Università degli Studi di Salerno

Dipartimento di Ingegneria Elettronica ed Ingegneria Informatica

Dottorato di Ricerca in Ingegneria dell'Informazione
X Ciclo – Nuova Serie

TESI DI DOTTORATO

Image partition and video segmentation using the Mumford-Shah functional

CANDIDATO: **ALFREDO CUTOLO**

TUTOR: **PROF. GIULANO GARGIULO**

CO- TUTOR: **PROF. ABDELAZIZ RHANDI**

COORDINATORE: **PROF. ANGELO MARCELLI**

Anno Accademico 2010 – 2011

To Mariarosaria
and my Parents

Contents

Introduction

The aim of this Thesis is to present an image partition and video segmentation procedure, based on the minimization of a modified version of Mumford-Shah functional. Generally, in most image processing applications, an image is usually viewed as a set of pixels placed on a rectangular grid. A single pixel provides an extremely local information making impossible any kind of interpretation.

The proposed approach, instead, follows a region based image representations. This approach is used, for instance, in MPEG-4 [27] or MPEG-7 [85] standards. In such cases the image is understood as a set of objects. Region-based image representations offer two advantages with respect to the pixel based ones: the number of regions is lower than the number of original pixels and regions represent a first level of abstraction with respect to the raw information.

The basic objects used of the image partition procedure are the upper and lower level sets of the image. In order to have a more local description of it, we deal with the connected components of (upper or lower) level sets. As proposed by Caselles et al. in [23], we have considered the boundary of these sets, that is the *level lines*, forming the topographic map.

To be able to handle discontinuous functions, more specifically, upper semicontinuous ones, we define level lines as the external boundary of the level sets of the image. This leads us to the notion of shape which consists in filling the holes of the connected components of the level sets, upper or lower, of the original image. The operation of hole filling was called saturation in [1], [68]. Thus, level lines are the boundaries of shapes and to give the family of level lines is equivalent to give the family of shapes.

Moreover, the family of connected components of upper level lines has a tree structure. And the same happens for the family of connected components of lower level lines. These two trees can be merged in a single tree: the "*Tree of Shapes*" of an image [69]. It gives a complete and non-redundant representation of the image and is

contrast independent. The tree is equivalent to the image: its knowledge is sufficient to reconstruct the image.

The image partition procedure determined by level lines is based on the minimization of a simplified version of the Mumford - Shah functional. If we minimize the functional with respect to all possible partitions, the problem of finding a global minimum is exponentially complex. But, if the minimization takes place in a hierarchy of partitions, global minima can be obtained quickly [43] [40].

To build the hierarchy of partitions the tree of image shape has been used. In particular the regions determined by level lines are taken as an initial partition of a hierarchy which can be constructed using the simplified Mumford-Shah functional. Then, using Guigues optimization algorithm [43], the global minima of the energy in the hierarchy can be defined at any scale obtaining the searched image partition.

The Mumford-Shah functional used for image partition has been then extended to develop a video segmentation procedure. Differently by the image processing, in video analysis besides the usual spatial connectivity of pixels (or regions) on each single frame, we have a natural notion of "*temporal*" connectivity between pixels (or regions) on consecutive frames given by the optical flow. In this case, it makes sense to extend the tree data structure used to model a single image with a graph data structure that allows to handle a video sequence.

We have developed the appropriate graph pre-computing a dense optical flow of the whole video sequence using any of the methods available in literature. So, we have defined the vertices of the graph as all the video pixels, assigning to each one its corresponding gray level. The edges of the graph are of two kinds: spatial edges and temporal edges. Spatial edges join each pixel with its 8-neighbors on the same frame. Temporal edges are defined using the pre-computed optical flow.

The video segmentation procedure is based on minimization of a modified version of a Mumford-Shah functional. In particular the functional used for image partition allows to merge neighboring regions with similar color without considering their movement. Our idea has been to merge neighboring regions with similar color and similar optical flow vector. Also in this case the minimization of Mumford-Shah functional can be very complex if we consider each

possible combination of the graph nodes. This computation becomes easy to do if we take into account a hierarchy of partitions constructed starting by the nodes of the graph. The global minima of the functional can be defined at any scale using the same optimization algorithm for the image partition [43] obtaining the video segmentation.

Plan of the thesis

The thesis is organized as follows:

The first chapter reports the different representations of the topology of the image that can be found in the literature. We address our attention at the tree of shape as the data structure for image representation. This structure allows to reconstruct the original image. In particular we show how is possible to construct it merging the tree of shape of upper and lower level lines of the image. To compute the merging operation has been necessary to define the notation of hole and saturation. Then in Chapter 2 we describe an image partition procedure based on minimization of a Mumford Shah functional. The problem of the quick computation of minima using a hierarchy of partitions constructed on the tree of image shape is faced. The section ends with some experimental results.

In Chapter 3 we revise some aspects of the image sequence formation, and the motion estimation problem. We also review the main optical flow estimation methods known in literature. Last section, Chapter 4, proposes a video segmentation procedure based on the minimization of a modified version of the Mumford Shah functional. We describe the data structure used to handle the video sequence characterized by spatial connections (between pixels or regions of the same frame) and temporal connections (defined by the optical flow vector). The procedure adopted to minimize quickly the functional is presented. At the end we show some experimental results comparing a video segmentation obtained with the simplified Mumford Shah functional for image partition with the new introduced functional.

Chapter 1

Data Structure for Image Representation

In this chapter we review some issues related to image representation. Representations based on regions are interesting for many image processing applications. Among them, we emphasize the tree of shapes of an image, which gives a compact structure of the level lines of an image. The level lines are the boundaries of the upper or lower level sets of an image. We revise the main properties of these level sets, and the definition of shapes from them, as well as we derive the tree structure of the shapes.

1.1 Different Image Representations

Image representations can be different depending on their purpose. The raw information, that is the values of the samples, or pixels, is a too low level of representation, and the image must be described with more elaborate models.

For a deblurring, restoration, denoising purpose, the representations based on the Fourier transform are generally the best since they rely on the generation process of the image (Shannon theory), and/or on the frequency models of the degradation as for additive noise, or spurious convolution kernel. However, the Fourier transform is purely frequency oriented and does not give directly any space information. The wavelet theory [59][65], achieves a localization of the frequencies, and, due to the linear structure of the

images at their smallest scales, the wavelet representation is to date the best representation of the image for compression purpose.

Nevertheless, from the image analysis point of view, frequency based representations do not give the adequate information. Indeed, the Fourier representation is nonlocal and the wavelet representation is sensitive to a translation, rotation or scaling in the image, disabling the recognition of objects independently of the viewpoint. Moreover, both of these representations have quantized observation scales.

Scale-space and edge detection theories propose to represent the images by some significant edges, where edges are defined suitably. The algorithms proceed in general in two steps (which sometimes can be merged): first the images are (linearly or not) smoothed [1][21] and secondly an edge detector is applied to the smoothed image. Edges are detected based on the second order derivatives of the image. The earliest definition of edges is due to Marr and Hildreth [61] and a variant was later proposed by Canny [22]. The scale represents the amount of smoothing prior to edge detection. The first scale-space based on edges is the zero-crossing of the Laplacian across the gaussian pyramid, that is the smoothing is a convolution with a gaussian kernel of varying variance. According to Marr, those zero-crossings represent the “raw primal sketch” of the image, that is the basis on which further vision algorithms should rely, see Marr [60] and Hummel [47]. In general, edges extraction can be formulated as a variational problem, see Nitzberg and Mumford [76], Morel and Solimini [70]. The image is approximated by a function that stands in a class of functions for which edges are properly defined: a famous example of such a class is the family of piecewise constant images having a bounded discontinuity length; in this class, the discontinuities lines of the approximating function are interpreted as the edges, see Mumford and Shah [71]. Then, a balance between how close and how complex the approximation is (e.g., with the previous example, the complexity can be the length of the discontinuity boundary), defines a scaled representation of the image.

Despite the generality of the variational approach, it suffers from the fact that there is no theory that says what the model should be. These representations by the edges have two major drawbacks that have been discussed, see Koenderink [51], Witkin [90] and Mallat [59], but not solved within the scale-space theory. First, the geometric

representation by the edges is incomplete: it does not allow a full reconstruction of the image, therefore some information has been lost in the process of edge detection. Secondly, the decomposition in scales yields a redundant representation.

Another problem with these approaches is linked to the fact that the image gray level is not an absolute data, since in many cases the contrast is camera dependent, and the optics of the camera is generally unknown, and in all cases hard to measure. This problem can be avoided by working in the morphological framework considering the level set and the level lines.

1.2 Level set and level lines

In natural images, the contrast depends on the type of camera, on the digitization process, due to the gray level quantization, to the lightning... Despite this multiplicity of factors changing the contrast, the perception of the image must remain identical, independent of the screen on which it is displayed. In other words, the contrast information is secondary relatively to the geometric information, and useful mainly for visual convenience.

The invariance under change of contrast has been first stated as a Gestalt principle by Wertheimer [89].

Matheron [62] and after him Serra [80], [81] propose a "*morphological*" representation of the images by their *level sets*. It yields a complete, contrast invariant representation of the image, independent on any parameter. A variant of this representation is proposed by Caselles et al. in [23], by considering the boundary of these sets, that is the *level lines*, forming the topographic map.

In general a (gray level) image is represented by a function $u:\Omega\to\mathbb{R}$ defined in a domain $\Omega\in\mathbb{R}^2$. The most basic elements of mathematical morphology are the level sets. We call superior level set Ω and inferior level set $X_\lambda u$ of value λ the subset of Ω defined as follows:

$$X^{\lambda}u = [u \geq \lambda] = \{p \in \Omega,\ u(p) \geq \lambda\}$$
$$X_{\lambda}u = [u < \lambda] = \{p \in \Omega,\ u(p) < \lambda\} \quad (1.1)$$

The convention to take strict inequality for lower level sets and large inequality for upper level sets is to get consistency results between them, i.e., $\Omega \setminus X^{\lambda}u = X_{\lambda}u$.

Whereas it is usually of minor importance because we do not mix upper and lower level sets (1.1), it becomes fundamental when we deal with both simultaneously.

Furthermore, topological characteristics extracted from level sets are also morphological. A particular case is the connected components of the boundaries of level sets, which are called level lines. Another case is taking the connected components of level sets, which are used in the following chapter to construct “shapes”.

Our interest about the level sets comes also from the fact that they are a *representation of the image*. From the lower level sets of an image u, we can recover u by the formula:

$$\forall p \in \Omega,\ u(p) = \inf\{\lambda : p \in X_{\lambda}u\} \quad (1.2)$$

and for the upper level set by the formula:

$$\forall p \in \Omega,\ u(p) = \sup\{\lambda : p \in X^{\lambda}u\} \quad (1.3)$$

In the last case, thanks to the non strict inequality, the supremum is actually a maximum, since $p \in X^{u(p)}u$.

1.3 Image and its topology

Once the image is segmented, one way or another, the resulting topology must be described. The usual notion of segmentation is a partition of the image into connected regions and the relations between these regions are meaningful. The first idea is to encode the

adjacency relations: we need to know when two regions have a common boundary. The classical way to represent this relation is through a graph, the Region Adjacency Graph (RAG): each region is represented as a vertex in the graph and when two regions are adjacent, an edge links the corresponding vertices, see Rosenfeld [79]. Nevertheless, adjacency is not the only meaningful relation between regions. For example, if two regions are adjacent, the number of connected components of their boundary is not encoded. The solution to this problem would be to add the corresponding number of edges between the two vertices, yielding then a multigraph. More annoying is the problem that the knowledge that a region is a hole inside another region is not contained in the (multi) graph. Gangnet et al. [41], recognizing that these data are missing, propose to add the inclusion structure of contours to the graphs. However, this represents the topology of the image in two graphs, making it uneasy to manipulate. Observing the difficulty to describe the relations between regions in terms of pixels only, Kovalesky in [54] proposes a cell-list representation, adding frontiers between regions as 1-dimensional elements and the junction points of regions of these frontiers as 0-dimensional elements. However, his structure is not a graph, and does not encode more data than the RAG.

Following the direction opened by Kovalesky, Fiorio in [37] uses the same elements to construct its representation as a combinatorial map (see Lienhardt [57]) and exposes an algorithm of linear complexity to construct his representation, the Frontiers Topological Graph. Fiorio emphasizes the fact that the representation must be consistent with the usual topology of the plane, and that it must introduce the minimum number of elements of non maximal dimension to this purpose. In [38], he generalizes to higher dimensions this representation, whereas in [39], he explains how to manipulate the Frontiers Topological Graph, in particular how to update the structure when two regions are merged and how to extract the Frontiers Topological Graph of a subimage, provided the subimage does not cut regions. Unfortunately, these basic operations are not obvious, coming from the fact that the combinatorial map is a fairly complex representation.

1.3.1 Topology of morphological representation

All these topological representations are based on a segmentation of the image understood as a partition into connected regions. But the basic elements of mathematical morphology, the level sets, do not compose a partition of the image; instead, they are hierarchical, because they are ordered. When talking about whole level sets, this order, the inclusion relation, is total, yielding a very elementary structure, an ordered list. However, it lacks an important feature of the above representations, the locality, or the fact that the atoms of the representation (the level sets) are not connected. Hence comes the need for considering instead the connected components of the level sets.

A fruitful approach is proposed by Ballester, Caselles and Morel in [5], where the atoms are some parts of the connected components of bilevel sets, that is points whose values are comprised between two given thresholds. They are chosen so that when the thresholds are changed in a manner to have an included bilevel, the subpart of the atom remains connected. These atoms are called the maximal monotone sections, and are invariant with respect to contrast change. Their study comes from a successful shape preserving local contrast enhancement algorithm proposed by Caselles et al. in [24] and [26]. However, the relations between these structures are not totally studied, and their efficiency in terms of compactness of the representation remains to be demonstrated.

Cox and Karron [30] explore the structure of the family of connected components of upper level sets in a 3-D image for purposes of coding and visualization of 3-D data. They show that the image can be described as a discrete structure, the tree of criticalities. They call it the Digital Morse Theory, because it is analogous to the Morse theory for continuously defined functions: a Morse function, that is a twice continuously differentiable function, in which the Hessian matrix is non degenerate at critical points, can be described by a tree of criticalities (see Milnor [67]). From discrete data, a three-dimensional array of gray levels, they define the continuous interpolated functions which are topologically consistent with the discrete data and show that they share the same tree of criticalities. Whereas they remark that using the discrete notions of connectedness (there are two: 4 and

8connectedness in 2-D, 6 and 26 connectedness in 3-D) without reference to the interpolated function can yield inconsistencies when we take the opposite of the image, they do not push this remark to its natural conclusion: upper level sets are not sufficient to describe topologically the image, because they are adapted to light objects, but the dark objects are not well represented in the digital Morse tree.

In a study on numerical functions defined on a rectangle of $\mathbb{R}^2$, published in 1950, Kronrod [55] avoids this drawback. Indeed, the atoms in his work are connected components of isolevel sets, which are continua. Given such a component K and a neighborhood U of K, if we call open set the family of the connected components of isolevel sets contained in U, the family of all these sets forms a topology on the set of connected components of isolevel sets of the image. The natural map, that with a point of the rectangle associates the connected component of isolevel set containing it, is continuous. Since the square is connected, locally connected and compact, so is the topological space of connected components of isolevel sets. He shows furthermore that no subset of this space is homeomorphic to the circle S^1, concluding that this space is actually a *tree*, in the topological sense. Moreover, he shows that this tree has an at most countable number of leaves and of ramification points, and that the leaves are connected components of isolevel sets not separating the rectangle (they are some regional extrema, but also what he calls concentric singularities), whereas ramification points are those separating the rectangle in at least three parts. He calls this tree the one-dimensional tree of the function and describes the functions which are in the same family as a given one: they are obtained by merging some parts of the tree. In many respects, this construction is remarkable: the family of connected components of isolevel sets is globally invariant under a contrast change, but also under an inversion of contrast (taking the negative of the function), which was the feature lacking to the digital Morse tree. However, from the image representation point of view it suffers from two drawbacks: isolevel sets are sparse and do not represent an object in the image and the tree is not ordered, meaning that there is no actual root. The first drawback is not related to Kronrod work, since his concern was not image analysis, but rather the study of functions, but the second he solves

only partially, although he does not emphasize the problem: If we fix a point of the square, the components of isolevel sets not containing this point can be ordered relative to this point. What this amounts to do is to isolate some connected component of isolevel set (the one containing the fixed point), and order the other ones relatively to it, giving a rooted tree. From the image analysis point of view, such a construction is not pertinent, since the point is chosen arbitrarily.

In many respects our work is closely related to Kronrod's one. We do not deal with isolevel sets but with connected components of upper and lower level sets, whose holes we fill. The notion of hole is not without flexibility, and we develop an axiomatic approach of the adequate definitions of hole. The fact we fill the holes permits to mix the upper and lower level sets in the same structure, namely a tree, which is oriented by inclusion. In this manner, the tree describes in a straightforward manner the topology of the image. This is related to Kronrod's article in the sense that the boundary of our atoms are (connected parts of) connected components of isolevel sets (at least for a continuous function), and that filling the holes of a connected component of upper level set is exactly the same as filling the holes of its boundary (see Proposition 1.18). In this manner, we precise what is the interior of a connected component of isolevel set, this interior being defined with no arbitrary choice, and this orders the atoms by inclusion. This keeps the advantages of Kronrod's tree, namely contrast and negative invariance properties, while being adapted to image analysis, because most objects in the image are likely to be formed of atoms of our representation. Moreover, we gain generality because the results are valid for a semicontinuous image.

