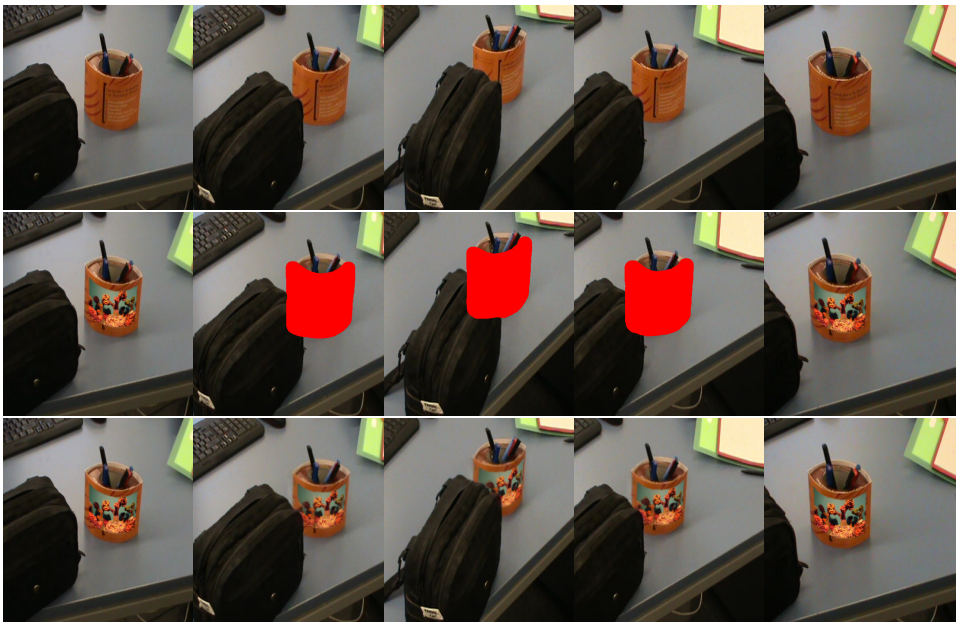


**Figure 13.8:** The first row shows the original sequence. A pen-holder is filmed while it gets occluded by black bag. The second row shows the input sequence after editing the first and last frames and where the editing domain has been marked in red. The third row shows the output after minimizing (13.1). From left to right, the frames shown correspond to $t = 0, 16, 26, 36$ and 42 respectively.

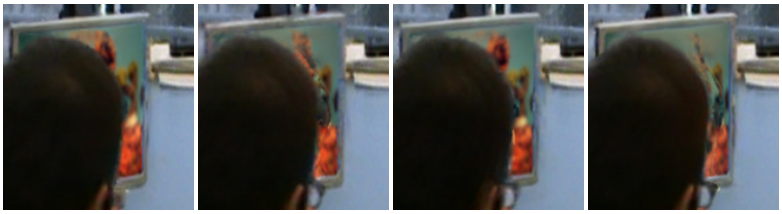


**Figure 13.9:** Experiment showing the behaviour of the model using four different optical flow algorithms. We show only one frame for each result. The complete result sequences can be found in Sadek et al. (2012). From left to right, the images correspond to using the optical flow algorithm presented in Sun et al. (2010), Chambolle and Pock (2011), Ayvaci et al. (2012) and Brox and Malik (2011) respectively.

### Optical flow related

It is well known that optical flow algorithms compute *apparent motion*, not the true motion of the scene. This implies that in some circumstances, as in the presence of moving shadows, the optical flow may not give a correct estimation of the true motion. If the apparent motion and the true motion coincide then using the optical flow gives very good results. If not, relying on apparent motion will likely give rise to visual artifacts. Confronting these situations is a current trend of research in the optical flow community, for example by incorporating gradient-based terms for handling illumination changes, or segmenting the scene to improve the occlusion handling. These improvements will in turn expand the applicability and performance of our algorithm. To highlight the dependence on the optical flow, we show two experiments.

In the first one we test four optical flow algorithms from the literature, applying them on the two-lid editing problem shown in Figure 13.7. We used the following optical flow algorithms: the large displacement optical flow algorithm of Brox and Malik Brox and Malik (2011), the TV-L1 optical flow of Chambolle and Pock Chambolle and Pock (2011), the optical flow with sparse occlusion detection of Ayvaci *et al.* Ayvaci et al. (2012) and finally the layer-based algorithm of Sun *et al.* Sun et al. (2010). For all optical flows, we use the code provided by the authors with the default parameters. For the optical flow algorithms presented in Brox and Malik (2011), Chambolle and Pock (2011) and Sun et al. (2010), since no occlusion masks are given by the algorithms, we use the occlusion detection method described in Appendix D. The optical flow of Ayvaci et al. (2012) provides occlusion masks and we use them (after a dilation of two pixels) to handle occlusions. Figure 13.9 shows one frame from the result of our model using the above discussed optical flow algorithms. We have chosen a frame where the differences between the different algorithms is apparent. Let us recall that the complete set of results can be found in Sadek et al. (2012). For all tested optical flow algorithms, the model behaves similarly and is able to handle the occlusion. The differences in the results are due to the differences in the motion perceived by each optical flow method. For example some optical flows show a "dragging" effect: parts of the screen that are about to be occluded by the head seem to be dragged by the head instead of being occluded. The reason is that the optical flow algorithm has assigned the movement of the head to these parts of the screen. It might also be interesting to test the proposed model with other optical flow methods which

**Figure 13.10:** The first row shows the original sequence. A person wearing a shirt moves by bending down and straightening back up. The second row shows the input sequence after editing the first frame and where the editing domain has been marked in red. The third row shows the output after minimizing (13.1). From left to right, the frames shown correspond to $t = 0, 9, 19, 26, 33, 38, 41$ and 44 respectively.

incorporate the temporal consistency in the computation of the optical flow Salgado and Sánchez (2007); Sun et al. (2012); Volz et al. (2011). It has been reported that considering a larger number of frames improves on the accuracy of the optical flow.

In the second experiment, in Figure 13.10 we show a different kind of artifacts. In this experiment, we edit a part of a shirt being worn by a person by adding to it the "UPF" logo. The person bends over and then straightens back up. While straightening up, the shirt suffers severe deformations. At that point, inaccuracies in the optical flow result in unnatural distortions of the transported texture (second half of the sequence). However, let us note that the model correctly handles the local illumination change caused by the shadow cast by the shirt on itself while bending. The result was obtained by minimizing (13.1). The total number of frames of this sequence is 35 with around $5 \cdot 10^5$ variables in the editing domain.

### Multiple occlusions

So far we have covered the following basic situations:

a) A single occlusion or dis-occlusion occurs. This can be treated as a one-lid setting, for an occlusion the lid is placed at $t = 0$, and for a dis-occlusion the lid is at $t = T$.

b) An occlusion followed by a dis-occlusion occurs. This can be handled by a two-lid setting, where the lids are at $t = 0$ and $t = T$.

Cases with multiple occlusions and dis-occlusions of the edited surface can be handled by temporally splitting the editing domain into temporal segments that fall into the above a) and b) basic cases. The splitting of the editing domain amounts to adding lids at the splitting frames. To illustrate this we consider a sequence taken from Liu et al. (2008a), available online through the webpage Liu et al. (2008b). Figure 13.11 shows an experiment with multiple occlusions of the edited surface.

A hand is moving back and forth, repeatedly dis-occluding and occluding a disk-shaped object in the background. The Figure shows first the result obtained by setting a lid at frame $t = 8$, after the first dis-occlusion. This splits the sequence into two one-lid problems: one from $t = 0$ to $t = 8$, and another from $t = 8$ to $t = 44$ (the last frame). In the second segment there is still one occlusion followed by a dis-occlusion and yet another occlusion towards the end. No trajectories from the lid reach the region that gets dis-occluded, and we can see what sort of artifacts one expects to see in regions where no information from the lid is arriving. As shown in the Figure, this can be corrected by further splitting the sequence and adding an intermediate lid before the last occlusion starts. Let us also mention that the occlusions in this experiment have been detected using the procedure described in Appendix D.

In summary, the rule is to ensure that all trajectories inside the editing domain reach at least one Dirichlet boundary condition.

### Big zooms and tilts

Consider the case of a one-lid setting where the editing has been performed on the first frame and a camera is doing a big zoom-in on the edited object. The resolution of the object increases considerably with time. The method will propagate the low resolution information given at the lid and will not recover a higher resolution version of the propagated information. The resulting propagation will be blurred. However, this could be solved by editing the last high resolution frame and propagate that editing backwards.

**Figure 13.11:** The first row shows the original sequence. A hand is moving back and forth, dis-occluding and occluding a disk-shaped object repeatedly in the background. From left to right, the frames shown correspond to $t = 0, 8, 20, 26, 30, 33, 37$ and 44 respectively. The second row shows the input sequence after editing frame 8 by removing the disk-shaped object and setting it as a lid. It also shows the remaining frames and the editing domain marked in red. The third row shows the output after removing the object in frame 8 and propagating the modification back to frame 0, and forward until frame 44 by minimizing (13.1) in a one-lid setting. Note that the first dis-occlusion (from 0 to 8) is correctly handled as well as the first occlusion (from 8 to 20). But, as a new dis-occlusion starts around frame 26, no information is reaching this area from the lid (frame 8) and it appears as if the hand spills its color into this area. This double occlusion could be handled by editing the frame 33 and setting it as a second lid, as shown in fourth row. In this experiment, the occlusions have been detected using the procedure described in Appendix D.

Consider now the case where a big zoom-out is followed by a big zoom-in. This problem can be solved in a two-lid setting where both lids are at high resolution. Figure 13.12 shows an example where a cloth is being laid down on a table and then taken back to its original position. This simulates the just discussed example with an added tilt transformation as well. We show two results in a one-lid setting with different interpolation schemes (bi-linear and bi-cubic) and one result in a two-lid setting. Notice how in the second half of the sequence, when zooming-in, the one-lid result presents considerable blur with the bi-linear interpolation, particularly at the last frames where the resolution increases significantly. Using the bi-cubic interpolation, the result is much sharper but it still suffers from blur artifacts. Adding a second lid at the last frame solves this issue. Let us also note that a considerable illumination change also occurs in the sequence and the method deals with it seamlessly. The result was obtained

by minimizing (13.1). The total number of frames of this sequence is 53 with around $5 \cdot 10^5$ variables in the editing domain.

Let us note that this limitation comes as a consequence of propagating a texture using correspondences between adjacent frames. Other methods such as Bugeau et al. (2010); Rav-Acha et al. (2008); Sand and Teller (2008) establish a transform between an input texture and all other frames in the sequence. The computation of these transforms is not trivial. In the case of Rav-Acha et al. (2008), the method relies on an accurate video segmentation, keypoint tracking and non-convex optimization in order to compute the mappings. On the other hand, and more related to this work, Bugeau et al. (2010); Sand and Teller (2008) integrate the optical flow to compute a set of trajectories covering the editing domain. This requires dealing with complexities inherent to the explicit management and computation of trajectories. In any case, these methods need post-processing steps in order to deal with illumination changes and filling-in holes that are not covered by the mapping (for instance due to occlusions). In our approach trajectories are dealt with implicitly, illumination correction is intrinsic to the model, and the filling-in of small holes caused by occlusions is taken care by the regularization term in the energy. This is attained by a single convex minimization process.

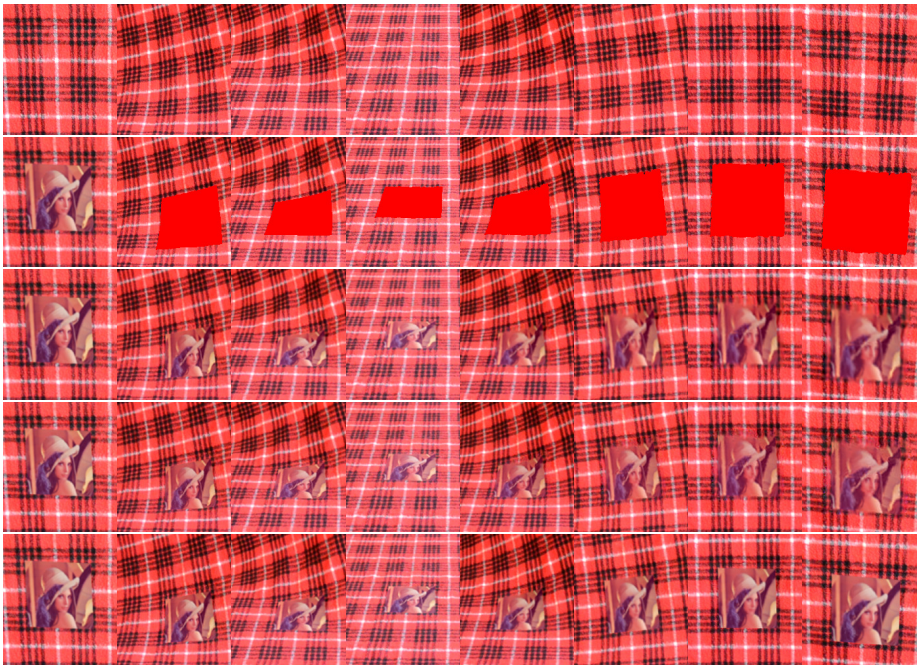


**Figure 13.12:** An experiment simulating the case of a zoom out with tilt followed by a zoom in and tilting back to the original position. The first row shows the original sequence. The second row shows the input sequence after editing the first frame and where the editing domain has been marked in red. The third row shows the output after minimizing Eq. (13.1) in a one-lid setting. The artifacts are clear when coming back from low-resolution to high resolution. The fourth row shows the output when using the bi-cubic interpolation scheme in a one-lid setting as well. It can be noticed that the artifacts diminished but they are still visible. The last row shows the output in a two-lid setting where the artifacts have been dealt with. From left to right, the frames shown correspond to $t = 0, 16, 19, 26, 32, 39, 44$ and 53 respectively.

Chapter 14

# Conclusions and future work

In this work we tackle a rather fundamental mathematical problem in video processing: the propagation gradient-domain information from one or two key-frames through the optical flow (as a dense motion estimate). We applied the model to video editing, where a user edits one or two frames of a video which need to be propagated throughout the remaining frames. We demonstrate that very good results can be obtained via the minimization of a simple quadratic energy, which amounts to solving a linear system of equations.

The proposed model handles fast and abrupt global illumination changes. Smooth spatial variations in the illumination change are also handled by the model implicitly, which allows to obtain a spatial and temporal consistent editing.

Furthermore, the model works with any dense motion estimate (such as the optical flow) and thus is not limited to parametric motion model.

A key element in the proposed method, is the DSCD numerical scheme for the convective derivative which allows to maintain sharpness of the texture during the propagation along motion trajectories for a large number of frames for relatively complicated sequences.

There are several lines of future research to better understand the model and overcome some of its current limitations.

The model requires an estimate of the motion, which prevents its application to problems like video inpainting in which the motion is unknown.

Therefore, it would be interesting to test this model in conjunction with motion interpolation techniques, for its application to video inpainting.

A related issue, is that the results provided by the model depend highly on the quality of the optical flow used, and on the occlusions detected whether by the optical flow algorithm or an by a posterior occlusion detection algorithm. Advances in optical flow computation will help in improving the results provided by the proposed model. But the same applies to other video editing tasks, equally based on optical flow, which are currently included in widely used professional video editing softwares. Such softwares have built-in optical flow computation algorithms which allow a user to interactively correct the errors of an automated algorithm. In this context, the proposed model could be useful.

Another problem, also related with the practical applicability of the model, is that for a two-lid problem, it is hard to manually edit two lids consistently. This could be alleviated if one could find correspondences between both lids which could be used to help the user locate how to editing the second lid once the first one has been edited.

Regarding the numerical scheme, a more thorough analysis on the conditioning of the resulting system of equations is required, particularly because it could lead to ways to improve the conditioning, and thus accelerate the convergence of iterative solvers.

Last but not least, the current implementation requires storing in memory sparse matrices implementing the discrete operators. This is quite demanding in terms of memory and limits the size of the problems that can handled. It would be interesting to use an ad-hoc solver exploiting the structure of the problem to lower the memory requirements, for example by working on a frame by frame basis.