1.4 Tree of shape as an image representation

Now we want to show that, under certain topological conditions concerning the images and their set of definition, the "shapes" have a tree structure. This notion of tree is not the classical one, in the sense that it is not a discrete structure, since it can have an infinite (and possibly not even countable) number of nodes, yet it is consistent with it: two arbitrary nodes are connected, and there is no loop.

The shapes of an image are built from the connected components of level sets. It is well known that connected components of level sets have a tree structure. The difference here is that we consider simultaneously superior and inferior level sets and the shapes constructed from them are stored in one structure, without redundancy. This may seem paradoxic, since the datum of the connected components of lower level sets, or the datum of the connected components of upper level sets, are each sufficient to reconstruct the image. The explanation of this paradox is that the shapes are not constructed from all those connected components, but from a selection of them, this selection being of course independent of the contrast. Moreover, this selection is consistent with what we expect to be "objects" in the image and discards the background. We do not pretend to solve the foreground-background ambiguity in general, but this ambiguity appearing only for regions meeting the frame of the image, most of the time the good choice is made.

The tree of shapes is complete and without redundancy. What these properties mean is that the datum of the shapes is sufficient to reconstruct the image (completeness) and that it is necessary for this operation (absence of redundancy), in the sense that removing a part of the tree does not permit to reconstruct the image or yields a different image. In these respects, the tree of shapes is a representation of the image. Moreover, we believe this tree is a representation adapted to image analysis, its contrast invariance being not the least of its advantages. Finally, for discretely defined images, a fast algorithm allows the decomposition, the reconstruction being trivial. This is exposed in the next chapter.

1.4.1 Basic definition

Unless otherwise defined, Ω will be any connected topological space. We call image an application from Ω to $\mathbb{R}$. Ω will sometimes need to be locally connected. We recall the definition of local connectedness:

Definition 1.1 *A topological space Ω is said to be locally connected if the following equivalent properties hold:*

1. *X has a basis of connected neighborhoods;*

2. *the connected components of any open set of* Ω *are open;*

Notice that local connectedness is a property totally independent of the fact that the topology is metric or not.

The notion of connectedness we use is the classical topological one:

Definition 1.2 (Connectedness) *A topological space X is said to be connected if any partition of* Ω *into two closed sets results in one of them being* ϕ*; and the other one* Ω*. A subset of* Ω *is said to be connected if it is connected as a topological space (for the induced topology).*

This can also be formulated with partitions into two open sets (it is enough to consider the complements), or saying that the only open and closed subsets of Ω are ϕ and Ω, or in an alternative formulation: the only subsets of Ω having ϕ as boundary are ϕ and Ω. Other notions of connectedness exist, as for example arc-wise connectedness, or strong connectedness, but we restrict the discussion to the classical one.

The two most important basic results that are useful are:

1. The union of a family of connected subsets of Ω having a nonempty intersection is connected.
2. If $C \subset \Omega$ is connected and $C \subset D \subset \bar{C}$, then D is connected.

The first point implies that any topological space can be partitioned in a family of maximal connected subsets, and this decomposition is unique. Its elements are called the connected components. The second point implies that if C is connected, $\bar{C}$ is connected, and an easy consequence is that the connected components of a set S are closed in S (but not necessarily open, except when S is locally connected, hence the interest of this notion of local connectedness).

It is clear that the family of superior level sets is decreasing, whereas the family of inferior level sets is increasing:

$$\forall \lambda \leq \mu,\ X^{\lambda}u \supset X^{\mu}u,\ X_{\lambda}u \subset X_{\mu}u \tag{1.4}$$

As explained in Section 1.2, each one of these families allowing to reconstruct the image from Equations 1.2 and 1.3.

1.4.2 From level sets to their components

Whereas contrast invariant, level sets are not compatible enough with our visual perception to have any hope of representing visual "objects". It seems true that the eye is the most at ease in comparing two light intensities (much more than for example in comparing hues), yet these comparisons do not seem to be global: it is able to isolate from two adjacent regions the brighter one, but for non adjacent regions, the comparison does not seem to be reliable (see Figure 1).

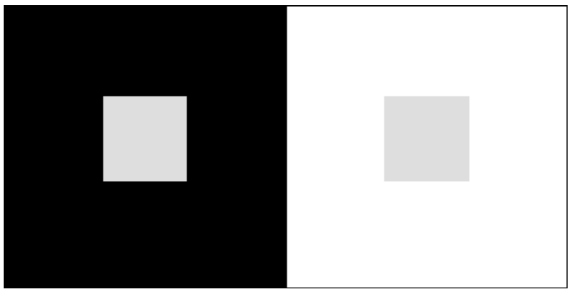


Figure 1: Comparing the two small gray squares, the eye is not at ease comparing their gray level. The left small square might appear brighter than the right one, whereas they have the same brightness.

The consequence is that global comparisons are not meaningful, that is only adjacent regions should be compared. The information left is in Figure 2. The arrows in this figure represent the relation "brighter than". This relation is transitive, but observe that it does not allow to compare the gray levels of the two squares.
Moreover, any homogeneous region appears as one "object", that is not split by the eye. This leads us to work with connected components of level sets rather than with the whole level sets. The fact that two regions are connected components of the same level set is not a relevant information, we do not compare their gray level. This is the case for the small gray squares in Figure 1.

Notation: given a point $x \, in \, X^{\lambda}u$, let us denote by $cc\left(X^{\lambda}u, p\right)$ the connected component of $X^{\lambda}u$ containing x. By convention, if $x \notin X^{\lambda}u$, $\mathrm{cc}\left(X^{\lambda}u, p\right)$ is ϕ. A similar notation applies to $\mathrm{cc}\left(X_{\lambda}u, p\right)$. We derive evidently from Equations 1.2 and 1.3 the reconstruction formulae:

$$u(p) = \inf\left\{\lambda \mid cc\left(X_{\lambda}u, p\right) \neq \phi\right\}$$

$$u(p) = \sup\left\{\lambda \mid cc\left(X^{\lambda}u, p\right) \neq \phi\right\}$$

The monotonicity of level sets translates into a tree structure for their connected components. Since their number need not be finite, we have to define a more general notion of tree.

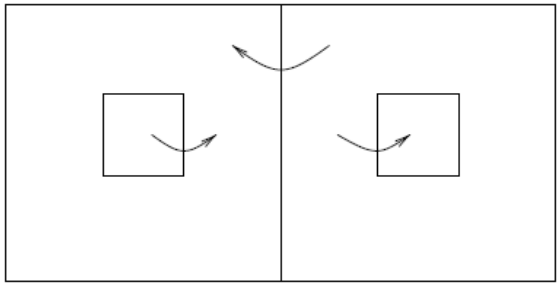


Figure 2: the information left from image of Figure 1 when only local comparisons are performed. The arrows represent the order relation "brighter than".

Definition 1.3: *Let ε be a family of sets and $\prec$ a partial order relation in ε. We say that $\prec$ induces a tree structure in ε if the two conditions hold:*

1. $\exists R \in \varepsilon, \ \forall E \in \varepsilon, \ E \prec R$;
2. $\forall A, B, C \in \varepsilon, \ \left.\begin{matrix} A \prec B \\ A \prec C \end{matrix}\right\} \Rightarrow B \text{ and } C \text{ are comparable.}$

The first condition expresses the connectedness of the structure, R being the root of the tree, and the second condition implies that

there is no loop, because, given four sets $A, B, C, D \in \varepsilon$, the following situation cannot happen:

$$A \subset B \subset D$$

$$A \subset C \subset D.$$

$$B \text{ and } C \text{ not comparable}.$$

A particular case occurs when the relation order is the inclusion of sets, in which case we talk about an *inclusion tree*.

With this definition we show the tree structure of connected components of level sets.

Proposition 1.4: Let u be an image. Let $A = cc(X^{\lambda}u, p)$ (resp. $A = cc(X_{\lambda}u, p)$) and $B = cc(X^{\mu}u, p)$ (resp. $B = cc(X_{\mu}u, p)$). Suppose that $A \cap B = \phi$. Then either $A \subset B$ or $B \subset A$.

Proof. Suppose, without losing generality, that $\lambda \leq \mu$. Then we have $[u \geq \mu] \subset [u \geq \lambda]$, thus $B \subset [u \geq \lambda]$. Let $z \in A \cap B$, then clearly $A = cc(X^{\lambda}u, z)$, and since B is connected, contains z and is contained in $[u \geq \lambda]$, we deduce that $B \subset A$.

The case of the connected components of inferior level sets is dealt with in the same manner. □

This implies (and is stronger than) the inclusion tree structure:

Corollary 1.5: For a bounded image u, the set of lower level sets $X_{\lambda}u$ and the set of upper level sets $X^{\lambda}u$ are each inclusion trees.

Proof: The root is the definition set of u. If A, B and C are lower level sets, $A \subset B$ and $A \subset C$, we get $A \subset B \cap C$, which proves that $B \cap C = \phi$ and, using Proposition 1.4, that B and C are comparable for inclusion order. The proof is similar for $X^{\lambda}u$. □

1.4.3 Beyond components of level set

The above simple result only is a small extension of Equation (1.4). Nevertheless, it is a substantial improvement over these formulas in the sense that it represents more faithfully the objects in the image. We have got locality, which was one of the main

motivations of this work. In these two trees, we expect to find the meaningful objects perceived by the eye. In this sense, these trees seem to be useful for image analysis.

The problem with their use is linked to reconstruction. It is acknowledged that the trees are sufficient information to reconstruct the image they are extracted from, but they are redundant. Since each tree represents exactly the image, if we want to deal as well with upper level sets as with lower level sets (which we do), manipulations of these trees is a problem. For example, the basic operation we would like to do on a tree is to remove one node. Since the other tree is not linked (except that it represents the same image initially), it must be extracted again so that it represents again the image of the first tree. There is no quick solution to this; we have to reconstruct the image from the modified tree and extract the other tree. This drawback is due to the lack of link between the two trees. Whereas the inclusion information is encoded for components of the same type of level set in their tree, there is no such information between components f different types of level sets. This is to be expected since such components are not nested, that is we cannot keep an inclusion tree structure with all components of lower as well as upper level sets.

Figure 3 illustrates the fact that both trees can have very different structures. Since no one should be privileged, the use of their tree structure is a problem. This example hints at what is lacking in both trees. The link between them is related to the notion of holes. In this figure, D is an hole in F and this information is interesting from an image analysis point of view.

Since each tree represents exactly the image, the datum of both is at the same time too much (since there is redundancy) and not enough because such relevant information as the relation of being a “hole” in an object does not appear in these data.

All these problems have a common solution: instead of considering connected components of level sets, we work with connected components of level sets *whose holes are filled*. This elementary operation yields what we call *shapes*. The shapes keep the same properties as connected components of level sets: locality and insensitiveness to contrast change. The relation between connected components of level sets of different types “is a hole in” translates in this framework to the relation “is contained in”. Fortunately, this

operation remains consistent with image analysis. Since we live in a world where numerous objects are “full”, a hole in their projection in an image must be due to occlusion, and representing such projections without their holes is faithful to the true object.
The redundancy between the two trees is automatically removed. Taking the example of image of Figure 3, the shapes based on components A, B, C and E are the same: the whole image.

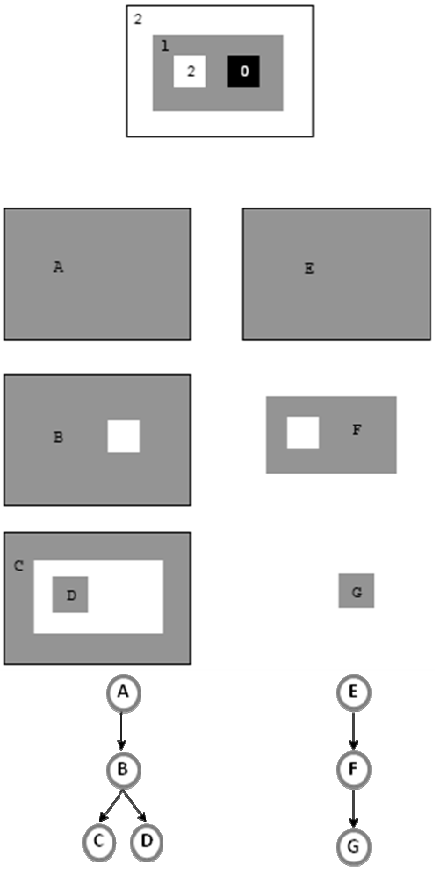


Figure 3: Top: an elementary image with three “objects”: two squares and one rectangle. Left column: the connected components of upper level sets with increasing thresholds from top to bottom. Right column: the connected components of lower level sets with decreasing thresholds from top to bottom. Bottom line: the two associated trees, where arrows represent the relation “contains”. The two squares, which are relevant from an image analysis point of view, are of different types and therefore appear in different trees, showing that both trees are of interest.

Whereas the square *G* is included in the rectangle *F*, there is no link between *D* and *F*.

The shapes based on D and G are D and G themselves, since these components have no hole. On the contrary, the component F become the same rectangle F', and D which was a hole in F, is a subset of F'. As shown in Figure 4, the shapes have an inclusion tree structure. In the following section, we investigate the conditions under which a continuously defined image can be represented by a tree of shapes. This will imply the definition of the notion of hole and of the concept of saturation.

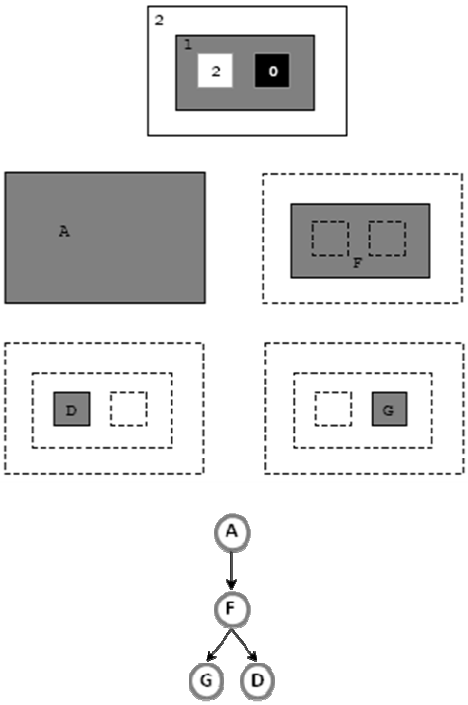


Figure 4: The shapes based on the elementary image as in Figure 3. The component F of Figure 3 becomes full here; D and G do not change since they have no hole, and all the other components become *A*, the whole image. The image is represented by a unique inclusion tree, where upper and lower level sets have equal importance. Notice that reversing the contrast (negating all gray values) would yield the same tree structure.

1.4.4 Saturation, hole and shape definition.

In this section we want to show that the shapes extracted from an image have an inclusion tree structure and to investigate the possibilities of reconstruction of an image from its shapes. Under these conditions, decomposition of an image into shapes will be a powerful image representation, well adapted to image analysis.

Heuristically, the tree of shapes is a data structure to encode in a tree the family of level lines of the image. To be able to handle discontinuous functions, more specifically, upper semicontinuous ones, we define level lines as the external boundary of the level sets of the image. This leads us to the notion of shape which consists in filling the holes of the connected components of the level sets, upper or lower, of u. The operation of hole filling was called saturation in [1], [68]. Thus, level lines are the boundaries of shapes and to give the family of level lines is equivalent to give the family of shapes. It is easy to imagine them when the image is smooth (its graph is a smooth topography).

Definition 1.6: Let Ω be a connected topological space and $A \subset \Omega$. We call hole of A in Ω the components of $\Omega \setminus A$.

Definition 1.7: Let $p_\infty \in \Omega \setminus A$ be a reference point, and let T be the hole of A in Ω containing p_∞. We define the saturation of A with respect to p_∞ as the set $\Omega \setminus T$ and we denote it by $Sat(A, p_\infty)$. We shall refer to T as the external hole of A and to the other holes of A as the internal holes. By extension, if $p_\infty \in A$ by convention we define $Sat(A, p_\infty) = \Omega$. Note that $Sat(A, p_\infty)$ is the union of A and its internal holes.

The saturation operator is the operator that transforms the connected components of level sets to “shapes”. This operator fills the holes of the connected components of level sets.

We will denote by T_A the set of holes of A and $Ext\ A$ the exterior of A. Then we have the identity:

$$sat(A) = A \cup \bigcup_{T \in \mathrm{T}_A} T$$

where the unions are disjoint.

Notice that the definitions of holes and exterior depend on the saturation operator chosen on Ω. But we will never consider several saturations at the same time, so that the context will be clear enough to disambiguate these notions.

Definition 1.8: Given an image u, we call shapes of inferior (resp. superior) type the sets

$$sat\left(cc\left(X_\lambda u, p\right)\right) \ \left(resp.\ sat\left(cc\left(X^\lambda u, p\right)\right)\right)$$

We call shapes of u any shape of inferior or superior type. We denote by $S(u)$ the family of shapes of u.

Examples of interesting saturation operators will be shown later, but here is a trivial one: consider the operator that transforms ϕ; to either ϕ; or Ω and any other set to Ω. This operator destroys all information from the connected components of level sets of an image and inhibits the reconstruction of an image from its shapes, which is one of our concerns.