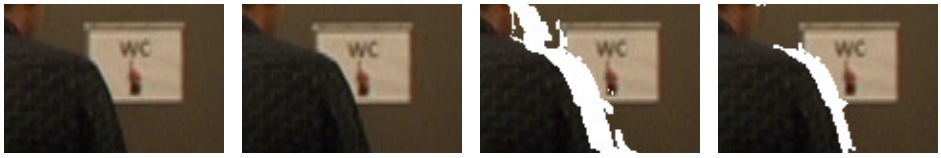
# Appendices of Part II

Appendix D

# An occlusion detection method

In this Section we discuss a two-step simple method to detect occlusions given a dense optical flow. In the first step we aim to identify regions which are potentially occluded. In the second step we categorize the previously selected regions into occluded and still visible. Figure D.1 shows an example of the two-step occlusion detection method.

The problem of occlusion detection is intrinsic to the optical flow problem. In fact, some optical flow algorithms estimate occlusions as part of the estimation of the movement Alvarez et al. (2007); Ayvaci et al. (2012). Such algorithms output an occlusion mask together with the optical flow.



**(a)** $\Omega_{t-1}$ **(b)** $\Omega_t$ **(c)** 1st step applied on $\Omega_{t-1}$ **(d)** 2nd step applied on $\Omega_{t-1}$

**Figure D.1:** The proposed two-step occlusion handling. For clarity, we show only the processing related to the forward optical flow from $\Omega_{t-1}$ to $\Omega_t$. Figures (a) and (b) show two consecutive frames. (c) shows the selection result of the first step described in Section D. (d) shows the selection result after the second step described in Section D. Figure 11.3 shows the output without and with occlusion handling respectively.

Many optical flow algorithms however, do not provide occlusion masks. To be able to work with such optical flows, we describe a simple method to detect occlusions. Any other occlusion detection method could be used instead.

### First step, the detection of potentially occluded regions

Let us consider two adjacent frames $t-1$ and $t$. In an ideal case the forward optical flow from $t-1$ to $t$ should map points in $t-1$ that exist in $t$, to their corresponding location in $t$. Occluded points at $t-1$, do not have a correspondence at $t$, and therefore an ideal optical flow should not be defined at these locations. The same applies for the backward optical flow.

In practice, most optical flow algorithms compute a dense correspondence from one frame to another. Thus we have a forward dense mapping from $t-1$ to $t$ given by the forward optical flow as $\boldsymbol{\varphi}^f_{t-1} : \Omega_{t-1} \to \Omega_t$, as $\boldsymbol{\varphi}^f_{t-1}(\boldsymbol{x}) = (\boldsymbol{x}+\boldsymbol{v}^f(\boldsymbol{x}, t-1), t)$. Similarly, we have a backward dense mapping at $t$ given by the backward optical flow, $\boldsymbol{\varphi}^b_t : \Omega_t \to \Omega_{t-1}$.

We will focus in the following on the forward mapping. An analogous discussion holds for the backward mapping.

For a given $\boldsymbol{x} \in \Omega_t$, we consider a neighborhood $Q_{\boldsymbol{x}} \subset \Omega_t$ given by $Q_{\boldsymbol{x}} = \boldsymbol{x} + (-1, 1)^2$. We denote by $[\boldsymbol{\varphi}^f_{t-1}]^{-1}(Q_{\boldsymbol{x}}) = \{\boldsymbol{z} \in \Omega_{t-1} : \boldsymbol{\varphi}^f_{t-1}(\boldsymbol{z}) \in Q_{\boldsymbol{x}}\}$ the pre-image of $Q_{\boldsymbol{x}}$ under the forward mapping $\varphi^f_{t-1}$. Note that this set is also defined even when $\boldsymbol{\varphi}^f_{t-1}$ is not invertible. We define the *forward area* as

$$\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}}) := \int_{[\boldsymbol{\varphi}^f_{t-1}]^{-1}(Q_{\boldsymbol{x}})} \mathrm{d}\boldsymbol{z}.$$

To compute this integral, we extend the discrete mapping $\boldsymbol{\varphi}^f_{t-1}$ to a continuous spatial domain using a bi-linear interpolation.

If $Q_{\boldsymbol{x}}$ is not being occluded nor dis-occluded from $t-1$ to $t$, then $[\boldsymbol{\varphi}^f_{t-1}]^{-1}$ is a well defined function on $Q_{\boldsymbol{x}}$. That is, every point in $Q_{\boldsymbol{x}}$ has a unique pre-image by the forward mapping $\boldsymbol{\varphi}^f_{t-1}$. Furthermore, for a locally translational flow $\boldsymbol{\varphi}^f_{t-1}$ we can expect $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}})$ to be close to the area of $Q_{\boldsymbol{x}}$, given by $\mathcal{A}(Q_{\boldsymbol{x}}) = \int_{Q_{\boldsymbol{x}}} \mathrm{d}\boldsymbol{z}$. In the case of $Q_{\boldsymbol{x}} = \boldsymbol{x} + (-1, 1)^2$ we have $\mathcal{A}(Q_{\boldsymbol{x}}) = 4$.

Based on the comparison between $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}})$ and $\mathcal{A}(Q_{\boldsymbol{x}})$ we will define potentially occluded regions to be further examined. We add a small margin

of $\varepsilon = 0.5$ to the comparison, to avoid marking all pixels as potentially occluded. We consider two cases:

1. $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}}) > \mathcal{A}(Q_{\boldsymbol{x}}) + \varepsilon$: There is an excess of points mapped into $Q_{\boldsymbol{x}}$ by $\boldsymbol{\varphi}^f_{t-1}$. This situation may arise when there is an occlusion from frame $t-1$ to $t$, causing several points in $\Omega_{t-1}$ to be mapped forward into the same location at frame $t$. Some of them are being occluded, whereas others remain visible. Therefore, we mark all pixels in $[\boldsymbol{\varphi}^f_{t-1}]^{-1}(Q_{\boldsymbol{x}})$ as candidates for occluded regions, for further examination. Note that there may be other reasons for which $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}}) > \mathcal{A}(Q_{\boldsymbol{x}}) + \varepsilon$, for example in the case of a zoom out. In this first step, we treat all of these cases equally.
2. $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}}) < \mathcal{A}(Q_{\boldsymbol{x}}) - \varepsilon$: Few points are mapped by $\boldsymbol{\varphi}^f_{t-1}$ into $Q_{\boldsymbol{x}}$. This may occur if $Q_{\boldsymbol{x}}$ lies in an region that has been dis-occluded from $t-1$ to $t$. If this is the case, some points in $Q_{\boldsymbol{x}}$ at frame $t$, do not have a correspondence in $t-1$, and we mark points in $Q_{\boldsymbol{x}}$ as candidates for dis-occluded regions (or occluded when looking from $t$ to $t-1$). Their backward optical flow will be examined in the second step. Note that the forward flows arriving at the dis-occluded region might be as well wrong (no point in $t-1$ should have a correspondence in a dis-occluded region at $t$). Thus we also mark them as candidates for further examination.

The same process is then applied considering the *backward area* mapped into frame $t-1$ from $t$ by the backward optical flow.

Figure D.1c shows an example output of the above discussed first-step occlusion detection.

**Remark.** A more general method for detecting occlusion candidates can be derived based on the consistency between the forward and backward optical flows Alvarez et al. (2007). In addition to the forward area $\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}})$, one could also define the *backward area* as

$$\mathcal{A}^b_t(Q_{\boldsymbol{x}}) := \int_{\boldsymbol{\varphi}^b_t(Q_{\boldsymbol{x}})} \mathrm{d}\boldsymbol{z}.$$

For an ideal optical flow, the forward and backward areas should be equal.

In an ideal case, the forward and backward flows are symmetrical (see for instance Alvarez et al. (2007)). Thus, the area of the pre-image of $Q_{\boldsymbol{x}}$

under the forward flow ($\mathcal{A}^f_{t-1}(Q_{\boldsymbol{x}})$) and the area of the image of $Q_{\boldsymbol{x}}$ under the backward flow ($\mathcal{A}^b_t(Q_{\boldsymbol{x}})$) should be equal. Therefore, differences between these areas could be considered as evidence of occlusions or dis-occlusions. Cases 1 and 2 in the above discussion apply, using $\mathcal{A}^b_t(Q_{\boldsymbol{x}})$ instead of $\mathcal{A}(Q_{\boldsymbol{x}})$. In practice, for the sequences we used, both approaches behave similarly.

### Second step, the categorization into occluded and still visible regions

As a result of the first step, we have for each frame $t$ two sets $C^f_t, C^b_t \subset \Omega_t$ of potentially occluded pixels. If a pixel $\boldsymbol{x}$ belongs to $C^f_t$, then its corresponding position at $\Omega_{t+1}$, given by the forward optical flow $\boldsymbol{\varphi}^f_t(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x}, t)$, might be occluded. Similarly, for a pixel $\boldsymbol{x} \in C^b_t$, its corresponding backward position $\boldsymbol{\varphi}^b_t(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{v}^b(\boldsymbol{x}, t)$ in $\Omega_{t-1}$ might be occluded. In the first step, all the analysis has been done based on the properties of the mapping between two frames, without considering the visibility of a pixel in the next frame. In this second step we try to determine which pixels in $C^f_t$ and $C^b_t$ are occluded and which are still visible. In what follows we describe this procedure for $C^f_t$. The same applies to $C^b_t$.

We base our occlusion detection on the error between a patch from frame $t$ centered at $\boldsymbol{x} \in C^f_t$ and its corresponding patch at frame $t+1$, centered at $\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x}, t)$. We will denote by $p_u(\boldsymbol{x}, t)$ the patch at $(\boldsymbol{x}, t)$. We define the patch error as the squared $L_2$ distance between corresponding patches:

$$e^f(\boldsymbol{x}) = \|p_u(\boldsymbol{x}, t) - p_u(\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x}, t), t+1)\|^2 = \sum_{\boldsymbol{h} \in \Omega_p} (u(\boldsymbol{x} + \boldsymbol{h}, t) - u(\boldsymbol{x} + \boldsymbol{v}^f(\boldsymbol{x}, t) + \boldsymbol{h}, t+1))^2, \quad \text{(D.1)}$$

where $\Omega_p \subset \mathbb{Z}^2$ denotes the patch domain (a square neighborhood of $\mathbf{0} \in \mathbb{Z}^2$ in our case).

To detect occluded regions we threshold $e^f$. However, setting a constant threshold that is not input sensitive might lead to undesired results. For that, we set a dynamic threshold adapted to each image pair. Therefore, we pose the occlusion detection as the following statistical test:

$\mathcal{H}_0$**:** the point at $(\boldsymbol{x}, t)$ is not occluded at $t+1$

$\mathcal{H}_1$**:** the point at $(\boldsymbol{x}, t)$ is occluded at $t+1$ .

As the statistic for the test we consider the patch error $e^f$, (we assume patch errors at different pixels to be independent). We estimate a statistical model for $\mathcal{H}_0$ by constructing a histogram of patch errors using the unlabeled points $S_t^f \setminus C_t^f$.

Based on this histogram, we compute a threshold $\tau_t^f > 0$ by fixing the false alarm rate $\alpha \in [0, 1]$, *i.e.* $\tau_t^f$ such that $P(e^f > \tau_t^f | \mathcal{H}_0) < \alpha$. In this way we compute an adaptive threshold for each frame transition. Patch errors above the threshold are considered as occluded. Let us mention that this test is designed not to be tolerant to false negatives (*i.e.* occluded pixels that have passed the test and been categorized as not occluded). Because of that, it is bound to allow for some false positives (*i.e.* pixels that are not occluded being categorized as occluded). In practice this over-estimation does not affect the quality of the result.

After applying this second step, for each frame in the sequence, the occlusion detection scheme yields two sets $K_t^f, K_t^b \subset \Omega_t$ of forward and backward occluded pixels.

Figure D.1d shows an example output of the above discussed second-step occlusion detection.

Appendix E

# The Euler-Lagrange equation

Throughout the rest of the paper, we assume that $O$ is a subset of $\Omega^T = \Omega \times [0,T]$ with Lipschitz boundary. Then the unit normal is defined almost everywhere on $\partial O$ with respect to the Hausdorff measure $\mathcal{H}^2$ (surface measure) on $\partial O$. Let us denote by $\boldsymbol{\nu}^O = (\boldsymbol{\nu}^O_{\boldsymbol{x}}, \nu^O_t)$ the outer unit normal to $\partial O$ (a vector in the unit sphere of $\mathbb{R}^3$) and $\boldsymbol{\nu}^{O_t}$ the outer unit normal to $\partial O_t$ (a vector in the unit circle of $\mathbb{R}^2$).

Let us compute the Euler-Lagrange equations associated to the energy

$$E_{\boldsymbol{\kappa},\lambda}(u) = \int_O \left( \frac{1}{2} \|\boldsymbol{\kappa}(\boldsymbol{x},t)\, \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x},t)\|^2 + \frac{\lambda}{p} \|\boldsymbol{\nabla} u(\boldsymbol{x},t)\|^p \right) \mathrm{d}\boldsymbol{x}\, \mathrm{d}t, \quad \text{(E.1)}$$

where $\lambda \geq 0$ and $p = 1, 2$. For that, assume that $u : O \to \mathbb{R}$ is a minimum of $E_{\boldsymbol{\kappa},\lambda}$. To compute the Euler-Lagrange equations, we consider a perturbation $\bar{u}$ such that $E_{\boldsymbol{\kappa},\lambda}(\bar{u}) < \infty$. Since $u$ is a minimum of $E_{\boldsymbol{\kappa},\lambda}$ we have

$$\begin{aligned} \lim_{\epsilon \to 0+} \frac{E_{\boldsymbol{\kappa},\lambda}(u + \epsilon \bar{u}) - E_{\boldsymbol{\kappa},\lambda}(u)}{\epsilon} &= \int_O \boldsymbol{\kappa} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\kappa} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} \bar{u}\, d\boldsymbol{x} dt \\ &+ \lambda \int_O \boldsymbol{\xi} \cdot \boldsymbol{\nabla} \bar{u} \mathrm{d}\boldsymbol{x}\, \mathrm{d}t = 0, \end{aligned}$$

where, when $\lambda > 0$ and $p = 1$, $\boldsymbol{\xi} : O \to \mathbb{R}^2$ is a measurable vector field such that $\|\boldsymbol{\xi}\|_\infty \leq 1$, $\boldsymbol{\xi} \cdot \boldsymbol{\nabla} u = |\boldsymbol{\nabla} u|$, and the arguments $(\boldsymbol{x}, t)$ of the functions are omitted for simplicity. If $\lambda > 0$ and $p = 2$, then $\boldsymbol{\xi} = \boldsymbol{\nabla} u$. Integrating by parts we have

$$0 = \int_O \boldsymbol{\kappa} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\kappa} \boldsymbol{\nabla} \partial_{\boldsymbol{v}} \bar{u}\, d\boldsymbol{x} dt + \lambda \int_O \boldsymbol{\xi} \cdot \boldsymbol{\nabla} \bar{u} \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$

$$= \int_O \partial_{\boldsymbol{v}}^* \boldsymbol{\nabla}^* (\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u) \bar{u} \, d\boldsymbol{x} dt + \lambda \int_O \boldsymbol{\nabla}^* \boldsymbol{\xi} \, \bar{u} \mathrm{d}\boldsymbol{x} \, \mathrm{d}t$$

$$+ \int_{\partial O} \boldsymbol{\nabla}^* (\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)(\nu_t^O + \boldsymbol{v} \cdot \boldsymbol{\nu}_{\boldsymbol{x}}^O) \bar{u} \, d\mathcal{H}^2 + \lambda \int_0^T \int_{\partial O_t} \boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t} \bar{u} d\mathcal{H}^1 dt$$

$$+ \int_0^T \int_{\partial O_t} \boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\nu}^{O_t} \partial_{\boldsymbol{v}} \bar{u} \, d\mathcal{H}^1 dt,$$

where $d\mathcal{H}^2$, resp. $d\mathcal{H}^1$, denotes the surface measure in $\partial O$, resp. the length measure in $\partial O_t$. We have denoted by $\boldsymbol{\nabla}^*$ (resp. $\partial_{\boldsymbol{v}}^*$) the adjoint operator, that is $\boldsymbol{\nabla}^* b = -\mathrm{div}\, b$ for any vector field $b : O \to \mathbb{R}^2$ (resp. $\partial_{\boldsymbol{v}}^* \psi = -\frac{\partial \psi}{\partial t} - \mathrm{div}\,(\boldsymbol{v}\psi)$, for any function $\psi : O \to \mathbb{R}$). By taking test functions that vanish in a neighborhood of the boundary we have $\bar{u} = 0$, $\partial_{\boldsymbol{v}} \bar{u} = 0$ on $\partial O$ and we deduce that

$$\partial_{\boldsymbol{v}}^* \boldsymbol{\nabla}^* (\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u) + \lambda \boldsymbol{\nabla}^* \boldsymbol{\xi} = 0 \quad \text{ in } O.$$

Introducing this in the above expressions we get

$$\begin{aligned} &\int_{\partial O} \boldsymbol{\nabla}^* (\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)(\nu_t^O + \boldsymbol{v} \cdot \boldsymbol{\nu}_{\boldsymbol{x}}^O) \bar{u} \, d\mathcal{H}^2 \\ &+\lambda \int_0^T \int_{\partial O_t} \boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t} \bar{u} d\mathcal{H}^1 dt \\ &+ \int_0^T \int_{\partial O_t} \boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\nu}^{O_t} \partial_{\boldsymbol{v}} \bar{u} \, d\mathcal{H}^1 dt = 0 \end{aligned} \tag{E.2}$$

and this holds for any admissible perturbation $\bar{u}$ that will be clarified below.