1.4.5 Saturation of complement

We derive from the definition the essential properties of a saturation operator on a connected topological space Ω.

Definition 1.9: We say that $A \subset \Omega$ is a simple set when A is connected and $sat(A) = A$.

In other words, a simple set is a connected set that has no holes, that is a connected fixed point of sat.

The first result is that a hole in a connected set is a simple set or its saturation is Ω.

Lemma 1.10: Let A be a connected subset of Ω and T a hole in A. Then either T is a simple set or $sat(T) = \Omega$, the last case implying $sat(A) = \Omega$.

Proof: T being a connected component of the complement of a connected set (A) in a connected space, we know that $\Omega \setminus T$ is connected (see [75], IV.3, Theorem 3.3). So this set is either a hole of T, in which case $sat(T) = \Omega$, or the exterior of T, in which case $sat(T) = T$.

If $sat(T) = \Omega$, then since $T \subset sat(A)$, the monotonicity of sat yields

$$\Omega = sat(T) \subset sat(sat(A)) = sat(A).$$

□

This immediately yields

Corollary 1.11: Let A a connected subset of Ω and T a hole in A. Then $sat(T) \subset sat(A)$.

1.4.6 Properties of saturation

We investigate here the topological properties of simple sets, in particular their position relative to their boundary. It appears that pathological situations are avoided when the space Ω is locally connected (see Definition 1.1). Notice that from the idempotency of the saturation operator simple sets are the image by the saturation operator of some sets, in other words, sets that are already saturated. The converse (i.e., the saturation of a set is a simple set) would be true at the condition this saturated set is connected.

Saturation preserves connectedness

First we prove that saturation preserves connectedness. This will be a direct consequence of the following lemma:

Lemma 1.12: Let Ω be a connected topological space. Suppose that Ω is locally connected. If $A \subset \Omega$, A is connected and T is a connected component of $\Omega \setminus \mathrm{A}$, then $A \cup T$ is connected.

Proof. Suppose that $A \cup T$ is not connected. Then A and T being connected, they are the connected components of $A \cup T$. Thus A and T are closed in $A \cup T$, and each one being the complement of the other one in this space, they are also open. Thus, there is an open set U in Ω such that $T \subset U$ and $U \cap A = \phi$. We can suppose U connected, otherwise it suffices to take the connected component of U that contains A (there is one since A is connected), and this component is open since Ω is locally connected. U is then connected, included in $\Omega \setminus \mathrm{A}$ and contains T. Since T is a connected component of $\Omega \setminus \mathrm{A}$, this implies $T = U$, an open set.

As T is closed in $A \cup T$, $\overline{T} \cap A = \phi$, and $\overline{T}$ being connected, $T = \overline{T}$. Since $\phi \neq T \neq \Omega$, the fact that T is open and closed is a contradiction with the connectedness of Ω.

This lemma allows us to show the connectedness preserving property of saturation:

Proposition 1.13: Let Ω be a connected and locally connected topological space, sat a saturation operator on Ω and $A \subset \Omega$ a connected set. Then $sat(A)$ is connected.

Proof. It suffices to write

$$sat(A) = \bigcup_{T \in \mathrm{T}_A} (A \cup T)$$

a union of connected sets (thanks to Lemma 1.12) having a nonempty intersection (A). $sat(A)$ is then connected. □

As a consequence of Proposition 1.13, all properties proved below apply to shapes of any image defined on Ω, since shapes are simple sets.

Saturation preserves topology

Next, we prove that saturation preserves topology:

Lemma 1.14: Let Ω a connected space, sat a saturation on Ω and $A \subset \Omega$. If A is open, $sat(A)$ is also open. If Ω is locally connected and A is closed, then $sat(A)$ is also closed.

Proof. If $sat(A) = \Omega$, the assertions become trivial, so we will suppose this is not the case.

$\Omega \setminus sat(A)$ is a connected component of $\Omega \setminus A$, so that it is closed in $\Omega \setminus A$, which is closed provided A is open. Thus $\Omega \setminus sat(A)$ is closed in Ω, which proves that $sat(A)$ is open.

If A is closed, then $\Omega \setminus A$ is open, and $\Omega \setminus sat(A)$ is a connected component of $\Omega \setminus A$, so $\Omega \setminus sat(A)$ is open (since Ω is locally connected), proving that $sat(A)$ is closed. □

Remark: A direct consequence of Lemma 1.14 is that the only shapes of an upper semicontinuous image u that are of inferior and superior type are ϕ and Ω. Indeed, since connected components of upper (resp. lower) connected components of level sets are closed (resp. open since Ω is locally connected), their saturation is also closed (resp. open). Thus a shape being simultaneously of inferior and superior type would be open and closed, the connectedness of Ω implying this shape would be Ω or ϕ. Remark this becomes false when u is not upper semicontinuous, as shown in Figure 5.

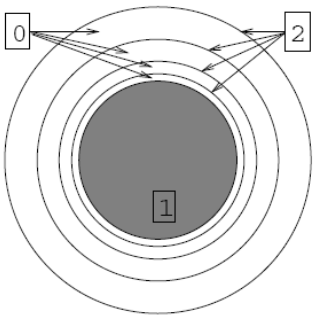


Figure 5: For an image that is not upper semicontinuous, a nontrivial shape can be of inferior and superior type. In this example, the central disk is approximated by a sequence of decreasing circles at level 2, whereas the gaps between circles are at

level 0. This disk is a connected component of X^1u u and X_2u, without holes for the natural saturation of $\mathbb{R}^2$.

Boundary of saturation sets

If A is a set in a topological space we denote with ∂A the boundary of A.

We now show that the boundary of the saturation of a set A is a subset of the boundary of A.

Lemma 1.15 If A is any subset of a locally connected space Ω, and $\{A_i, i \in I\}$ are its connected components, then

$$\bigcup_{i \in I} \partial A_i \subset \partial A$$

Proof. Let $i \in I$. On one hand, we have:

$$\partial A_i \subset \overline{A}_i \subset \overline{A}.$$

On the other hand, $\Omega \setminus A \subset \overline{\Omega \setminus A}$, so that taking the complement of each member we get

$A \supset \Omega \setminus \overline{\Omega \setminus A}$ (*)

Then (**)

the last inclusion coming from the fact that $\overline{A}_i \cap A = A_i$, expressing that A_i is closed in A, since it is a connected component of A. Since $\Omega \setminus \overline{\Omega \setminus A}$ is open and Ω is locally connected, its connected components are also open. Thanks to $(*)$, each connected component of $\Omega \setminus \overline{\Omega \setminus A}$ is contained in a connected component of A. Therefore, $\Omega \setminus \overline{\Omega \setminus A}$ being moreover open, each one of its connected components is contained in the interior of a connected component of A. Thanks to $(**)$, we get

$$(\partial A_i) \cap \left(\Omega \setminus \overline{\Omega \setminus A}\right) \subset \overset{\circ}{A}_i$$

which implies that $(\partial A_i) \cap \left(\Omega \setminus \overline{\Omega \setminus A}\right) = \phi$ since $(\partial A_i) \cap \overset{\circ}{A}_i = \phi$ meaning $\partial A_i \subset \overline{\Omega \setminus A}$. □

Remark: without additional assumptions, the converse inclusion is false. Consider as an example $\Omega = \mathbb{R}$ with the usual topology and

$A = \mathbb{Q}$. Then $\partial A = \Omega$ whereas the connected components of A are composed of one rational, thus for each i, $\overline{A_i} = A_i$ and $\bigcup_i \overline{A_i} = A$. Nevertheless, if I is finite, the fact that the A_i are connected components of A implies $A = \bigcup_{i \in I} A_i$ which is sufficient to prove the converse inclusion.

Proposition 1.16: If Ω is locally connected and $A \subset \Omega$,

$$\partial sat(A) \subset \partial A$$

Proof. If $sat(A) = \Omega$, we get $\partial sat(A) = \phi$ and the result is trivial. Now suppose that $\Omega \setminus sat(A) \neq \phi$.

$\partial sat(A) = \partial\left(\Omega \setminus sat\left(A\right)\right)$ and $\Omega \setminus sat\left(A\right)$ is a connected component of $\Omega \setminus A$. Thus,

$$\partial\left(\Omega \setminus sat\left(A\right)\right) \subset \partial\left(\Omega \setminus A\right)$$

meaning $\partial sat(A) \subset \partial A$. □

The next important result links the saturation of a set to the saturation of its boundary.

Lemma 1.17: Let Ω be a topological space and $A \subset \Omega$ be an open connected set. Then A is a connected component of $\Omega \setminus \partial A$.

Proof. Since A is open, $A \subset \Omega \setminus \partial A$ and moreover

$$A = \overline{A} \cap \left(\Omega \setminus \partial A\right)$$

proving that A is closed in $\Omega \setminus \partial A$, and since it is also open in it and connected, it is a connected component of $\Omega \setminus \partial A$. □

Proposition 1.18: Let Ω a connected and locally connected topological space and $A \subset \Omega$ such that $sat\left(A\right) \neq \Omega$. Then $sat\left(A\right) \subset sat\left(\partial A\right)$, and if A is closed, we get $sat\left(A\right) = sat\left(\partial A\right)$.

1.4.7 Decomposition of an image into shapes

The above results concerning the properties of the saturation operator are the tools needed to prove that shapes have an inclusion tree structure. Nevertheless, this requires additional assumptions on the space Ω, which, as we will see, are met with $\mathbb{R}^n$.

Our first proposition is the easy part of our general theorem, and does not need further hypotheses about Ω. It compares the saturations of connected components of the same type of level set.

Proposition 1.19: Let Ω be a connected and locally connected space and u an image defined on Ω. Let A and B be two shapes of u of the same type such that $A \cap B \neq \phi$. Then either $A \subset B$ or $B \subset A$. It deals with the comparison of the saturations of connected components of level sets of different types. Notice that it involves a strong hypothesis on the boundary of the open shape, which explains why additional hypotheses on Ω are required, so that this hypothesis is automatically satisfied for all open shapes. Notice the proposition is formulated in such a way that the two connected components have one point (p) in common.

Proposition 1.20: Let u be an upper semicontinuous image on Ω, $A = sat\left(cc\left(X^{\lambda}, p\right)\right)$ and $B = sat\left(cc\left(X_{\lambda}, p\right)\right)$ two shapes of u. Suppose also that ∂B is connected. Then either $A \subset B$ or $B \subset A$.

The following lemma deals with the last case: when the connected components of level sets are disjoint.

Lemma 1.21: Let A and B be two disjoint connected sets of a connected and locally connected topological space. Then $sat(A)$ and $sat(B)$) are either nested or disjoint.

The following theorem sums up the three preceding results and is the achievement of this section.

Theorem 1.22: Let u be an upper semicontinuous image on the connected and locally connected space Ω, A and B two shapes of u with connected boundary. Then A and B are either disjoint or nested.

From this result, we can conclude that the set of shapes of an (upper semicontinuous) image has an inclusion tree structure. For simplicity, we assume that our image is discrete. Then we can represent the tree as a finite structure; the shapes are the tree nodes and the parent-child relationship, represented by the links between nodes, is determined by inclusion (the child A being a shape contained in the father A^f with no other shape B such that $A \subset B \subset A^f$).

1.4.8 Unicoherent spaces

As we have seen, the shapes of an image have an inclusion tree structure under some restrictive condition on u: that its shapes have connected boundary. Actually this can be ensured from the upper semicontinuity of u if the definition set Ω is unicoherent. We recall the definition of a unicoherent space:

Definition 1.23: A topological space Ω is said to be unicoherent if it is connected and whatever connected closed subsets F and F' such that $X = F \cup F'$, we have $F \cap F'$ is connected.

Let us give an example of unicoherent spaces. $\mathbb{R}$, and any interval I of $\mathbb{R}$, are unicoherent. Indeed, a connected subset of $\Omega = \mathbb{R}$ or I is an interval. So if Ω is the union of two closed intervals, they intersect and their intersection is a closed interval, thus a connected set. It is harder to prove that $\mathbb{R}^n$ and any hypercube of $\mathbb{R}^n$ are unicoherent. In particular, the closure of a Jordan domain in $\mathbb{R}^n$ is unicoherent, since it is homeomorphic to a hypercube in $\mathbb{R}^n$.

Proposition 1.24: If Ω is a unicoherent and locally connected space, sat is a saturation on Ω and u is an upper semicontinuous image defined on Ω, then all shapes of u have a connected boundary.

We deduce the following

Corollary 1.25: In a unicoherent and locally connected space Ω with a saturation, two shapes of an upper semicontinuous image defined on Ω are either disjoint or nested.

1.4.9 Applications

Until now, we have shown that provided some hypotheses on the topological space Ω are true, the shapes of a semicontinuous image defined on Ω have an inclusion tree structure. But the definition of shapes requires that we have a saturation operator on Ω. The goal of this section is to exhibit saturation operators that are relevant to image analysis. We will do this when Ω is a closed Jordan domain in $\mathbb{R}^n$ (for example a hypercube), for $n \geq 2$.

When the image u is defined only on a bounded subset of $\mathbb{R}^n$, we would like to have a property similar to Theorem 1.22, where shapes

should have an easy interpretation in terms of image analysis. The idea is that only a part of an image defined on $\mathbb{R}^n$ is observed. The first (bad) solution would be to extend the image u to $\mathbb{R}^n$ by an arbitrary value. The problem is precisely that this value is arbitrary, and different values would give different trees.

We would like that "objects" totally included in the definition set are described in the same manner they would be if the whole image on $\mathbb{R}^n$ were observed. So that connected components of level sets not meeting the frame of the definition set are supposed not to be cut. At this condition, whatever the image u outside the definition set, its holes are the components of the complement not meeting the frame. For the same reason, the saturation of a connected set containing the frame is the definition set itself (see Figure 6). There remains to deal with the connected components of level sets that meet the frame without containing it.

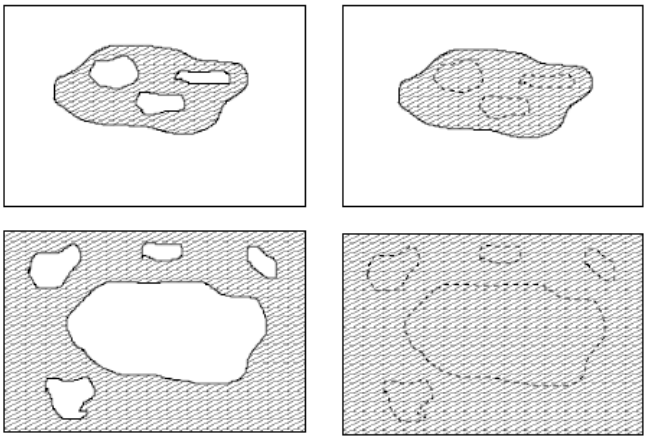


Figure 6: Saturation of some sets in a bounded definition set. Left: two sets (dashed) in their respective image. Right: their saturation (dashed). The top-left set does not meet the frame of the image. It is saturated as if the image were infinite (whatever the image outside the definition set, the result is the top-right set). The bottom-left set contains the frame of the image. It is also saturated as if the image were infinite (whatever the image outside the definition set, the result would contain the whole definition set, shown bottom-right). The saturation of a set containing the frame of the image is always the whole definition set.

The intuitive notion of a hole is that of a connected component of the complement "smaller" than the exterior. When the definition set is $\mathbb{R}^n$, in some sense a bounded set is "smaller" than an unbounded set, so that we can define the holes and the exterior in agreement with the intuition. When u is defined on a bounded set, we quantify this notion with the help of measure theory. Therefore, we need to suppose that we are provided with a measure on the definition set.

We need moreover this definition set to be unicoherent. This imposes strong constraints. We suppose that Ω is the closure of a Jordan domain in $\mathbb{R}^2$, or more generally in $\mathbb{R}^n$, i.e., the closure of the interior of a subset of $\mathbb{R}^n$ homeomorphic to S^{n-1}. Then we know that Ω is a connected and locally connected subset of $\mathbb{R}^n$ $n \geq 2$, and also unicoherent, for the usual topology induced by $\mathbb{R}^n$. We suppose also that a Borel measure μ is given on Ω. Therefore, since Ω is compact, $\mu(x) < \infty$. The boundary of Ω as a subset of $\mathbb{R}^n$, denoted by $\partial\Omega$, is called the frame of the definition set; it is a connected set (the Jordan hypersurface).

From these remarks, we define the saturation as follows:

Definition 1.26: Let A a measurable subset of Ω. We define $sat(A)$ as:

$$\begin{cases} A \cup \{C = cc(\Omega \setminus A) : C \cap \partial\Omega = \phi\} & if A \cap \partial\Omega = \phi \\ \Omega & if\ \partial\Omega \subset A \\ \Omega \setminus \{C = cc(\Omega \setminus A) : C \cap \partial\Omega \neq \phi \text{ and } \mu(C) > \mu(\Omega)/2\} & if\ \ \phi \neq A \cap \partial\Omega \neq \partial\Omega \end{cases} \tag{1.5}$$

The new case is concerned with sets that meet the frame of the image without containing it. The construction of the associated shape is illustrated in Figure 7. That half the area of the image plays a specific role is justified by the fact that this yields a saturation operator.

The fact that we use a Borel measure yields:

Lemma 1.27: If $A \subset \Omega$ is measurable, then every connected component of A and of $\Omega \setminus A$ is measurable.

We are in a position to prove that the sat operator, as defined in 1.28, is indeed a saturation operator.