Let us discuss the boundary conditions that can be specified for the problem. We use the definition and notations given in Section 10.3. A set of natural boundary conditions are those for which the identity (E.2) holds. Let us discuss the possible choices.

*Dirichlet boundary conditions.* Dirichlet boundary conditions for $u$ can be specified on a given set $A \subset \partial O$ if $\lambda > 0$ or on a subset $A \subset \partial O \setminus \partial O_{\mathrm{tang}}$ if $\lambda = 0$. Namely we can specify

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t) \quad (\boldsymbol{x}, t) \in A. \tag{E.3}$$

If $u$ satisfies (E.3) and we take test functions $\bar{u}$ such that $\bar{u} = 0$ on $A$, then $u + \epsilon \bar{u}$ satisfies (E.3) and the first and second integrals in (E.2) vanishes on $A$.

Observe that, since $\partial O$ is Lipschitz,

$$\{(\boldsymbol{x},t) : \boldsymbol{x} \in \partial O_t, t \in (0,T)\} = \partial O_{\text{tang}} \cup \partial O_{\text{obli}},$$

where strictly speaking this equality holds modulo null sets with respect to the surface measure.

*Specifying $\partial_{\boldsymbol{v}} u$ on the boundary.* We can specify $\partial_{\boldsymbol{v}} u$ on a given subset of $\{(\boldsymbol{x},t) : \boldsymbol{x} \in \partial O_t, t \in (0,T)\}$. Namely we can specify

$$\partial_{\boldsymbol{v}} u(\boldsymbol{x},t) = g_0(\boldsymbol{x},t) \quad (\boldsymbol{x},t) \in B \subset \partial O_{\text{tang}} \cup \partial O_{\text{obli}}. \tag{E.4}$$

If $u$ satisfies (E.4) and we take test functions $\bar{u}$ such that $\partial_{\boldsymbol{v}} \bar{u} = 0$ on $B \subset \partial O_{\text{tang}} \cup \partial O_{\text{obli}}$, then $u + \epsilon \bar{u}$ satisfies (E.4) and the third integral in (E.2) vanishes on $B$.

*Specifying other boundary conditions.* We can specify the boundary condition at $(\boldsymbol{x},t) \in A' \subset \partial O$

$$\boldsymbol{\nabla}^*(\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)\boldsymbol{\nu}^O \cdot (\boldsymbol{v},1) + \lambda \boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t} = 0 \tag{E.5}$$

with the convention that $\boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t} = 0$ if $(\boldsymbol{x},t) \in \partial O_{\text{vert}} \cup O_0 \cup O_T$. Then the sum of the first and second integrals in (E.2) vanishes on $A'$.

Notice that if $\lambda = 0$, (E.5) reduces to

$$\boldsymbol{\nabla}^*(\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)\boldsymbol{\nu}^O \cdot (\boldsymbol{v},1) = 0 \tag{E.6}$$

and is trivially satisfied if $(\boldsymbol{x},t) \in \partial O_{\text{tang}}$ since in that case $\boldsymbol{\nu}^O \cdot (\boldsymbol{v},1) = 0$. That is, this gives no boundary condition at points $(\boldsymbol{x},t) \in \partial O_{\text{tang}}$. Thus, when $\lambda = 0$ we can only impose (E.6) on subsets $A' \subset \partial O \setminus \partial O_{\text{tang}}$.

If $\lambda > 0$, we can impose (E.5) on any subset $A' \subset \partial O$, understanding that it reduces to

$$\boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t} = 0. \tag{E.7}$$

*Specifying $\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\nu}^{O_t} = 0$ on the boundary.* We can specify the boundary condition at $(\boldsymbol{x},t) \in B' \subset \partial O_{\text{tang}} \cup \partial O_{\text{obli}}$

$$\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u \cdot \boldsymbol{\nu}^{O_t} = 0.$$

Then the second integral in (E.2) vanishes on $B'$.

Depending on the problem we choose a set of boundary conditions. The only requirements are that

$$A \cup A' = \partial O \quad \text{if } \lambda > 0, \text{ or } A \cup A' = \partial O \setminus \partial O_{\text{tang}} \quad \text{if } \lambda = 0,$$

and

$$B \cup B' = \partial O_{\text{tang}} \cup \partial O_{\text{obli}}.$$

This implies that the identity (E.2) holds.

*Boundary conditions for the one-lid setting.* In the context of the one-lid problem, we choose the set of boundary conditions

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}, 0), \qquad \boldsymbol{x} \in O_0, \tag{E.8}$$

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \partial O_{\text{vert}}, \tag{E.9}$$

$$\partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = g_0(\boldsymbol{x}, t) \ , \qquad (\boldsymbol{x}, t) \in \partial O_{\text{tang}} \setminus \partial \Omega^T, \tag{E.10}$$

$$\begin{array}{l} u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t) \\ \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) = g_0(\boldsymbol{x}, t) \end{array} , \qquad (\boldsymbol{x}, t) \in \partial O_{\text{obli}} \setminus \partial \Omega^T, \tag{E.11}$$

to which, when $\lambda > 0$, we add

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t) \qquad (\boldsymbol{x}, t) \in \partial O_{\text{tang}} \setminus \partial \Omega^T, \tag{E.12}$$

where the videos $u_0$ and $g_0$ are given. Notice that the boundary condition (E.12) is interpreted classically if $p = 2$ and it has to be interpreted in a relaxed sense if $p = 1$. This is discussed with more detail in Sadek et al. (2012).

The boundary conditions on the rest of $\partial O$ are

$$\boldsymbol{\nabla}^*(\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)(\boldsymbol{x}, t) = 0, \qquad \boldsymbol{x} \in O_T, \tag{E.13}$$

$$\begin{array}{l} \lambda \boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0 \\ \boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0 \end{array} , \qquad (\boldsymbol{x}, t) \in \partial O_{\text{tang}} \cap \partial \Omega^T, \tag{E.14}$$

$$\begin{array}{l} \boldsymbol{\nabla}^*(\boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u)(\boldsymbol{x}, t) + \lambda \boldsymbol{\xi} \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0 \\ \boldsymbol{\kappa}^2 \boldsymbol{\nabla} \partial_{\boldsymbol{v}} u(\boldsymbol{x}, t) \cdot \boldsymbol{\nu}^{O_t}(\boldsymbol{x}, t) = 0 \end{array} , (\boldsymbol{x}, t) \in \partial O_{\text{obli}} \cap \partial \Omega^T. \tag{E.15}$$

*Boundary conditions for the two-lid setting.* They are given by (E.8),(E.9),(E.10),(E.11),(E.14),(E.15), and (E.13) is replaced by

$$u(\boldsymbol{x}, T) = u_0(\boldsymbol{x}, T) \quad \text{in } O_T. \tag{E.16}$$

Let us observe that the boundary conditions (E.8),(E.9),(E.10), (E.11), and (E.16) in the two lid-case, are specified in the set of admissible functions in which $E_{\boldsymbol{\kappa},\lambda}$ will be minimized.

**Remark.** Under some assumptions on the vector field $\mathbf{v}$, we can prove existence and uniqueness of minima of $E_{\boldsymbol{\kappa},\lambda}$ in a suitable class of functions (the functional space where the energy is finite and permits to incorporate boundary conditions). In particular, this shows that the boundary conditions are sufficient to determine the solution. This is discussed in detail in Sadek et al. (2012).

Appendix F

# Analytic solutions of the Euler-Lagrange equation

To illustrate how the prescribed boundary conditions determine the solution, we compute in this section the analytic solution of the Euler-Lagrange equation of the continuous energy $E_{\kappa,\lambda}$ with $\lambda = 0$. We consider two simple examples, one for the one-lid setting and one for the two-lid setting.

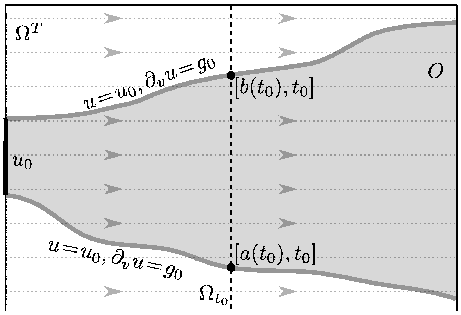



**Figure F.1:** Domain and boundary conditions for a one-lid problem. The optical flow in this example is zero.

We will consider a simple case, in which $v(x,t) = 0$ everywhere in $\Omega$. In this case the convective derivative coincides with the partial derivative with

respect to time: $\partial_v u = u_t$. The energy is therefore

$$E(u) = \int_O (u_{xt}(x,t))^2 \mathrm{d}x\mathrm{d}t. \tag{F.1}$$

We consider an editing domain $O = \{(x,t) : t \in [0,T], x \in [a(t), b(t)]\}$, such as the one depicted in Figure F.1. We consider $a : [0,T] \to \mathbb{R}$ and $b : [0,T] \to \mathbb{R}$ to be functions with a continuous bounded derivative, and such that $a(t) < b(t)$ for $t \in [0,T]$. We also suppose for simplicity that $a$ is a strictly decreasing function, whereas $b$ is strictly increasing.

## One-lid problem

For the one-lid problem, the boundary conditions are as follows:

$$u(x,0) = u_0(x), \qquad x \in [a(0), b(0)], \tag{F.2}$$
$$u(a(t),t) = u_a(t), \qquad t \in [0,T], \tag{F.3}$$
$$u(b(t),t) = u_b(t), \qquad t \in [0,T], \tag{F.4}$$
$$u_t(a(t),t) = g_a(t), \qquad t \in [0,T], \tag{F.5}$$
$$u_t(b(t),t) = g_b(t), \qquad t \in [0,T]. \tag{F.6}$$

The minimum of the energy can be computed by solving the Euler-Lagrange equation:

$$u_{txxt}(x,t) = 0, \qquad \forall (x,t) \in O \tag{F.7}$$

with the following additional boundary condition, stemming from the computation of the first variation:

$$u_{xxt}(x,T) = 0, \qquad x \in [a(T), b(T)]. \tag{F.8}$$

Let us compute the solution of the PDE (F.7). Integrating it with respect to $t$, between $t$ and $T$ yields,

$$u_{xxt}(x,t) = u_{xxt}(x,T) = 0, \qquad \forall (x,t) \in O,$$

the last equality is due to the Neumann boundary condition at $t = T$. If we now integrate w.r.t. $x$, between $a(t)$ and $x$ we obtain

$$u_{xt}(x,t) = u_{xt}(a(t),t), \qquad \forall (x,t) \in O.$$

We integrate again on variable $x$:

$$u_t(x,t) - u_t(a(t),t) = \int_{a(t)}^{x} u_{xt}(a(t),t)\mathrm{d}s = u_{xt}(a(t),t)(x - a(t)).$$

If we evaluate this expression on $x = b(t)$, we can express $u_{xt}(a(t),t)$ in terms of $g_a$ and $g_b$:

$$u_{xt}(a(t),t) = \frac{u_t(b(t),t) - u_t(a(t),t)}{b(t) - a(t)} = \frac{g_b(t) - g_a(t)}{b(t) - a(t)}.$$

Therefore we have that

$$u_t(x,t) = g_a(t) + \frac{g_b(t) - g_a(t)}{b(t) - a(t)}(x - a(t)). \tag{F.9}$$

As a function of $x$, $u_t(x,t)$ is a linear function passing through $(a(t), g_a(t))$ and $(b(t), g_b(t))$, *i.e.* the rate of illumination change is a smooth interpolation of the values specified at the boundary.

To obtain the solution of the PDE we now integrate with respect to $t$. Let us define the function $\ell : [a(T), b(T)] \to R$ as

$$\ell(x) = \begin{cases} a^{-1}(x) & \text{if } a(T) \leqslant x \leqslant a(0), \\ 0 & \text{if } a(0) < x < b(0), \\ b^{-1}(x) & \text{if } b(0) \leqslant x \leqslant b(T). \end{cases}$$

Note that $(x, \ell(x))$ with $x \in [a(T), b(T)]$, corresponds to the "left" boundary of $O$, where the value of $u$ is specified by $u_a$, $u_0$ and $u_b$. We now integrate (F.9) on $t$ between $\ell(x)$ and $t$, yielding

$$u(x,t) - u_0(x,\ell(x)) = \int_{\ell(x)}^{t} g_a(s)\mathrm{d}s + \int_{\ell(x)}^{t} \frac{g_b(s) - g_a(s)}{b(s) - a(s)}(x - a(s))\mathrm{d}s.$$

This example demonstrates that the given boundary conditions are sufficient to compute a minimizer of the energy. It also shows how each boundary condition is used.

In this example, for fixed $t$, the illumination change rate $\partial_v u(x,t) = u_t(x,t)$ is the result of a linear interpolation between the illumination change rate given at the boundary, $g_a(t)$ and $g_b(t)$. Once the illumination change rate is known in the whole editing domain, the solution is computed by integrating it along the trajectories. The temporal integration starts at $\ell(x)$, the time instant where the trajectory through $(x,t)$ reaches a point of the boundary where $u_0$ is specified (including the first lid where $\ell(x) = 0$). This integration starts with $u_0(x,\ell(x))$ and propagates it along the trajectory, while accommodating for the illumination changes previously computed in $u_t$.

## Two-lid problem

For the two-lid problem, in addition to the boundary conditions of the one-lid problem, we add the second lid:

$$u(x,T) = u_T(x), \quad x \in [a(T), b(T)] \tag{F.10}$$

in substitution of (F.8). We use the notation of Section F.

Let us compute the solution of

$$u_{txxt}(x,t) = 0, \qquad (x,t) \in O. \tag{F.11}$$

Observe that

$$O = \{(x,t) \in \Omega \times [0,T] : t \geq \ell(x)\ x \in [a(t), b(t)]\}.$$

Integrating with respect to $t$, we have

$$u_{xxt}(x,t) = u_{xxt}(x, \ell(x)) \quad (x,t) \in O. \tag{F.12}$$

Let us denote $G(x) = u_{xxt}(x, \ell(x))$. Integrating again with respect to $t$, we get

$$u_{xx}(x,t) = u_{xx}(x, \ell(x)) + G(x)(t - \ell(x)), \quad (x,t) \in O. \tag{F.13}$$

For $t = T$ and $x \in [a(T), b(T)]$ we have

$$u_{xx}(x,T) = u_{xx}(x, \ell(x)) + G(x)(T - \ell(x)). \tag{F.14}$$

Evaluating the above expression for $x \in [a(0), b(0)]$, we obtain

$$G(x) = \frac{1}{T}(u_{xx}(x,T) - u_{xx}(x,0)) \quad x \in [a(0), b(0)]. \tag{F.15}$$

Integrating (F.13) with respect to $x$, we obtain

$$\begin{aligned} u_x(x,t) = u_x(a(t),t) &+ \int_{a(t)}^{x} u_{xx}(s, \ell(s))\, ds \\ &+ \int_{a(t)}^{x} G(s)(t - \ell(s))\, ds. \end{aligned} \tag{F.16}$$

Integrating (F.16) with respect to $x$ from $a(t)$ to $x$, we obtain

$$\begin{aligned} u(x,t) &= u(a(t),t) + u_x(a(t),t)(x-a(t)) \\ &+ \int_{a(t)}^{x}\int_{a(t)}^{y} u_{xx}(s,\ell(s))\,ds\,dy + \int_{a(t)}^{x}\int_{a(t)}^{y} G(s)(t-\ell(s))\,ds\,dy \end{aligned} \tag{F.17}$$

and integrating (F.16) from $x$ to $b(t)$, we obtain

$$\begin{aligned} u(x,t) &= u(b(t),t) + u_x(a(t),t)(x-b(t)) \\ &- \int_{x}^{b(t)}\int_{a(t)}^{y} u_{xx}(s,\ell(s))\,ds\,dy - \int_{x}^{b(t)}\int_{a(t)}^{y} G(s)(t-\ell(s))\,ds\,dy. \end{aligned} \tag{F.18}$$