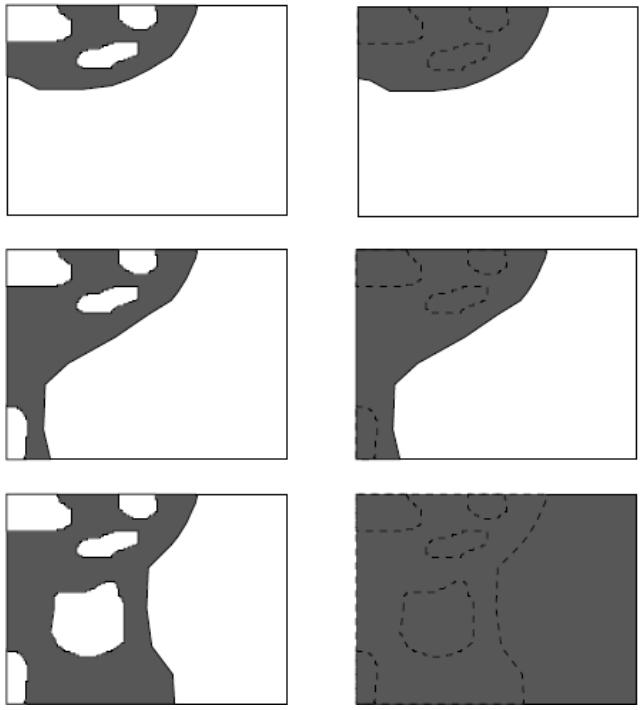


Figure 7: The construction of the shape associated to a set meeting the frame of the image but not containing it. This is the case that was not illustrated in Figure 6. Left: the sets. Right: the associated shapes. In the first two cases (two first rows), one connected component of the complement has a (Lebesgue) measure larger than half the one of the image, this is the exterior of the set. The other connected components are the holes. In the third case (third row), no connected component of the complement has a sufficient measure, they are all considered as holes and the associated shape is the whole image.

Proposition 1.28: The operator $A \subset \Omega$ of Formula (1.5) is a saturation operator on Ω.

This implies that the shapes (according to the saturation operator of Definition 1.26) of an upper semicontinuous image defined on Ω, the closure of a Jordan domain in $\mathbb{R}^n$, $n \geq 2$, have a tree structure.

Chapter 2

Image segmentation based on minimization of Mumford-Shah functional

In this chapter we address the problem of image segmentation. In particular, we introduce a variational approach which use the classical Mumford-Shah functional and is subordinated to the tree of shapes of the image. We carry out the minimization of the functional using a hierarchical processing algorithm. At the end we show some results of segmentation.

2.1 The simplified Mumford-Shah functional on the Tree of Shapes

Let be $u:\Omega \to \mathbb{R}$ an image defined in a domain $\Omega \in \mathbb{R}^2$. The idea of computing a segmentation by selecting a subset of the family of level lines of u can be applied to the simplified version of Mumford-Shah energy functional, leading to a version of it subordinated to the Topographic Map of the image.
According to Mumford-Shah [71], a segmentation of an image u is defined as a pair $(B,\tilde{u})$ where $\tilde{u}$ is piecewise regular function, regular in $\Omega \setminus B$, and B is a the set of boundaries where $\tilde{u}$ is discontinuous. The set of curves B represents a partition of the image domain Ω. In particular, if we assume that $\tilde{u}$ is piecewise constant, then $\Omega \setminus B$ is a union of regions and $\tilde{u}$ takes a constant value on each of them which is equal to the mean value of u on it. We define the simplified Mumford-Shah functional E_{MS}^{λ} as

$$E_{MS}^{\lambda}\left(B,\tilde{u}\,|_{\Omega\setminus B}\right)=\lambda H^{1}\left(B\right)+\int_{\Omega\setminus B}\left(u-\tilde{u}\right)^{2} \tag{2.1}$$

where $H^{1}\left(B\right)$ denotes the length of the system of curves B, $\tilde{u}$ is a piecewise constant image, i.e., constant on each region of $\Omega\setminus B$, and $\lambda>0$ is a parameter. We observe that, given B, the minimum of E_{MS}^{λ} with respect to the variable $\tilde{u}$ is explicitly given by

$$\tilde{u}=\sum_{O_i}u_{O_i}|\chi_{O_i}$$

where

$$u_{O_i}=\frac{1}{|O_i|}\int_{O_i}u\;dp$$

O_i being the connected components of $\Omega\setminus B$ (as usual, for any set O, $\chi_{O_p}=1$ if $p\in O$, $\chi_{O_p}=0$, if $p\notin O$). This observation permits us to write the energy as a function of B and denote it by $E_{MS}^{\lambda}\left(B\right)$ instead of $E_{MS}^{\lambda}\left(B,\tilde{u}\,|_{\Omega\setminus B}\right)$. This energy is a multiscale energy which can be written as $E_{MS}^{\lambda}\left(B\right)\approx\left(C,D,\lambda\right)$ where

$$C\left(B\right)=H^{1}\left(B\right),\; D\left(B\right)=\int_{\Omega\setminus B}\left(u-\tilde{u}\right)^{2}dp$$

Observe that $C\left(B\right)$ is strictly subadditive (See Definition 5 of [73]).

We shall restrict us to the case of digitized images, i.e., we assume that the domain $\Omega=\left\{1,...,N\right\}x\left\{1,...,M\right\}$, $N,M\in N$, and the image $u:\Omega\rightarrow\left\{1,...,L\right\}$, $L\in N$. Let $S\left(u\right)$ be the tree of shapes of u. Observe that any set of shapes $\mathcal{T}\subseteq S\left(u\right)$ can be endowed with a tree structure whose nodes are the shapes in $\mathcal{T}$, two consecutive shapes of $\mathcal{T}$ being related by an edge. Let

$ST(u)=\{\mathcal{T}:\mathcal{T}\subseteq S(u)\}$.

Let us denote

$\partial\mathcal{T}=\cup_{A\in\mathcal{T}}\partial A$

We consider the minimization of (2.1) restricted to the set $\{\partial\mathcal{T}=\mathcal{T}\in ST(u)\}$ i.e.

$$\min_{B=\partial\mathcal{T},\ \mathcal{T}\in ST(u)} E_{MS}^{\lambda}(B) \tag{2.2}$$

Minimizing the simplified Mumford-Shah functional subordinated to the topographic map is a segmentation which contains a similarity criterion and computes regions whose boundaries are level lines. This is not the most general context for a segmentation, since boundaries of objects may be bounded by curves formed by pieces of level lines and may not coincide with full level lines. In spite of this, level lines are robust and contrast invariant objects, and the main edges of the image are contained in them.

Observe that the computation of the optimum has an exponential complexity on the number of shapes if all possible combinations of them are taken into account. This computation becomes feasible if we restrict our search space to a hierarchy of partitions of Ω.

2.1.1 Optimization of a multiscale energy on a hierarchy of partitions

There are several alternative but related strategies to minimize an energy on a hierarchy of partitions, see [28], [43] and [40]. We shall follow here the approach in [6]. Let Ω be the image domain, and let $P(\Omega)$, $Part(\Omega)$ denote the family of subsets and partitions of Ω, respectively.

Definition 2.1: Let $R_0\in Part(\Omega)$. We say that H is hierarchy of partitions of Ω constructed over R_0 if H is a family of nonempty subsets of Ω such that

1. $\Omega\in H$.

2. Any two sets in H are either nested or disjoint.
3. Any set in H contains a set in R_0

A family H' of nonempty subsets of Ω satisfying 2 and 3 is called a pre-hierarchy over R_0.

A cut of H is a partition of Ω whose elements are in H. Figure 8 displays a hierarchy with two possible cuts. The set of cuts of H is the set of partitions of Ω that we can build from H. We shall assume that the hierarchies we consider are finite, i.e., we assume that H has a finite number of elements. In this case, H is a tree whose nodes are the subsets of Ω in H. Two nodes are related by an edge (of the tree) if one is contained in the other and no other set in the hierarchy is in between. The sets in R_0 are the leaves of the tree, Ω is the root, and the concepts of father, children and siblings apply.

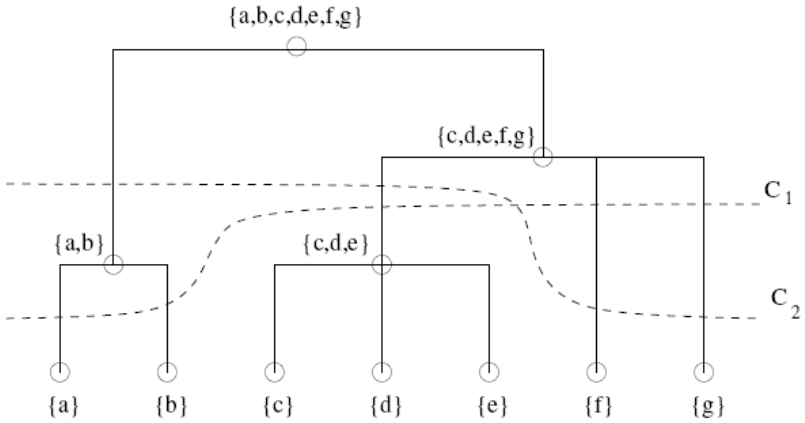


Figure 8: Hierarchy representation in dendrogram form with two possible cuts *C*1 and *C*2 [43].

Definition 2.2: We say that $E^{\lambda}: Part(\Omega) \to \mathbb{R}^{+}$ is an affine energy on $Part(\Omega)$ if there exist two functions $C, D: Part(\Omega) \to \mathbb{R}^{+}$ such that $E^{\lambda}(R) = \lambda C(R) + D(R)$ for any $R \in Part(\Omega)$. In this case, we denote $E^{\lambda} \approx (C, D, \lambda)$.

Definition 2.3: We say that $E : Part(\Omega) \to \mathbb{R}^+$ is separable if there exists a function on the subsets of Ω which we denote by E such that

$$E(\Re) = \sum_{R \in \Re} E(R) \quad \forall \Re \in Part(\Omega).$$

We say that $E : Part(\Omega) \to \mathbb{R}^+$ is subadditive if

$$E(R \cup S) \leq E(R) + E(S) \quad \forall R, S \subseteq \Omega$$

such that $R \cap S = \phi$.

Definition 2.4: Let $E^\lambda \approx (C, D, \lambda)$ be an affine energy. We say that E^λ is a multiscale energy if C, D are separable and C is subadditive. The value λ is called the scale parameter of the energy.

From now on we assume that $E^\lambda \approx (C, D, \lambda)$ is a multiscale energy. We assume that the multiscale energy is defined on the cuts of H . For any λ, let $C_\lambda^*(H)$ be the cut of H minimizing E^λ. Let us review the main ideas of the algorithm proposed by Guigues in [43] to compute $C_\lambda^*(H)$ for any $\lambda > 0$ which is based on a the dynamic programming functional relation.

For each $R \in H$, let

$$H(R) = \{S \in H : S \subseteq R\}$$

We call $H(R)$) the partial hierarchy on the node R. As it is proved in [43], if $R \in C_\lambda^*(H)$ then R is locally optimal in H, that is, $E^\lambda(R) \leq E^\lambda(Y)$ for any cut Y of the partial hierarchy $H(R)$. Let $R_\lambda^*(H)$ the set of nodes of H which are locally optimal in H for the energy E^λ.

Let

$$\Lambda^*(R) = \{\lambda \in R^+ : R \in C_\lambda^*(H)\}.$$

The set $\Lambda^*(R)$ represents the set of scales such that R is in the cut of H minimizing E^λ.

Let

$$\Lambda_{up}^*(R) = \{\lambda \in R^+ : R \in R_\lambda^*(H)\}.$$

The set $\Lambda_{up}^{*}(R)$.represents the set of scales for which R is locally optimal in H for the energy E^{λ}. As proved in [43], $\Lambda_{up}^{*}(R)$is an interval of type $[a,\infty)$.. We denote by $\lambda^{+}(R)$ the left point of the interval and we refer to it as the scale of apparition of R in an optimal cut of the multiscale energy E^{λ}. Then Guigues [43] proved the following result:

Proposition 2.5: For any $R\in H$, $\Lambda_{up}^{*}(R)=\left[\lambda^{+}(R),\lambda^{-}(R)\right)$ where $\lambda^{-}(R)=\min_{S\in H:R\subseteq S}\lambda^{+}(S)$. Thus

$$C_{\lambda}^{*}(H)=\left\{R\in H:\lambda^{+}(R)\le\lambda<\lambda^{-}(R)\right\}.$$

We call the set $\Lambda_{up}^{*}(R)$ the interval of persistence of the region R. The persistent hierarchy obtained from H and E^{λ} is

$$H^{*}=\left\{R\in H:\Lambda_{up}^{*}(R)\neq\phi\right\}.$$

On the persistent hierarchy H^{*} we have $\lambda^{-}(R)=\lambda^{+}\left(R^{f}\right)$ where R^{f} denotes the father of R in H^{*} .

For each $R\in H$, $\lambda\in R^{+}$, we define

$$E(\lambda,R)=\lambda C(R)+D(R).$$

We define the partial energy of the node $R\in H$ as the energy of the optimal cut of $H(R)$with respect to E^{λ} and we denote it by $E^{*}(\lambda,R)$. That is

$$E^{*}(\lambda,R)=E^{\lambda}\left(C_{\lambda}^{*}\left(H(R)\right)\right).$$

Observe that for any leave R of the hierarchy we have $E^{*}(\lambda,R)=E(\lambda,R)$ for any $\lambda\in R^{+}$.

Proposition 2.6: The partial energies $E^{*}(\lambda,R)$of the nodes of H are related by the dynamic programming equation

$$E^{*}(\lambda,R)=\inf\left\{E(\lambda,R),\ \sum_{S\in F(R)}E^{*}(\lambda,S)\right\}\quad\forall R\in H,$$

where $F(R)$is the family of children of R.

Proposition 2.7: Assume that $E^{\lambda} \approx (C, D, \lambda)$ is a multiscale energy on the hierarchy H . Then for any $R \in H$ we have:

1. $E^{*}(\lambda, R)$ is a piecewise affine, nondecreasing, continuous and concave function of λ .
2. We have $E^{*}(\lambda, R) = \sum_{S \in F(R)} E^{*}(\lambda, S)$ if $\lambda < \lambda^{+}(R)$, while $E^{*}(\lambda, R) = E(\lambda, R)$ for any $\lambda \geq \lambda^{+}(R)$.
3. If C is strictly subadditive, i.e., if $C(X) < \sum_{Y \in F(X)} C(Y)$ for any $X \in H$, then $\lambda^{+}(R) \in R$ and is the only solution of $E(\lambda, R) = \sum_{S \in F(R)} E^{*}(\lambda, S)$.

Combining the results of Propositions 2.5, 2.6 and 2.7 we are able to compute the $\lambda - cuts$ $C_{\lambda}^{*}(H)$.

The above algorithm can be implemented once we have the hierarchy as it happens with the algorithms used in [28], [40]. Usually this hierarchy is constructed with a different merging algorithm [40]. On the contrary, the climbing algorithm proposed by Guigues [43] constructs the hierarchy at the same time that it implements the dynamic programming principle of Proposition 2.6.

2.2 Proposed approach

Our approach is based on the construction of the hierarchy from an initial partition using the mergings obtained with a greedy optimization algorithm for the simplified Mumford-Shah energy at several scales. Then we use Guigues algorithm described in Section 2.1 to obtain the minimum of the energy on this hierarchy at any scale λ. This approach can be used for computing multiscale segmentations with the simplified Mumford-Shah energy.

Starting with the initial partition determined by $\partial S(u)$ we construct a hierarchy using the mergings produced by a greedy algorithm applied

to the energy (2.1) at several scales λ_k, $k \geq 1$. The greedy algorithm produces a local minimum of (2.2) and the hierarchy will contain all the merging steps to compute the local minima at several scales. Then by the algorithm described in the last section we compute the global minimum of E_{MS}^{λ} on this hierarchy for any value of λ. Notice that the global optimum corresponding to $\lambda = \lambda_k$ does not necessarily coincide with the local one obtained using the greedy algorithm.

The basic operation of the greedy algorithm is the merging of two neighboring regions which, in the present context is equivalent to the suppression of a shape. Given $\mathcal{T} \in ST(u)$, the suppression of a shape A in $\mathcal{T}$ gives $\mathcal{T} \setminus \{A\} \in ST(u)$. Let us describe this operation as a merging of two regions of $\Omega \setminus \partial\mathcal{T}$. For that, let A^f be the father of A, let $\{B_1, ..., B_p\}$ be the children of A, and let $\{\{A_1, ..., A_k\}$ be the siblings of A. It is implicitly understood that, if A is a leaf of $\mathcal{T}$, the family of children of A is empty. Similarly, it may happen that the family of siblings of A is empty. The shape A determines two regions

$$A^u = \overline{A^f} \setminus \cup \left(A \cup \cup_{i=1}^{k} \overline{A_i} \right)$$

$$A^d = \overline{A} \setminus \cup \left(\cup_{i=1}^{p} \overline{B_i} \right).$$

and the merging of these two regions produces the region (see Figure 9)

$$G(A) = A^u \cup A^d = \overline{A^f} \setminus \cup \left(\cup_{i=1}^{p} \overline{B_i} \cup \cup_{i=1}^{k} \overline{A_i} \right)$$

2.2.1 Merging algorithm

Let us describe the greedy algorithm proposed in [52] and [70] which finds a local minimum of (2.2). Since this algorithm could be applied to any energy, let us denote it by E instead of E_{MS}^{λ}. Let $\Delta E(\mathcal{T}, A) = E(\mathcal{T}) - E(\mathcal{T} \setminus \{A\})$.

Set $\mathcal{T}_0 = S(u)$.

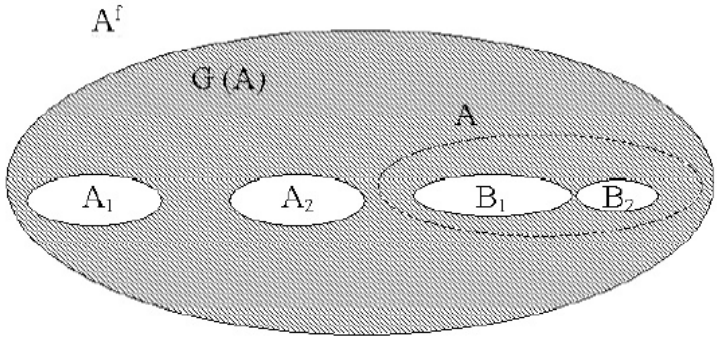


Figure 9: The domain $G(A)$ obtained after suppression of the shape A. It is the region determined by the father of A, denoted by A^f, the external shape in the Figure, the siblings of A, denoted by A_1, A_2 and the children of A, denoted by B_1, B_2.

Step 1: For any $A \in \mathcal{T}_0$ compute $\Delta E(\mathcal{T}_0, A)$ and insert it in a queue Q with priority $\Delta E(\mathcal{T}_0, A)$, the highest priority corresponding to the highest value of $\Delta E(\mathcal{T}_0, A)$.

Step 2: Iterate the following procedure: Choose the shape $A^* \in \mathcal{T}_i$ which corresponds to the first element in the queue constructed in *Step 1* if $\Delta E(\mathcal{T}_i, A^*) > 0$, and define $\mathcal{T}_{i+1} = \mathcal{T}_i \setminus \{A^*\}$. Recompute the values of $\Delta E(\mathcal{T}_{i+1}, A') > 0$ for all shapes A' which are adjacent to A^* (i.e., parent, children, or siblings of A^*) and reorder again the queue in decreasing order of the values $\Delta E(\mathcal{T}_{i+1}, A)$, $A \in \mathcal{T}_{i+1}$ (the highest priority corresponding to the highest value). We stop when no shape A^* exists with $\Delta E(\mathcal{T}_i, A^*) > 0$.

The last tree obtained $\mathcal{T}^*$ determines the boundaries and the regions of the segmentation. It is a local optimal solution of (2.2), in the sense that any other merging of regions of the segmentation increases the energy [52], [70].

2.2.2 Construction of hierarchy

Since E_{MS}^{λ} is a multiscale energy, we can compute its minimum on a hierarchy of partitions using Guigues algorithm [43] (see Section 2.1.1). To explain our construction of the hierarchy of partitions let us recall the definition of completion. Let $R_0 \in Part(\Omega)$ and let H be a pre-hierarchy over R_0. The operation of adding to H a node R constructed by merging two regions of H without father is called a completion. Then we start with the initial partition R_0 determined by $\partial S(u)$ and we take the pre-hierarchy $H' = \{R : R \in R_0\}$. Then we choose $\lambda_1 \geq 0$ and we minimize $E_{MS}^{\lambda_1}$ using the algorithm described in the previous section, adding to the hierarchy H' the completions corresponding to the merging of neighboring regions performed during the execution of the algorithm. Let R_1 be the locally optimal solution obtained. We continue iteratively this process by minimizing the simplified Mumford-Shah energy $E_{MS}^{\lambda_{k+1}}$, $\lambda_{k+1} = 2\lambda_k$, $k \geq 1$, (one could also use $\lambda_{k+1} = \lambda_k + \Delta$, for some value $\Delta > 0$) on the initial partition R_k using the greedy algorithm and storing the successive mergings as nodes of the hierarchy. The construction may be stopped either when the value of λ_k attains a maximum scale value $\lambda_{\max}$, or when we reach the set Ω. The value of λ^+ at each node is computed using Propositions 2.6 and 2.7. Then, using Propositions 2.5, we are able to compute the $\lambda - cuts$ on the constructed hierarchy for any $\lambda > 0$. These $\lambda - cuts$ are local minima of E_{MS}^{λ}; they are also global minima when restricted to the hierarchy. The implementation of this algorithm is based on the results of Guigues [43]. It can be used for computing multiscale segmentations.

2.3 Experimental results

We display some results obtained by minimizing the simplified version of the Mumford-Shah functional E_{MS}^{λ} given by (2.2). The functional E_{MS}^{λ} is minimized on a hierarchy of partitions constructed with the algorithm described in Section 2.2. In order to simplify the nomenclature we call the complete algorithm hierarchy-based algorithm. To construct it we started with the value $\lambda_1 = 2$ and updated it with $\lambda_{k+1} = 2\lambda_k$, $k \geq 1$, up to a maximal scale which gives the region Ω as segmentation.

For each experiment, we shall display the original image, the boundaries of the segmentation B and the image $\tilde{u}$ which takes the mean value of u on each region of the segmentation.

The energy functional is a multiscale one. The value of λ determines the minimal size of the regions of the segmentation [70]. If we do not know a priori this size, by taking ,for instance, $\lambda = 2^k$ we can obtain a multiscale family of segmentations of the image which contain the information at several scales [52], [70]. Figure 10 displays the results obtained minimizing E_{MS}^{λ} (2.2) applied to *Lena* image with several λ values. Figure 10 (a) display the original image. Column at the right displays the set of curves B obtained, and column at the left displays the reconstruction $\tilde{u}$.

Figure 11 shows different segmentation of the same original image with different λ values. Is possible to observe that when the λ value increase the number of regions that compose $\tilde{u}$ decrease and, consequently, the image segmentation is characterize by few details according to the algorithm used. Indeed in this case the number of region of the hierarchy of partition that satisfy the Proposition 2.5 decrease. Vice versa when the λ value is little there are a lot of nodes of hierarchy that satisfy the Proposition 2.5 and, in this case, the image $\tilde{u}$ is composed by a lot of region and, consequently, is possible to observe more details.

Figure 10: Segmentation result of the Lena image obtained minimizing E_{MS}^{λ} with several values of λ using the hierarchy-based algorithm. a) First row: original image. Column at the left: segmentation boundaries. Column at the right: image $\tilde{u}$. b) Second row: λ = 20, c) Third row: λ = 60, d) Forth row: λ = 100.

Figure 11: Image partition with different λ values. a) On top original image. b) Second row: image partition with $\lambda = 20$. c) Third row: image partition with $\lambda = 500$. d) Fourth row: image partition with $\lambda = 2000$.

We select some reasonable value of λ depending on the image for the rest of the experiments. Using a different λ near to the one we used will not change much the results. On the other hand one could filter the hierarchy of partitions so that all regions obtained have a minimal size, but this is not related with the optimization of the functional on the hierarchy.

Figure 12 displays the results obtained minimizing E_{MS}^{λ} applied to the Bureau image in Figure 12 (a) with $\lambda = 60$. Figure 12 (b) displays the set of curves B obtained, Figure 12 (c) displays the reconstruction $\tilde{u}$.

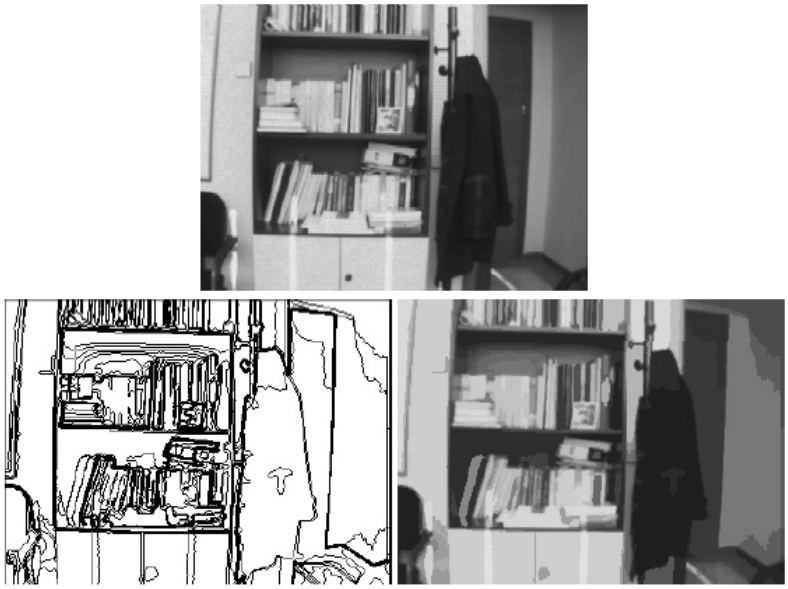


Figure 12: Segmentation result of the Bureau image obtained minimizing E_{MS}^{λ} with $\lambda = 60$ using the hierarchy-based algorithm. a) Top: Original image. b) Bottom left: Segmentation boundaries. c) Bottom right: the image $\tilde{u}$.

Figure 13 displays the results obtained minimizing E_{MS}^{λ} applied to the geographic image in Figure 13 (a) with $\lambda = 200$. Figure 13 (b) displays the set of curves B obtained, Figure 13 (c) displays the reconstruction $\tilde{u}$.

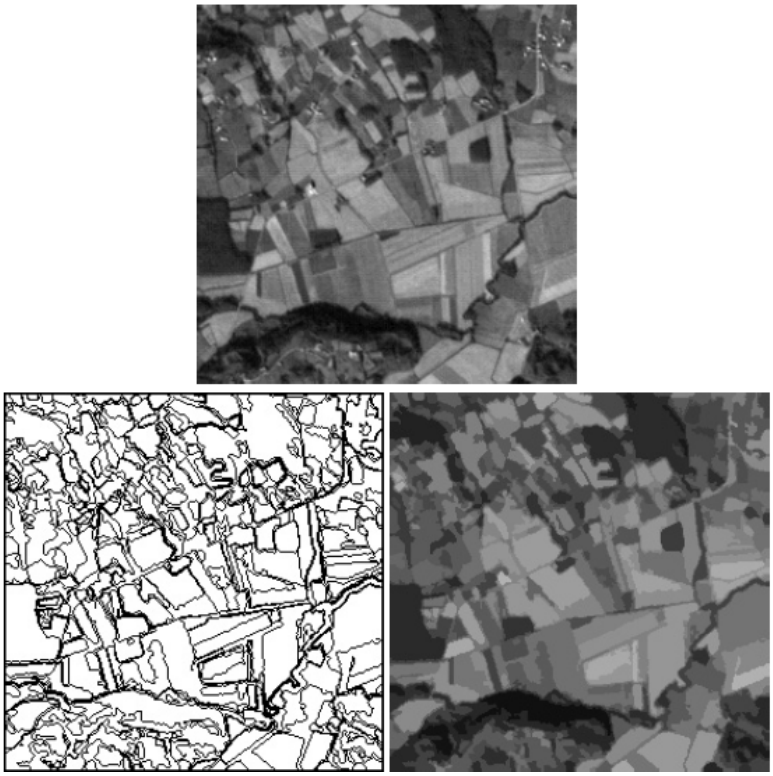


Figure 13: Segmentation result of the geographic image obtained minimizing E_{MS}^{λ} with $\lambda = 200$ using the hierarchy-based algorithm. a) Top: Original image. b) Bottom left: Segmentation boundaries. c) Bottom right: the image $\tilde{u}$.

Figure 14 displays the results obtained minimizing E_{MS}^{λ} applied to the Hamburg Taxi image in Figure 14 (a) with $\lambda = 50$. Figure 14 (b) displays the set of curves B obtained, Figure 14 (c) displays the reconstruction $\tilde{u}$.

Figure 14: Segmentation result of the Hamburg taxi image obtained minimizing E_{MS}^{λ} with $\lambda = 50$ using the hierarchy-based algorithm. a) Top: Original image. b) Bottom left: Segmentation boundaries. c) Bottom right: the image $\tilde{u}$.

Chapter 3

Motion estimation

In this chapter we introduce the motion estimation problem and we review some basic aspects involved in the digital image sequence formation. Some classical and more recent optical flow estimation methods are commented.

3.1 Introduction

Computing the apparent motion of objects in a sequence of images is one of the key problems in video processing known as the optical flow computation. Once computed, the measurements of image velocity can be used in a number of applications in video processing and compression as well as in computer vision [53]. In video compression, the knowledge of motion helps to remove temporal data redundancy and therefore attain high compression ratios [29]. In video processing, motion information is used for deblurring (motion-compensated restoration), noise suppression (motion-compensated filtering) or standard conversion (motion-compensated 3D sampling structure conversion). Tracking moving objects is another important application in video processing. In computer vision, 2D motion usually serves as an intermediary in the recovery of camera motion or scene structure. For references in these topics see [53], [87].

The required features of the motion field are application dependent. Tasks such as the inference of egomotion and surface structure require velocity measurements being accurate and dense, providing a close approximation of the 2D motion filed, whereas motion detection only needs approximated motion field but well located.

Most known motion estimation methods, in one form or another, employ the optical flow constraint which states that the image

intensity remains unchanged from frame to frame along the true motion path. The movement of objects present in the scene may be recovered by minimizing an error measure based on this assumption [87]. However, it is known that motion estimation is an ill-posed problem: the solution can not be unique, or solutions may not depend continuously on the data [10].

Current approaches try to solve this issue by imposing additional assumptions about the structure of the 2D motion field. These constraints are introduced into the error measure either by adding a smoothness term to it, or by restricting it to a particular motion model. The former strategies are called dense motion field estimation approaches, whereas the latter ones are usually called parametric motion estimation approaches.

Classical methods for dense motion field estimation seek for a motion field that satisfies the optical flow constraint with a minimum pixel-to-pixel variation between the flow vectors (smoothness term). Parametric motion estimation methods usually consider a partition of an image into disjoint regions and estimate the motion of these regions restrained them to parametric motion models. We will review dense and parametric methods in Section 3.4.

The optical flow constraint assumption is generally violated in image sequences taken from the real world. Global or local changes in illumination due to, for instance, a moving camera or a change in the shade of an object can make the optical flow constraint to fail. Alternatives to the classical brightness constancy assumption have been already proposed in the literature (see Section 3.4.2).

3.2 Geometric image formation

Imaging systems capture time-varying $3D$ scenes as $2D$ projections. These projections can be represented by a mapping from $4D$ space to a $3D$ space, $m:\mathbb{R}^4 \to \mathbb{R}^3 : (X,Y,Z,t) \to (x,y,t)$ where (X,Y,Z) are the $3D$ world coordinates, (x,y) are the $2D$ image coordinates, and t is the time, and all of them are continuous

variables. These projections can be *perspective* or *orthographic* [49], [87] and entail some loss of depth information, which engenders several problems such as aperture and occlusions (see Section 3.3).

Perspective projection (or central projection) is the projection of points in the scene onto the intersection of the image plane with the ray connecting the points and the focal point (or center of projections) C. We consider the image-centered coordinate system, where the image plane is parallel to the $XY-plane$ of the $3D$ world coordinates and the focal point C is a distance l away from the image plane on the negative side of the $Z-axis$. That is, C is placed on $(0,0,-l)$. The distance from the focal point to the image plane, l, is called focal length. In Figure 15 perspective projection is illustrated for this configuration.

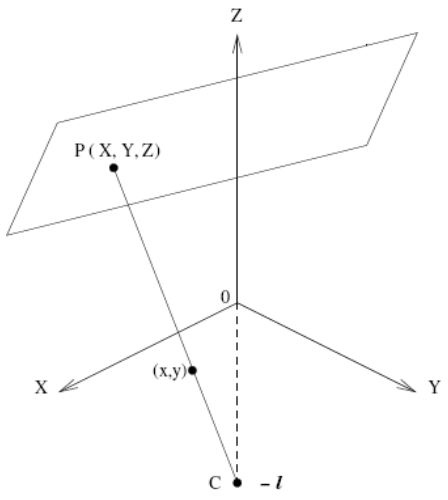


Figure 15: Perspective projection model

The perspective transformation for this configuration gives the following relations:

$$x = \frac{lX}{l+Z} \text{ and } y = \frac{lY}{l+Z}.$$

Orthographic projection (or parallel projection) is the projection by parallel rays orthogonal to the image plane. See Figure 16 for an example of orthographic projection when image-centered coordinate system is considered.

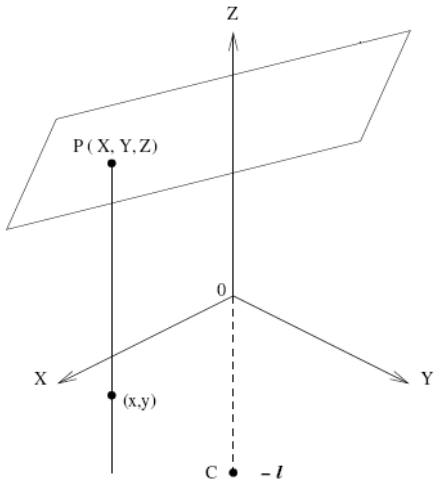


Figure 16: Orthographic projection model

The orthographic projection can be described as $x = X$ *and* $y = X$.