Evaluating (F.17) for $x = b(t)$ we obtain

$$\begin{aligned} u_b(t) &= u_a(t) + u_x(a(t),t)(b(t)-a(t)) \\ &+ \int_{a(t)}^{b(t)}\int_{a(t)}^{y} u_{xx}(s,\ell(s))\,ds\,dy + \int_{a(t)}^{b(t)}\int_{a(t)}^{y} G(s)(t-\ell(s))\,ds\,dy. \end{aligned} \tag{F.19}$$

Using (F.14) we obtain

$$(t-T)\int_{a(t)}^{b(t)}\int_{a(t)}^{y} G(s)\,ds\,dy = A(t), \tag{F.20}$$

where

$$\begin{aligned} A(t) &:= u_b(t) - u_a(t) - u_x(a(t),t)(b(t)-a(t)) \\ &- \int_{a(t)}^{b(t)}\int_{a(t)}^{y} u_{xx}(s,T)\,ds\,dy. \end{aligned} \tag{F.21}$$

Notice that by differentiating (F.3) with respect to $t$ and using (F.5) we obtain

$$u_x(a(t),t) = \frac{1}{a'(t)}(u_a'(t) - g_a(t)). \tag{F.22}$$

Similarly, we obtain

$$u_x(b(t),t) = \frac{1}{b'(t)}(u_b'(t) - g_b(t)). \tag{F.23}$$

Now, taking derivatives with respect to $t$ in (F.18) and evaluation the expression for $x = b(t)$ we obtain

$$\begin{aligned}(u_x(a(t),t) - u_x(b(t),t) - u_x(a(t),T) + u_x(b(t),T))b'(t) \\ = -\int_{a(t)}^{b(t)} G(s)(T-\ell(s))\,ds - \int_{a(t)}^{b(t)} G(s)(t-\ell(s))\,ds.\end{aligned} \tag{F.24}$$

Let $B(t)$ be the left hand side of (F.24). By differentiating twice (F.20) with respect to $t$, we get

$$G(b(t))b'(t) - G(a(t))a'(t) = Q'(t), \tag{F.25}$$

where

$$Q(t) = \frac{1}{(t-T)b'(t)}\left(A'(t) - \frac{A(t)}{(t-T)}\right).$$

By differentiating (F.24) with respect to $t$, we get

$$G(b(t))b'(t) - G(a(t))a'(t) = \frac{B'(t)+Q(t)}{t-T}. \tag{F.26}$$

From (F.25), (F.26) we get $G(a(t))$ and $G(b(t))$. Together with $G(x)$, $x \in [a(0), b(0)]$ we have $G(x)$ for all $x \in [a(T), b(T)]$.

From (F.14) we obtain $u_{xx}(x, \ell(x))$. Then we have all ingredient to evaluate $u(x,t)$ using (F.17).

This example illustrates in a simple case how the boundary conditions are used to determine the solution in the two-lid setting.

# Bibliography

Each reference indicates the pages where it appears.


R.A. Adams. *Sobolev spaces*. Academic Press, New York, 1975.

A. Agrawal and R. Raskar. Gradient domain manipulation techniques in vision and graphics, 2007. ICCV'07 short course.

M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representatio. *IEEE Trans. on Signal Processing*, 54(11):4311–22, 2006.

C.D. Aliprantis and K.C. Border. *Infinite dimensional analysis: a hitchhiker's guide*. Springer Verlag, 2006.

A. Almansa, V. Caselles, G. Haro, and B. Rougé. Restoration and zoom of irregularly sampled, blurred, and noisy images by accurate total variation minimization with local constraints. *Multiscale Modeling and Simulation*, 5(1):235–72, 2006.

L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. *Int. J. Comput. Vision*, 75(3):371–385, December 2007.

L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford University Press, USA, 2000.

P. Arias, V. Caselles, and G. Sapiro. A variational framework for non-local image inpainting. In *EMMCVPR*, Lecture Notes in Computer Science, pages 345–58. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision*, 93:319–347, 2011.

P. Arias, V. Caselles, and G. Facciolo. Analysis of a variational framework for exemplar-based image inpainting. *Multiscale Modeling & Simulation*, 10(2):473–514, 2012.

J.-F. Aujol, S. Ladjal, and S. Masnou. Exemplar-based inpainting from a variational point of view. *SIAM J. Math. Anal.*, 42(3):1246–85, 2010.

S. P. Awate and R. T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. on PAMI*, 28(3): 364–376, 2006.

A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *International Journal of Computer Vision (IJCV)*, 97(3):322–338, May 2012.

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. 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 Trans. on IP*, 10(8):1200–11, 2001.

C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: a randomized correspondence algorithm for structural image editing. In *Proc. of SIGGRAPH*, pages 1–11, New York, NY, USA, 2009. ACM.

C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*, 2010.

M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of SIGGRAPH*, New York, NY, USA, 2000. ACM.

M. Bertalmío, L. Vese, G. Sapiro, and S. J. Osher. Simultaneous structure and texture inpainting. *IEEE Trans. on IP*, 12(8):882–89, 2003.

D.P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, MA, 1999.

P. Bhat, C.L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M.F. Cohen, B. Curless, and S.B. Kang. Using photographs to enhance videos of a static scene. In *Proceedings of the Eurographics Symposium 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*, 29:1–14, April 2010.

R. Bornard, E. Lecan, L. Laborelli, and J.-H. Chenot. Missing data correction in still images and image sequences. In *Proc. ACM Int. Conf. on Multimedia*, 2002.

F. Bornemann and T. März. Fast image inpainting based on coherence

transport. *J. of Math. Imag. and Vis.*, 28(3):259–78, 2007.

T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(3):500–513, 2011.

T. Brox, O. Kleinschmidt, and D. Cremers. Efficient nonlocal means for denoising of textural patterns. *IEEE Trans. on IP*, 17(7):1057–92, 2008.

A. Buades, B. Coll, and J.-M. Morel. A non local algorithm for image denoising. In *Proc. of the IEEE Conf. on CVPR*, volume 2, pages 60–65, 2005.

A. Bugeau, P. Gargallo, O. D'Hondt, A Hervieu, N. Papadakis, and V. Caselles. Coherent Background Video Inpainting through Kalman Smoothing along Trajectories. In *Modeling, and Visualization Workshop*, page 8, 2010.

E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

P. Cannarsa and C. Sinestrari. *Semiconcave Functions, Hamilton-Jacobi Equations, and Optimal Control*, volume 58 of *Progress in Nonlinear Differential Equations and their Applications*. Birkhäuser, 2004.

F. Cao, Y. Gousseau, S. Masnou, and P. PÃ©rez. Geometrically guided exemplar-based inpainting. *SIAM Journal on Imaging Sciences*, 4(4): 1143–1179, 2011.

A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

T. Chan and J. H. Shen. Mathematical models for local nontexture inpaintings. *SIAM J. App. Math.*, 62(3):1019–43, 2001.

T. Chan, S. H. Kang, and J. H. Shen. Euler's elastica and curvature based inpaintings. *SIAM J. App. Math.*, 63(2):564–92, 2002.

Y. Cheng. Mean shift, mode seeking and clustering. *IEEE Trans. on PAMI*, 17(8):790–99, 1995.

A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. *IEEE Trans. on IP*, 13(9):1200–1212, 2004.

L. Demanet, B. Song, and T. Chan. Image inpainting by correspondence maps: a deterministic approach. *Applied and Computational Mathematics*, 1100:217–50, 2003.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. of the Roy. Stat. Soc. Series B*, 39(1):1–38, 1957.

I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. *ACM Trans. on Graphics. Special issue: Proc. of ACM SIGGRAPH*, 22 (3):303–12, 2003.

A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of the IEEE ICCV*, pages 1033–38, September 1999.

M. Elad, J. L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

S. Esedoglu and J. H. Shen. Digital image inpainting by the Mumford-Shah-Euler image model. *European J. App. Math.*, 13:353–70, 2002.

G. Facciolo, P. Arias, V. Caselles, and G. Sapiro. Exemplar-based interpolation of sparsely sampled images. In *EMMCVPR*, Lecture Notes in Computer Science, pages 331–44. Springer Berlin Heidelberg, 2009.