Note that the orthographic projection corresponds to the limit case of the perspective projection when $l \to \infty$. It appears when the objects are, compared with the focal length l , very small or located very far away from the viewer (i.e., $l / Z \to 1$, where Z denotes the $Z - coordinate$ of the object in the world coordinate system).

The perspective projection produces a distortion of angles and distances. The size of the view will vary when the relative positions of the eye, the image plane, and the object are altered. In the case of orthographic projection the size of the view of the object will not vary with the distance between the object and the image plane; projected parallel straight lines stay parallel; distances and angles are transformed consistently. It is usual to replace the perspective projection (non-linear transformation) by orthographic projection (linear transformation) if it is possible.

3.3 2D Motion estimation

Image video sequences are $2D$ projections of $3D$ scenes at different time instants. Thus, $2D$ motion refers to the projection of the $3D$ motion of objects and camera onto the image plane. In Figure 17 the point P_t moves with velocity W_t to point $P_{t'}$. The $3D$ motion is projected over the image plane by a perspective projection. The corresponding image point p_t moves on the image plane with velocity w_t to point $p_{t'}$.

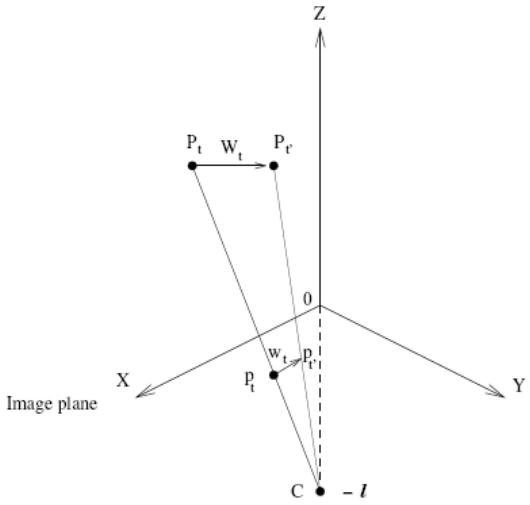


Figure 17: 3D motion projection

The presence of $2D$ motion manifests itself on the image plane by changes of the intensity values of the pixels along time. These changes are referred as optical flow field or apparent motion field. The optical flow field is, in general, different from the $2D$ motion field due to the following effects:

1. **The 2D motion field may not always be observable:**
   Lack of sufficient spatial image gradient may produce an unobservable motion; think, for instance, in the motion generated by a circle with uniform intensity which rotates

about its center. This motion is unobservable. The same thing may happen when an image made by a periodic structure moves and stays unchanged after the motion. In Figure 18 we display the two examples mentioned previously. This problem is a particular case of the aperture problem.

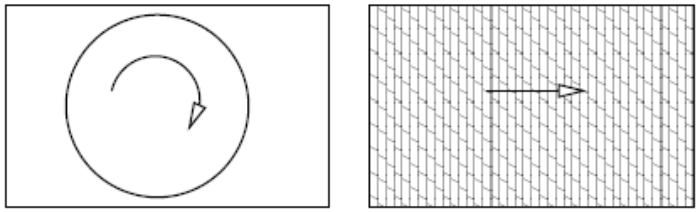


Figure 18: Examples of projected motion that do not generate optical flow.

2. **An observable motion may not always corresponds to an actual motion:**
   If external illumination varies from frame to frame, then a change will be observed in the sequence image intensity even though there is no motion. In Figure 19 we can see the effect of an illumination change over a scene. In this example the shades change the geometry of the image. This problem may appear in satellite images, and is difficult to solve. However, generally the situation is not so dramatic as in this example of optical flow field.

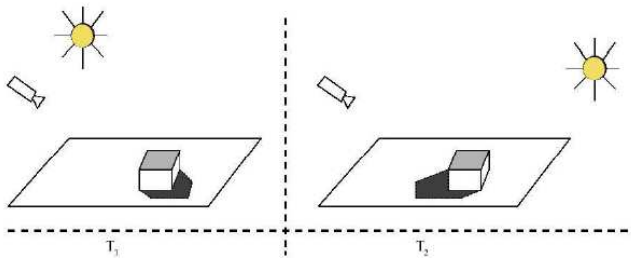


Figure 19: Example of projected objects that generate optical flow.

Therefore, since only the optical flow field can be observed, generally it is assumed that the estimated optical flow corresponds to the $2D$ motion field. Thus, the objective of motion estimation techniques is to estimate the optical flow field. Figure 20 shows an example.

a) b) c)

Figure 20: Optical flow estimation example: a) wheel at time t, b) wheel at time $t+1$, c) estimated optical flow field.

Optical flow constraint

As we have mentioned in the Introduction, the most usual assumption is that image intensity remains constant along the motion trajectory (optical flow constraint). This assumption involves that the intensity changes are due exclusively to motion, scene illumination is constant, and the object surface is Lambertian.

Let $I:[T_0,T_1]x\Omega \to \mathbb{R}$ be an image sequence with rectangular spatial domain $\Omega \subset \mathbb{R}^2$ and time interval $[T_0,T_1]$. The optical flow constraint can be written as:

$$I(t,\ x,\ y) = I\left(t+\Delta t, x+u(t,x,y), y+v(t,x,y)\right) \qquad (3.1)$$

where $t,\ t+\Delta t \in [T_0,T_1]$ are two different time instants, and $\left(u(t,x,y),\ v(t,x,y)\right)$ is the optical flow field. When no confusion arises (t,x,y) will be dropped out.

Assuming that the displacements (u,v) are small or that the image changes slowly in space, this constant-intensity assumption leads to the linearized optical flow constraint, which is called the *Optical Flow Equation* (OFE).

$$\partial_t I + \partial_x Iu + \partial_y Iv = 0 \qquad (3.2)$$

where ∂_x, ∂_y and ∂_t denote the partial derivatives with respect to x, y and t respectively.

Ill-posed problem

The 2D motion estimation problem based only on two frames and constrained only by Equation (3.1) is an *ill-posed* problem in the absence of any additional assumptions about the nature of the motion. A problem is called ill-posed if a solution is not unique, or does not exist, or the solution do not continuously depend on the data [87]. Estimation of 2D motion has the existence, uniqueness and continuity problems:

- **Existence of a solution**: No correspondence can be established for covered/uncovered points. This is known as the *occlusion* problem. This concept is illustrated in Figure 21 where the object indicated by the solid lines translates in the x direction from time t to $t+1$. The dotted region in the frame t indicates the background to be covered in frame $t+1$. Thus, it is not possible to find correspondence for these pixels in frame $t+1$. The dotted region in frame $t+1$ indicates the background uncovered by the motion of the object. Clearly, there is no correspondence for these pixels in frame t.

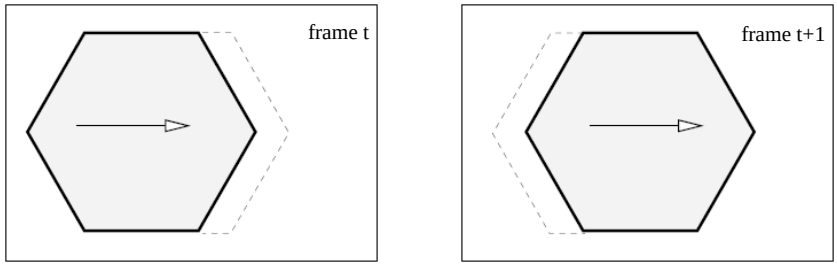


Figure 21: the occlusion problem

- **Uniqueness of the solution**: If the components of the displacement at each pixel are treated as independent

variables, then the number of unknowns (two vector components) is twice the number of observation (3.1). This leads to the so-called *aperture* problem. In such cases, we can only determine motion that is in the direction of the spatial image gradient, called the *normal flow*, at any pixel: we denote $\boldsymbol{w} = (v, v)$ and write the OFE (3.2) as

$$\nabla I \cdot \boldsymbol{w} = -I_t$$

where the operator nabla denotes the spatial gradient $\nabla I = \left(\partial_x I, \partial_y I\right)^T$. The optical flow field $\boldsymbol{w}$ can be decomposed into $\boldsymbol{w} = \boldsymbol{w_n} + \boldsymbol{w_t}$, where the normal flow $\boldsymbol{w_n}$ and the tangential flow $\boldsymbol{w_t}$ are respectively parallel and perpendicular to ∇I. Then, the OFE becomes

$$\nabla I \cdot \left(\boldsymbol{w_n} + \boldsymbol{w_t}\right) + I_t = \nabla I \cdot \boldsymbol{w_n} + I_t = 0$$

Hence, only the normal flow can be determined and is given by

$$\boldsymbol{w_n} = w_n \frac{\nabla I}{\|\nabla I\|} \text{ where } w_n = -\frac{I_t}{\|\nabla I\|}$$

In Figure 22 we display an illustration of the aperture problem. In particular, we consider an object which moves following the vector $\boldsymbol{w}$ up to the line L. If we estimate the motion vector based on the point $\boldsymbol{p}$, then it is not possible to determine which of the vectors painted over $\boldsymbol{p}$ corresponds to the motion of the object, and only the normal flow can be determined. The same problem may appear if we estimate the motion based on a neighborhood of pixels which has uniform gray level patches. This is the case of the window of the Figure 22 indicated as Aperture 1. However, the aperture problem may be overcomed by considering a neighborhood that contain sufficient gray-level variation. This is achieved in the second window of the Figure, indicated as Aperture 2.

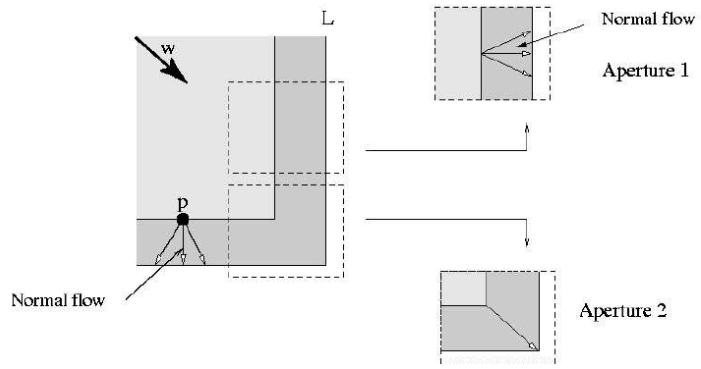


Figure 22: The aperture problem

- **Continuity of the solution**: Motion estimation is highly sensitive to the presence of noise in video images. Even, small amount of noise may result in large deviations in the estimates.

These problems can be solved using different restriction over the optical flow as will be seen in next Section.

3.4 Optical flow estimation method

One of the first works about the optical flow estimation problem was developed in [10]. The algorithm proposed in this paper is based on the OFE. In particular, it follows the variational approach which define the following energy to be minimized over the whole image domain Ω

$$\min \int_{\Omega} \left(\partial_x Iu + \partial_y Iv + \partial_t I \right)^2 \tag{3.3}$$

Many other works have been developed based on the same energy function. These methods are called *differential techniques*, because the time and space derivatives of the image intensity function are needed for estimating the motion.

Equation (3.3) is not sufficient to uniquely specify the $2D$ motion field (aperture problem). Current motion estimation approaches try to solve the latter issue by imposing additional regularizing assumptions

about the structure of the $2D$ motion field. They can be classified into two groups: *dense motion estimation techniques*, and *parametric motion estimation techniques*. The first ones introduce additional constraints into the error measure by adding a smoothness term to it, whereas the second ones restrict the error measure to a particular motion model. In the first case, the domain is the whole image and each pixel has a displacement vector associated. In the second case, the domain can be a different size window or an arbitrary region whose pixels follow the same motion model. The latter methods may be also called *region-based* approaches. Furthermore, the dense motion estimation techniques are also classified into global and local techniques according to the involved smoothness term [18]. The smoothness term is also called regularization term. In global methods (for instance, Horn and Schunck method) the regularization term is applied globally on the whole image domain, whereas in local methods (for instance, Lucas and Kanade method) it is applied locally on a neighborhood. We discus below more details of the methods.

Before continue, remark that we denote $\phi(x, y)$ the general transformation of pixel (x, y) and in the dense motion estimation techniques we can write $\phi(x, y) = (x+u, y+v)^T$.

3.4.1 Dense Motion Estimation techniques

Global dense motion field estimation approaches yield flow fields with 100% density, but are experimentally known to be more sensitive to high gradient noise, as we will see later in this Section. These methods differ mainly in the particular smoothing strategies adopted.

Horn and Schunck method:

The Horn and Shunck method [45] is a classical method for dense motion field estimation. It attempts to determine the optical flow vector field (u, v) based on two assumptions:

- The OFE (3.2) is satisfied.
- The optical flow vector field varies smoothly from pixel to pixel. This can be expressed by requiring the integral

$$\int_{\Omega}\left(\|\nabla u\|^2+\|\nabla v\|^2\right)dxdy \tag{3.4}$$

to be minimum. Recall that $\nabla=\left(\partial_x,\partial_y\right)$.

The energy functional to be minimized includes both constrains, and is defined as

$$E_{HS}=\int_{\Omega}\left(\partial_x Iu+\partial_y Iv+\partial_t I\right)^2+\alpha^2\left(\|\nabla u\|^2+\|\nabla v\|^2\right)dx\,dy$$

where $\alpha>0$ is the weight to control the influence of the smoothness constraint. Larger values of α^2 result in stronger penalization of large flow gradients and lead to smoother flow fields. This functional is well-posed (as established Schnörr [82]). Thus, it has a unique minimizer that implicitly entails an interpolation process: at locations where $\|\nabla I\|\approx 0$, no reliable local flow estimate is possible, but the regularizer of Equation (3.4) fills in information from the neighboring flow. This method is classified as a global technique, since the used regularization term is global. Note that this method and, in general, global dense motion field estimation methods may be sensitive to high gradient noise as is discussed in [18].

Lucas and Kanade method:

Lucas and Kanade [48] proposed to estimate the motion of a pixel by assuming that the motion vector associated to the OFE remains unchanged in a neighborhood of the pixel. Thus, the method allows to estimate a translational motion vector for that block and assign this vector to the pixel.

The authors propose to determine u and v, at some location (x,y) and time t, from a weighted least-square fit by minimizing the functional:

$$E_{LK}=K_\sigma *\left(\partial_x Iu+\partial_y Iv+\partial_t I\right)^2$$

where K_σ represents a neighborhood of (x,y) of size σ. The window function K_σ may be a Gaussian with standard deviation σ. Let us remark that in this case the regularization term is applied locally on a neighborhood. Thus, this method is classified as a local technique. A sufficiently large value for s makes this method robust under noise.

A minimum (u,v) of E_{LK} satisfies the equations $\partial_u E_{LK} = 0$ and $\partial_v E_{LK} = 0$. This gives the system

$$\begin{pmatrix} K_\sigma * (\partial_x I)^2 & K_\sigma * (\partial_x I \partial_y I) \\ K_\sigma * (\partial_x I \partial_y I) & K_\sigma * (\partial_y I)^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -K_\sigma * \partial_x I \partial_t I \\ -K_\sigma * \partial_y I \partial_t I \end{pmatrix}$$

which can be solved if its symmetric matrix is invertible. This is not the case of flat regions, where the image gradient vanishes. In some other neighborhood K_σ, the smallest eigenvalue of the system matrix may be close to zero and consequently, the data does not allow a reliable determination of the full optical flow. This is a form of the aperture problem mentioned earlier.

Nagel and Enkelmann method:

The method of Horn-Schunck imposes the OFE and the smoothness constraint globally over the whole image. As a consequence, the flow is also smoothed across motion boundaries. Thus, the result is a blurry flow field which is ignorant of the true motion boundaries. This is an important drawback of this method. The first modification to alleviate this problem has been proposed by Nagel. In [74], he introduces the oriented smoothness (or image driven) constraint, which imposes that optical flow field should vary piecewise smoothly in space.

They formulate the problem as the minimization of the functional:

$$E_{NE} = \int_\Omega (\partial_x I u + \partial_y I v + \partial_t I) + \alpha^2 \left[(\nabla u)^T D(\nabla I) \nabla u + (\nabla v)^T D(\nabla I) \nabla v \right] dx\, dy$$

where $D(\nabla I)$ is a projection matrix perpendicular to ∇I defined as:

$$D(\nabla I) = \frac{1}{\|\nabla I\| + 2\delta} \begin{pmatrix} (\partial_y I)^2 + \delta & -(\partial_x I)(\partial_y I) \\ -(\partial_x I)(\partial_y I) & (\partial_x I)^2 + \delta \end{pmatrix}$$

Here, δ serves as regularization parameter that prevents the matrix $D(\nabla I)$ from getting singular.

Using this new functional to computer the optical flow, the diffusion across image boundaries with large $\|\nabla I\|$ is reduced. This

attenuates the variation of the flow in the direction of the spatial gradient. Well - posedness for this functional has been established in [82].

Weickert et al. methods

Weickert et al. [84], [14], [5] and [88] are currently interested in the study of improved optical flow estimation techniques based on Horn-Schunck and Nagel methods. They noted that the smoothness term proposed by Nagel has an important drawback: in specific situations image discontinuities may not coincide with flow discontinuities. For instance, in the case of an image containing strongly textured objects, it has many texture edges which are not motion boundaries. Thus, the previous method may lead an over-segmentation flow. In such cases, a smoothness term which respects flow discontinuities instead of image discontinuities is desirable.

Therefore, the authors of [83] and [84] introduced the flow-driven smoothness term by replacing the quadratic smoothness term of Equation (3.4) by the following term

$$\int_\Omega \rho\left(\|\nabla u\|^2 + \|\nabla v\|^2\right)\, dx\, dy$$

where $\rho\left(s^2\right)$ is a robust function. In this case, the modified energy functional is

$$E_W = \int_\Omega \left(\partial_x Iu + \partial_y Iv + \partial_t I\right)^2 + \alpha^2 \rho\left(\|\nabla u\|^2 + \|\nabla v\|^2\right) dx\, dy \qquad (3.5)$$

The function ρ has an associated function $\psi\left(s^2\right) = \rho'\left(s^2\right)$ that is called the *influence function* [4] or *diffusivity function* [84] and controls the activity of the Euler-Lagrange equations of the Functional (3.5). In [88], for instance, the following regularizer have been considered: $\rho\left(s^2\right) = \sqrt{s^2 + \varepsilon^2}$, where the parameter ε serves to ensure that the function ρ is differentiable $\forall s \in \mathbb{R}$, and the influence function is

$$\psi\left(s^2\right) = \frac{1}{2\sqrt{s^2 + \varepsilon^2}}$$

Observe that this function takes small values for large arguments. Then, this choice of ρ for the Functional (3.5) penalizes diffusion

across flow discontinuities and, consequently, helps to preserve them in a better way. Moreover, in [83] a complete review and classification of rotation invariant convex regularizers can be found.

In [14] the authors apply a robust function ρ over the data term, as well as the spatio-temporal smoothness term. The data term is derived from the intensity constancy assumption and a gradient constancy assumption. Note that incorporating gradients, geometry information is considered. Furthermore, the approach is embedded in a coarse-to-fine strategy. Finally, a very interesting combination of Horn-Schunck method and Lucas-Kanade method with an efficient implementation based on multigrid schemes has been proposed in [18]. Multigrid strategies are developed in several areas included motion estimation [36] and [63]. The multigrid framework [44] is nowadays an active and important subject of research.

3.4.2 Parametric Motion Estimation techniques

Parametric motion estimation approaches may offer relatively high robustness under noise, but the estimated motion is constrained to motions described by the specific model. These techniques are well suited when there is enough confidence that the underlying structure behaves as the enforced model.

Models for Motion Representation

Motions present on an image sequence can be described in a parametric form using a finite, usually small, number of parameters. Since the $2D$ motion results from the projection of $3D$ moving objects onto the image plane, a model for $2D$ motion fields can be derived from models describing $3D$ motion, $3D$ surface function and camera projection geometry. Note that identical $2D$ motion models may result from different assumptions about $3D$ motion, surface and camera projection models [87], [49] and [53].

We shall assume that motion of objects in an image sequence can be modeled locally with an affine model. An affine transformation $\phi(x, y)$ of a point (x, y) is described as:

$$\phi(x,y)=\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}+\begin{pmatrix} e \\ f \end{pmatrix}. \tag{3.6}$$

To understand the incidence of $3D$ subjacent motion on each term of this Equation we refer to Chapter 7 of the book [49], where there is a very accurate study of the transformations that derives from the projective geometry when observing the movement of a planar surface. We give some of the ideas.

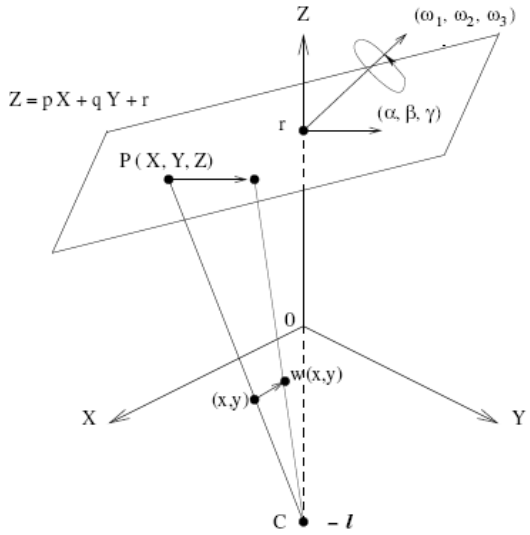


Figure 23: A planar surface moving with translation velocity (α,β,γ) at $(0,0,r)$ and rotation velocity $(\omega_1,\omega_2,\omega_3)$ around it. An optical flow is induced on the image plane by perspective projection from the focal point $(0,0,-l)$.

We consider a planar surface moving in the scene given by the equation: $Z = pX + qY + r$, where p,q are the gradient of the surface and r designates the distance of the surface from the image plane along the $Z-axis$. In Figure 23 we display a planar surface moving with translation velocity (α,β,γ) at $(0,0,r)$ and rotation velocity $(\omega_1,\omega_2,\omega_3)$ around it. An optical flow is induced on the image plane by perspective projection from the focal point $(0,0,-l)$. This optical flow is denoted $\boldsymbol{w}$ and is given by

$$\boldsymbol{w}(\boldsymbol{p}) = \begin{pmatrix} E \\ F \end{pmatrix} + \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} G \\ H \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix}$$

The flow $\boldsymbol{w}$ uses a different notation from one in Equation (3.6), since $\boldsymbol{w}$ only models the flow vectors, and not the full transformation. To obtain the transformation as is described in ϕ, we just need to add the original vector $\boldsymbol{p} = (x, y)$, we use the identity denoted by Id, for that purpose.

$\phi(\boldsymbol{p}) = \boldsymbol{w}(\boldsymbol{p}) + Id(\boldsymbol{p})$.

That is,

$$\phi(p) = \begin{pmatrix} E \\ F \end{pmatrix} + \begin{pmatrix} A+1 & B \\ C & D+1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} G \\ H \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.7}$$

The flow parameters A, B, C, D, E, F, G, and H are related with the motion parameters α, β, γ, ω_1, ω_2, ω_3 and the surface (p, q, r) by following equations:

$$E = \frac{l\alpha}{l+r},\ F = \frac{l\beta}{l+r}$$

$$A = p\omega_2 - \frac{p\alpha+\gamma}{l+r},\ B = q\omega_2 - \omega_3 - \frac{q\alpha}{l+r}$$

$$C = -p\omega_1 + \omega_3 - \frac{p\beta}{l+r},\ D = -q\omega_1 - \frac{q\beta+\lambda}{l+r}$$

$$G = \frac{1}{l}\left(\omega_2 + \frac{p\gamma}{l+r}\right),\ H = \frac{1}{l}\left(-\omega_1 + \frac{q\gamma}{l+r}\right)$$

The parameters E and F are zero order parameters, A, B, C, D are first order parameters and the G, H are second order parameters. It is also important to notice that r and l are, respectively, the distance of the surface to the camera and the focal length and must be treated as constants.

From the projective model in Equotion. (3.7) we can obtain simpler models. If we consider an orthographic projection, i.e. the case $l \to \infty$, the second order parameters G and H are 0. If we also assume there is not any $3D$ rotation in the scene, i.e. $(\omega_1, \omega_2, \omega_3) = (0, 0, 0)$, then the first order parameters A, B, C and D

are also 0. Therefore, if the second order parameters vanish, we obtain an affine model with six parameters; whereas, if the first and second order parameters vanish, we lead to a translational model with two parameters.

In Figure 24 we display four examples of parametric motion vector fields which correspond to a translational model (a), an affine model (b), a projective model (c) and a model induced by perspective or orthographic projecton of rigid motion of curve surfaces (d). The corresponding motion-compensated predictions of a centered square [53] are displayed in the column at the right.

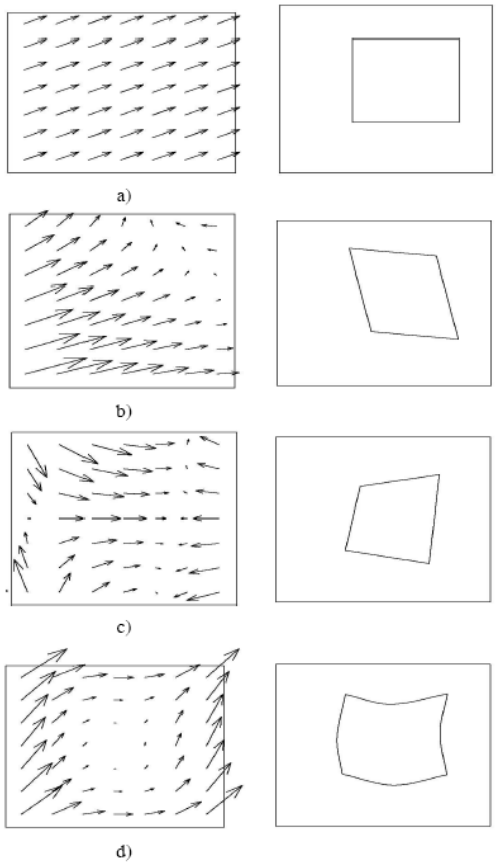


Figure 24: Examples of parametric motion vector fields (sampled) and corresponding motion-compensated predictions of a centered square [53].

These models can be used efficiently for the estimation, interpretation and transmission of certain classes of motion fields. Observe that more model parameters imply more complexity in the functional minimization, but more precision, whereas less parameters imply more computational simplicity and more robustness, but less precision. We conclude that the affine motion model (induced by orthographic projection of rigid motions of planar surfaces) gives a good trade-off between complexity and representativeness.

Parametric motion estimation methods:

In [35] a region-based affine motion estimation method for the identification of $2D$ and $3D$ motion models in image sequences is presented. In this case two images at two different time instants, generally consecutive, of an image sequence are taken. The first of them is partitioned into disjoint connected regions.

These regions are assumed to be extracted from the image using a particular partitioning strategy, such as a luminance homogeneity criterion. Matching of regions is carried out by minimizing a cost functional based on the brightness constancy assumption. In particular, the functional is the mean square reconstruction error after motion compensation

$$E = \sum_{(x,y)\in\mathbb{R}} DFD^2\left[x, y, \phi\right]$$

where R is the region, and DFD denote the displaced frame difference: $DFD\left[x, y, \phi\right] = I\left(t, x, y\right) - I\left(t+1, \phi\left(x, y\right)\right)$. Their approach assume an affine motion model for each region. Moreover, the technique is embedded in a multiresolution scheme in order to improve the robustness of the method.

Robust motion estimation

Assumptions about the world embedded into algorithms for recovering optical flow (like constant intensity) are, necessarily, simplifications and hence, will be violated. Therefore, at the same time that realistic constraints that avoid model violations are formulated, other optical flow estimation approaches are developed with the goal of performing well even when violations are present. These approaches are called robust methods, and their goal is to detect and reduce the violations of the adopted assumption.

In motion estimation algorithms presented in the previous Section, there is an assumption which is implicitly derived from the brightness constancy and spatial smoothness assumptions: in a finite image region only a single motion is present. However, when a region contains pixels of two different objects, multiple motions may appear within this region and cause violations of these assumptions. The large error values produced are called *outliers*.

Note that these gross errors may be arbitrarily large and therefore cannot be averaged out, as is typically done with small-scale noise. A popular robust technique is based on the known *M-estimators* [46]. Let $X = \{p_i\}$ be a set of data points and let m be a k-dimensional parameter vector to be estimated. The objective functions used in robust estimation are defined in terms of an error distance or residual function, denoted by $f_i = f(p_i, m),\ (p_i \in X)$ that may correspond, for instance, to the displaced frame difference (DFD). The standard least-squares method tries to minimize the quadratic error, $\sum_{p_i \in X} f_i^2$, which is unstable if there are outliers present in the data (Figure 25). The M-estimators try to reduce the effect of outliers by replacing the squared residual f_i^2 by another function of the residual. The M-estimate of m is defined as

$$\hat{m} = \arg\min_m \sum_{p_i \in X} \rho(f_i, \sigma_i)$$

where $\rho(u)$ is a robust error function and σ_i is the scale parameter associated with f_i, which may or may not be present [11].

The choice of different functions ρ results in different robust estimators. The robustness of a particular estimator refers to its insensitivity to outliers. A tool to analyze the robustness of the function ρ is the associated influence function, ψ [86]. The influence function, characterizes the bias that a particular error measurement has on the solution and is defined as the derivative of the function ρ. We next gather some functions ρ used in computer vision:

- **Quadratic error** $(L^2 \text{ norm})$: $\rho(s) = s^2$, with $\psi(s) = 2s$

- **Absolute error** $\left(L^1 \text{ norm}\right)$:

  $\rho(s) = |s|$, with $\psi(s) = sign(s)$

- **German and McClure** $\left(L^1 \text{ norm}\right)$:

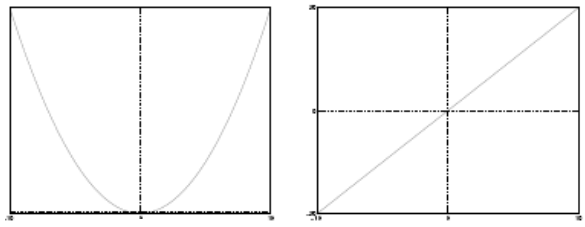
  $$\rho(s) = \frac{s^2}{\sigma + s^2}, \text{ with } \psi(s) = \frac{2s\sigma}{\left(\sigma + s^2\right)^2}$$

- **Cauchy function:**

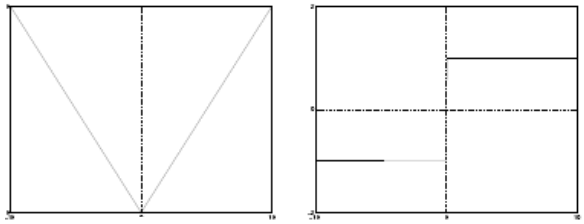
  $$\rho(s) = \log\left(1 + \frac{1}{2}\left(\frac{s}{\sigma}\right)^2\right), \text{ with } \psi(s) = \frac{2s}{2\sigma + s^2}$$

See Figure 25 for illustrations of these functions ρ and the associated *influence* functions, ψ. As can be seen the quadratic and the absolute errors are convex functions, this property is very interesting for the function minimization and is not fulfilled by the other error estimators. However, the quadratic errors are not robust, because their influence function is not bounded. In Figure 25 (b) one can observe that the outlying points are less weighted. However, the absolute errors are not stable, since the function $|s|$ is not differentiable in $s = 0$. The Cauchy and German and McClure do not guarantee a unique solution. For these two functions the influence of large errors decreases linearly with their size as can be seen in Figure 25 (c) and (d). Concave functions have been used as robust functions in [64]. For more details about robust parameters estimation in computer vision see [86], [4] and [11].

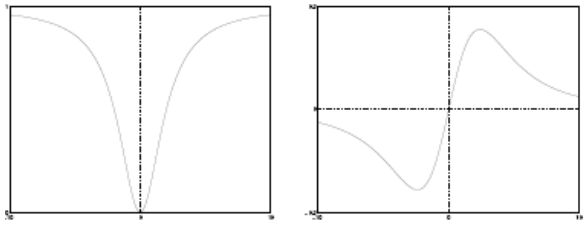
Black and Anandan [4] present a framework based on robust estimation that addresses the problem to improve accuracy and robustness of flow estimates in regions containing multiple motions by relaxing the single motion assumption. They estimate the dominant motion (that is, the apparent camera motion) accurately ignoring the other existing motions. Additionally the approach detects where the single motion assumption is violated (i.e.,where the error measure is large). Then, these positions are examined to see if they correspond to a consistent motion.



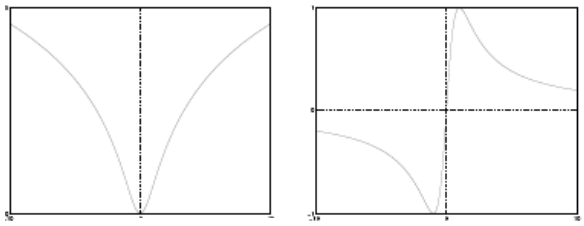
a) Quadratic error



b) Absolut error



c) German & McLure



d) Cauchy

Figure 25: Common robust ρ-functions and ψ functions

Other motion estimation approaches

The classical brightness constancy assumption is generally violated in image sequences taken from the real world. Global or local changes in illumination due to, for instance, a moving camera or a change in the shade of an object may change the appearance of a region. These kind of situations may prevent the correct motion to be

estimated. Alternatives to the optical flow constraint have been already proposed in the literature [50], [66], [77] and [56].

A common approach to handle non constant intensity is through explicit modeling of the illumination change in the OFE [12]. This approach requires complex minimization since, in addition to the motion field, illumination fields must also be estimated. A parametric affine motion model is chosen. The defined functional is linearized, and robust estimation is used. Furthermore the scheme is embedded in a multiresolution scheme.

In [10] a constraint based on spatial gradient's constancy is proposed. This constraint can be written as $\frac{d}{dt}\nabla I(t,x,y)=0$. It relaxes the classical assumption, but requires that the amount of dilation and rotation in the image be negligible (this limitation is often satisfied in practice according to [53]). The resulting technique has been demonstrated to be very robust in the presence of time-varying illumination.

More recently, it has been shown empirically in [9] that the direction of the intensity gradient is invariant to global light changes. In particular, the authors of [9] propose a probabilistic approach in which they analytically determine a probability distribution for the image gradient as a function of the surface's geometry and reflectance. Their distribution reveals that the direction of the image gradient is also relative insensitive to changes in illumination direction. They verify this empirically by constructing a distribution for the image gradient from more than 20 million samples of gradients in a database of 1280 images of 20 inanimate objects taken under varying lighting conditions. The work presented in [20] is based on the latter properties. In particular, a sort of optical flow constraint equation based on probability distributions of gradient directions is proposed. The problems in computing the gradient directions in homogeneous regions (where they are not defined) and at proximity of straight edges (where they do not vary) are avoided by using the stochastic approach. Furthermore, the probability density function is made dependent on the gradient magnitude, becoming sharper on and at proximity of edges, and flatter in homogeneous regions.