G. Facciolo, R. Sadek, A. Bugeau, and V. Caselles. Temporally consistent gradient domain video editing. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 59–73. Springer-Verlag, 2011.

C.-W. Fang and J.-J. J. Lien. Rapid image completion system using multiresolution patch-based directional and nondirectional approaches. *IEEE Trans. on IP*, 18(12):2769–2779, 2009.

R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics*, 21:249–256, July 2002.

G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(1):59–68, 2006.

T. Georgiev. Image reconstruction invariant to relighting. In *Eurographics*, pages 61–4, 2005.

D. Gilbarg and N.S. Trudinger. *Elliptic partial differential equations of second order*. Springer Verlag, 2001.

G. Gilboa and S. J. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling and Simulation*, 6(2):595–630, 2007.

G. Gilboa and S. J. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling and Simulation*, 7(3):1005–1028, 2008.

P. Harrison. *Texture tools.* PhD thesis, Monash University, 2005.

K. He and J. Sun. Statistics of patch offsets for image completion. In *Proc. of ECCV*, 2012.

M. Holtzman-Gazit and I. Yavneh. A scale-consistent approach to image completion. *Int. J. Multiscale Comput. Eng.*, 6(6):617–628, 2008.

B.K.P. Horn. *Robot Vision.* Electrical Engineering and Computer Science. MIT Press, 1986.

B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

H. Igehy and L. Pereira. Image replacement through texture synthesis. In *Proc. of the IEEE ICIP*, October 1997.

E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–30, 1957.

J. Jia and C.-K. Tang. Inference of segmented color and texture description by tensor voting. *IEEE Trans. on PAMI*, 26(6):771–86, 2004.

J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang. Video repairing under variable illumination using cyclic motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):832–9, 2006.

N. Kawai, T. Sato, and N. Yokoya. Image inpainting considering brightness change and spatial locality of textures and its evaluation. In *Ad. in Image and Video Tech.*, pages 271–82. Springer Berlin Heidelberg, 2009.

S. Kimball, P. Mattis, and the GIMP Dev. Team. GIMP: GNU Image Manipulation Program. http://www.gimp.org/. Version 2.6.8 released on December 2009.

S. Kindermann, S. Osher, and P.W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation*, 4 (4):1091–1115, 2006.

A.C. Kokaram, B. Collis, and S. Robinson. Automated rig removal with bayesian motion interpolation. *IEEE Journal on Vision, Image and Signal Processing*, 152:407–414, Aug 2005.

N. Komodakis and G. Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Trans. on IP*, 16(11):2649–61, 2007.

V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Trans. on Graph.*, 24(3):795–802, 2005.

A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *Proc. of IEEE ICCV*, 2003.

E. Levina and P. Bickel. Texture synthesis and non-parametric resampling of random fields. *Annals of Statistics*, 34(4), 2006.

O. Lezoray, A. Elmoataz, and S. Bougleux. Graph regularization for color image processing. *Comput. Vis. Image Underst.*, 107(1-2):38–55, 2007.

C. Liu, W.T. Freeman, E.H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2008a.

C. Liu, W.T. Freeman, E.H. Adelson, and Y. Weiss. Human-assisted motion annotation. online code and database. http://people.csail.mit.edu/celiu/motionAnnotation/, 2008b.

Y. Liu and V. Caselles. Exemplar-based image inpainting using multiscale graph cuts. *IEEE Trans. on IP*, 22(5):1699–1711, 2013.

J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *SIAM Multiscale Modeling and Simulation*, 7(1):214–241, 2008.

S. Masnou. Disocclusion: a variational approach using level lines. *IEEE Trans. on IP*, 11(2):68–76, 2002.

S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Proc. of IEEE ICIP*, 1998.

E. Meinhardt-Llopis and J. Sánchez. Horn-schunck optical flow with a multiscale strategy. 2012. preprint.

M. Mézard and A. Montanari. *Information, physics and computation*. Oxford University Press, 2009.

J.-M. Morel and G. Yu. On the consistency of the SIFT Method. *Preprint, CMLA*, 26, 2008.

S. Negahdaripour. Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(9):961–979, 1998.

N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision (IJCV)*, 67(2):141–158, April 2006.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers, Inc., 1998.

P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22:313–318, 2003.

P. Pérez, M. Gangnet, and A. Blake. PatchWorks: Example-based region tiling for image editing. Technical report, Microsoft Research, 2004.

G. Peyré. Manifold models for signals and images. *Comp. Vis. and Im. Unders.*, 113(2):249–260, 2009.

G. Peyré, S. Bougleux, and L. Cohen. Non-local regularization of inverse problems. In *ECCV '08*, pages 57–68, Berlin, Heidelberg, 2008. Springer-Verlag.

G. Peyré, S. Bougleux, and L. D. Cohen. Non-local regularization of inverse problems. *Inverse Problems and Imaging*, 5(2):511–530, 2011.

L. Pizarro, P. Mrázek, S. Didas, S. Grewenig, and J. Weickert. Generalised nonlocal image smoothing. *IJCV*, 90:62–87, 2010.

Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Proc. of the 12th ICCV*, pages 151–158, Kyoto, Sept 2009.

M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. *IEEE Trans. on IP*, 18 (1):36–51, 2009.

A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P Goldman-Rakic, and J. Duncan. A review of image denoising algorithms, with a new one. *Medical Image Analysis*, 4(1):379–398, 1997.

A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: A new representation for video editing. *ACM Transactions on Graphics (SIGGRAPH 2008)*, August 2008.

K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

R. Sadek, G. Facciolo, P. Arias, and V. Caselles. A variational model for gradient-based video editing. Online supplementary material. `http://gpi.upf.edu/static/gbve12/`, 2012.

A. Salgado and J. Sánchez. Temporal constraints in large optical flow estimation. In *Proc. of the 11th Int. Conf. on Comp. Aided Systems Theory*, EUROCAST'07, pages 709–716, Berlin, Heidelberg, 2007. Springer-Verlag.

P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision (IJCV)*, 80: 72–91, 2008.

J. Shen, X. Jin, and C. Zhou. Gradient based image completion by solving poisson equation. *Ad. in Multimedia Information Processing*, pages 257–

68, 2005.

T. Shiratori, Y. Matsushita, X. Tang, and S.B. Kang. Video completion by motion field transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 411–418, 2006.

S.M. Srivastava. *A course on Borel sets*, volume 180. Graduate Texts in Mathematics, Springer Verlag, 1998.

D. Sun. Online optical flow matlab code. `http://www.cs.brown.edu/~dqsun/research/software.html`, 2010.

D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439, 2010.

D. Sun, E. Sudderth, and M. J. Black. Layered segmentation and optical flow estimation over time. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR*, 2012.

J. Sun, L. Yuan, J. Jia, and H. Y. Shum. Image completion with structure propagation. In *Proc. of SIGGRAPH*, New York, NY, USA, 2005. ACM.

D. Tschumperlé and R. Deriche. Vector-valued image regularization with PDE's: a common framework for different applications. *IEEE Trans. on PAMI*, 27(4), 2005.

S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–87, 1988.

S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. In *Proc. Thirteenth Int. Conf. on Computer Vision*, Barcelona, November 2011. IEEE Computer Society Press.

H. Wang, X. Xu, R. Raskar, and N. Ahuja. Videoshop: A new framework for spatio-temporal video editing in gradient domain. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proc. of the SIGGRAPH*, New York, NY, USA, 2000. ACM.

J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational optic flow methods for small displacements. In O. Scherzer, editor, *Mathematical Models for Registration and Applications to Medical Imaging*, volume 10 of *Mathematics in Industry*. Springer, Berlin, 2006.

Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Trans. on PAMI*, 29(3):463–476, 2007.

C.F.J. Wu. On the convergence properties of the EM algorithm. *The Annals*

*of Statistics*, pages 95–103, 1983.

A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006.

Y. Zhang, J. Xiao, and M. Shah. Motion Layer Based Object Removal in Videos. In *7th IEEE Workshops on Application of Computer Vision*, 2005.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th DAGM Symposium*, pages 361–368. Springer, 2005.

M. H. Zhou, M. Mascagni, and A. Y. Qiao. Explicit Finite Difference Schemes for the Advection Equation. *Relation*, 10(1.55):7098, 1998.