An interesting requirement for motion estimation approaches is to be *contrast invariant*. We shall say that an operation T on an image I is contrast invariant if

$$T\left(g\left(I\right)\right)=T\left(I\right)$$

for any no decreasing contrast change g [23].

In [33] a contrast invariant approach for morphological image registration is presented. The proposed functional is based in measuring the errors between the unit normals in two images, and is presented together with suitable regularizations. This approach is contrast invariant, since it is based in unit normals, which are contrast invariant image elements. In [32] alignment of unit normals and other geometric features like curvature have been used for registration of brain images. Other contrast invariant functional have been proposed based on mutual information [58] and [8]. This kind of functionals are widely used in medical image registration. In particular, [58] deals with the image registration problem and analyzes the effects of interpolation methods and resampling in the registration results. Another contrast invariant functional, based on Bayesian inference, was proposed in [31], for piecewise parametric motion segmentation. The authors interpret geometrically the optical flow constraint, then derive a model for the conditional probability of the spatio-temporal image gradient (given a particular velocity vector), and propose a priori assumption on the estimated motion field favoring motion boundaries of minimal length. In that way, their energy functional is an extension of the Mumford- Shah functional [72] from the case of gray value segmentation to the case of motion segmentation.

The proposed functional is

$$E=\sum_{r\in R}\int_{r}\frac{\left(I_{t}+u\partial_{x}I+v\partial_{y}I\right)^{2}}{\left\|w\right\|^{2}\left\|\left(\partial_{t}I,\partial_{x}I,\partial_{y}I\right)\right\|^{2}}dx\ dy+\lambda L\left(C\right)$$

where w is the velocity of region $r\in R$, R a given image partition, $L\left(C\right)$ is the length of the boundary C separating regions, and λ is a weight of the second constraint. Observe that the quotient in the first term is related with the angle between the vectors w and $\left(\partial_{t}I,\partial_{x}I,\partial_{y}I\right)$.

Chapter 4

Video segmentation

In this paper we present a video segmentation procedure obtained minimizing a modified version of the simplified Mumford - Shah functional used for image partition. This procedure uses a graph with spatial and temporal connections to model a video sequence. The temporal connections are defined pre-computing the dense optical flow using methods available in the literature. To simplify the functional minimization we construct the hierarchy of partitions that allows to obtain a very quickly computation.

4.1 Introduction

Video segmentation problem has attracted the attentions of many researchers in the computer vision field because it plays a very important role in many applications, such as video compression, tracking and motion detection. It refers to partitioning video into spatial - temporal regions that correspond to independently moving objects.

Although video segmentation has been studied for several decades, it still remains a difficult problem to solve and various methods for segmentation of images into coherent moving regions have been proposed.

Methods which utilize spatio-temporal image intensity and gradient information have been chosen by some authors [78]. In [78] a 3D segmentation based on luminance information is performed by morphological operators. The scene is segmented according to a criterion of uniform luminance (instead of coherence motion). Another common approach, presented among others in [1], is a two-

step procedure which consists in estimating first the optical flow field between two frames and then segmenting the image based on the estimated optical flow field. In particular, the authors assume that an image is modeled by a set of overlapping layers. They compute initial motion estimates using a least-squares approach within image patches. Then, they use K-means clustering to group motion estimates into regions of consistent affine motion. The accuracy of segmentation results using this approach depends on the accuracy of the estimated optical flow field. Moving object boundaries have usually inaccurate optical flow due to occlusion and use of smoothness constraints. A solution for that issue is proposed in [2] where the authors propose a two-step iteration method similar to the previous ones, but incorporating several changes such as a region-based label assignment approach which favors the obtaining of a spatially continuous segmentation that is closely related to actual object boundaries.

Furthermore, some authors have proposed that optical flow estimation and segmentation should be carried out simultaneously to obtain better results. The algorithm presented in [34] assures very precise motion boundaries by exploiting the static segmentation. In particular it avoids problems related to occlusion and uncovered background.

The authors of [13] treated the analysis of the dynamic content of a scene from an image sequence. They propose an elaborated algorithm which performs a motion-based segmentation using 2D affine models, and apply a statistical regularization approach without the explicit estimation of optic flow fields. Furthermore, they build a temporal link between partitions of successive frames of the sequence. Finally, the interpretation process is carried out in different ways depending of the application.

The authors of [31] propose to segment the image plane into a set of regions of parametric motion on the basis of two consecutive frames. As in [13] they exploit the Bayesian framework to derive a different cost functional which depends on parametric motion models for each of the set of regions and on the boundary separating these regions. As differences we can outline that this formulation is continuous and uses a contour representation of motion discontinuity set (spline or level set based).Furthermore, the data term is based on a different (normalized) likelihood.

The authors of [42] present a method for approximating optical flow by scaled piecewise regular vector field. The error functional balances two terms, an evaluation of the uniformity of pixel motion and a segmentation complexity term. The method is a variational approach similar to the region merging minimization procedure described by Morel and Solumini [70].

4.2 Proposed approach for video segmetation

A video sequence can be modeled as a function $f:[T_i,T_f]\times\Omega\to R$ with spatial domain Ω and time interval $[T_i,T_f]$. We assume that the time is discrete $\{t_n\}_{n\in[1,N]}$. Our purpose is to compute the segmentation of a video sequence defined by a pair $(C,\tilde{f})$ such that:

- $\tilde{f}:([T_i,T_f]\ x\ \Omega)-C\to R$ is a regular function in $([T_i,T_f]\ x\ \Omega)-C$ domain.
- C is the set of boundaries where $\tilde{f}$ is discontinuous.

Since a video is a sequence of images, the boundary C is given by $\bigcup_{n=1}^{N} C_n$ where C_n is the set of boundaries each of one is related to the frame observed at time t_n.

To solve this problem we propose a video segmentation procedure based on two different steps. In the first step we construct a new data structure to handle a video pre-computing the optical flow. In the second step we define a video segmentation minimizing a modified version of the Mumford-Shah functional defined in Section 2.1 for image partition.

4.2.1 Data structure for video handling: graph

In Section 1 we have seen that under certain topological conditions a single image can be modeled by a tree structure. In this case is possible to define a spatial connection between different pixels (or regions) of the same image.

In Video analysis, instead, besides the usual spatial connectivity of pixels on each single frame, we have a natural notion of "temporal" connectivity between pixels on consecutive frames given by the optical flow. In this case, it makes sense to extend the tree data structure used to model a single image with a graph data structure that allows to handle a video sequence.

Given a video sequence we can build the appropriate graph in the following way. First, we pre-compute a dense optical flow of the whole video sequence using any of the methods available in literature (we tried some of them with similar final results [16], [15], [24] and [19]). This flow assigns a vector on every pixel of each frame but the last. Now, the vertices of the graph are defined as all the pixels of the video, each one assigned its corresponding gray level. The edges of the graph are of two kinds: spatial edges and temporal edges. Spatial edges join each pixel with its 8-neighbors on the same frame. Temporal edges are defined using the pre-computed optical flow: If the flow vector on pixel (x, y, t) is (u, v), then we add to the graph an edge joining pixel (x, y, t) with pixel $(x+[u],\ y+[v],\ t+1)$, where the square brackets denote the nearest integer. Pruning and simplifying the branches of this graph corresponds to applying spatial-temporal coherent morphological operators to the video sequence. A selection of regions of this graph can be regarded as a segmentation of the video. For example, by selecting very few regions we obtain a very coarse segmentation of the video. We call the segments of this segmentation the "*tubes*" of the video. The tubes encodes temporally coherent segmentations of all the objects on the video, which can be used for tracking. This structure is useful to write higher level algorithms on the video.

The intersections of tubes with frames are called "*regions*". Thus, the regions of a given frame are segmentation of that frame. So, to handle a video sequence, we must consider the following object that characterize it:

- pixels;
- frames;
- tubes;
- regioins.

The list of pixels and the list of frames are trivially related, because each frame contains the same number of pixels. The relationships among the other lists are the true interest of the data structure, since they hint a combinatorial representation for the video objects. See Figure 26 for a diagram illustrating the relative inclusions between these structures.

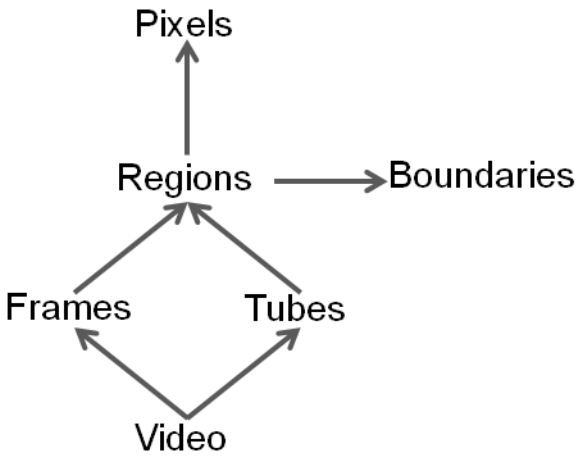


Figure 26: Inclusion relationships between the parts of the data structure. A video is divided into frames, and into tubes. The intersection of a tube with a frame is a region. Each region is a set of pixels. Neighboring regions on the same frame are separated by their common boundaries.

4.2.1.1 Simple computation using the tubes

The mere act of storing a video sequence using “the tubes” lends itself to certain higher level algorithms, which provide raw analysis of the objects that appear in the video. Here we list four of these algorithms. The algorithm for relative depth from motion will be an example of a more complex one.

Tube statistics. The simplest thing that we can do with the tubes is to compute statistics of all the regions. For each tube, we can see how its area evolves along frames, its mean motion (to select immediately the fastest moving objects in the video), the evolution of the length of its boundary, etc.

Tube topology. The tubes can be classified by their topology, looking how it evolves in time. The simplest case is that of a tube which intersects each frame (from a certain interval of frames) in a single connected region. A different case is that of two objects that merge or split as time passes, for example when one object occludes another one of the same color. In that case, the tube has the shape of the letter Y, with the junction appearing at the frame where the objects merge or split. By single traversal of the data structure, we can build a list of the branched and unbranched tubes, and of the regions that they span long the video.

Optical flow regularization. We can use the structure of tubes and regions to improve a given dense optical flow. If we suspect, as often happens, that the optical flow is wrong or imprecise near the boundaries of the objects, we can discard those samples of the dense flow, and extrapolate their values from the inner parts of the region. This is a regularization of the optical flow in a single frame. But we can also smooth the flow along several frames, to enhance its temporal consistency. The connectivity of the regions assures that we will not be mixing flow samples from different layers of movement.

Flow from segmentation. As an extreme case of the previous computation, we can construct an optical flow from scratch, just by looking at evolution of the tubes in time. If we find the best match from each region into the next one, we already have a model of the movement of that region. By sampling that motion on the pixels of the region, we produce a dense optical flow. While it is very crude, this method does not depend on the resolution of the video, only on the structure of the tubes. Thus, it can be used as a starting point for more precise algorithms. The quality of the results depends on the criterion for registering pairs of consecutive regions. A naive criterion that minimizes Hausdorff distance (or that matches the center of mass) will produce incorrect results for occluded objects, but perfect results for

objects which are on top of all the others, and move in a plane perpendicular to the line of view. In some circumstances, this may be useful.

4.2.2 Modified version of a simplified Mumford-Shah functional for video segmentation

As we have explained in Section 2 Simplified Mumford-Shah functional is largely used in image partition operation. Our purpose is to extend it to use a new version of the simplified Mumford-Shah functional for video segmentation.

The Mumford-Shah functional (2.1) used to partition a single image could be incomplete to obtain a good video segmentation because it takes into account only the color of the regions but there are not information about the regions "movement". We have modified it introducing an additional term to solve the video segmentation problem obtaining the following functional:

$$J_{MS}^{\lambda}(C,\tilde{f}\,|_{\Omega x[T_i,T_f]\setminus C}) = \lambda H^1(C) + \int_{\Omega x[T_i,T_f]\setminus C}(f-\tilde{f})^2 + \int_{\Omega x[T_i,T_f]\setminus C}((u,v)-(\tilde{u},\tilde{v}))^2 \qquad (4.1)$$

where (u,v) is the optical flow vector of pixels of the original video sequence and $(\tilde{u},\tilde{v})$ is the optical flow vector of pixels of the approximate video sequence. The second integral allows to consider the movement of the region using the optical flow pre-compute in the first step. The idea is to merge neighboring region with similar color and similar movement.

Observe that the minimization of (4.1) has an exponential complexity on the size of graph that models the video, if all possible combinations of its nodes are taken into account. This computation becomes feasible if we restrict our search space to a hierarchy of a partitions of a video domain $[T_i,T_f]\ x\ \Omega$ as we have seen in section 2 for image partition operation.

4.2.3 Minimization of the modified version of M-S functional using a hierarchy of partition

To minimize (4.1) we must construct the hierarchy of a partitions. The construction follows a bottom-up procedure. Indeed the leaves of the hierarchy are the nodes of the graph that models the video. We construct next levels merging nodes of the reduced level without father. We iterate this procedure until we reach the set $[T_i, T_f]\ x\ \Omega$. To choose the nodes to merge at each iteration we perform the following steps.

Let $I(R) = \int_R (f - \tilde{f})^2 + \int_R ((u,v) - (\tilde{u},\tilde{v}))^2$

so we can rewrite (4.1) as:

$J^{\lambda}_{MS}(C, \tilde{f}\,|_{\Omega x [T_i,T_f] \setminus C}) = \lambda H^1(C) + I(\Omega x [T_i, T_f] \setminus C).$

Suppose to merge each region of the segmentation with its neighbor for all neighbors. We obtain a new set of border $\hat{C}$, a new function $\hat{f}$, a new functional $\hat{J}$ and, for the merging region, a new vector optical flow $(\hat{u}, \hat{v})$. Follows that

$$\Delta J = J - \hat{J} = \lambda \Delta H + \Delta I \qquad (4.2)$$

where

$\Delta H = H^1(C) - H^1(\hat{C})$

and

$\Delta I = I(\Omega x [T_i, T_f] \setminus C) - I(\Omega x [T_i, T_f] \setminus \hat{C}).$

Setting $\Delta J = 0$ we calculate λ by (4.2) as:

$\lambda = -\frac{\Delta I}{\Delta H}$.

Observe that ΔH is a positive number because the length of the set of curve of the regions that compose the single image decrease when we merge together two or more adjacent regions. While ΔI is a negative number because we loss energy when we merge together one or more adjacent regions.

We repeat this procedure for each region. At the end we merge the region and its neighbor that are characterized by the minimum λ

value which has been calculated. In this case, for each level of the hierarchy, we define a local optimal solution of (4.1) for the considered lambda value, in the sense that any other merging of regions of the segmentation leads to an increase of the functional (4.1).

For each node of the hierarchy we define an interval $\left[\lambda^{+},\lambda^{-}\right[$ where λ^{+} indicate the lambda value in which node appears in the hierarchy and λ^{-} the λ^{+} of its father.

Returning at (4.1), to minimize it we fix a λ value and define a cut of hierarchy selecting the nodes such that $\left[\lambda^{+} \le \lambda < \lambda^{-}\right[$.

4.3 Experimental results

In this section we present some experimental results obtained using the video segmentation procedure described above.
In the first example we consider the video sequence composed by five frames showed in Figure 27.

Figure 27: foreman video sequence on frame 1,2,3,4 and 5 from left to right and top to bottom.

In this video sequence foreman performs a rotation of the head and close the eyes.

In Figure 28 is possible to observe the optical flow pre-computed between frame 1 and frame 2:

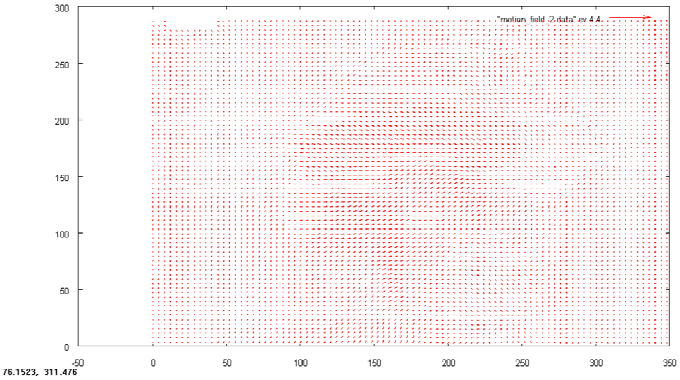


Figure 28: optical flow between frame 1 and frame 2 of the foreman video sequence.

Segmentation result has showed Figure 29.

Figure 29: foreman segmentation sequence

Observing the segmentation of the frames that compose the video sequence it is possible to follow the movements of the foreman.

Consider now the “Hamburg taxi” video sequence in Figure 30.

Figure 30: Hamburg taxi video sequence on frame 1,2,3 and 4 from left to right and top to bottom.

This sequence is more complex than foreman video because some objects that move in the scene (car at left and right side of each frame) have a similar color of a static background (street).

If we try to segment this video using a graph to model the video and the original Mumford-Shah functional used for image partition (2.1) we obtain a result showed in Figure 31.

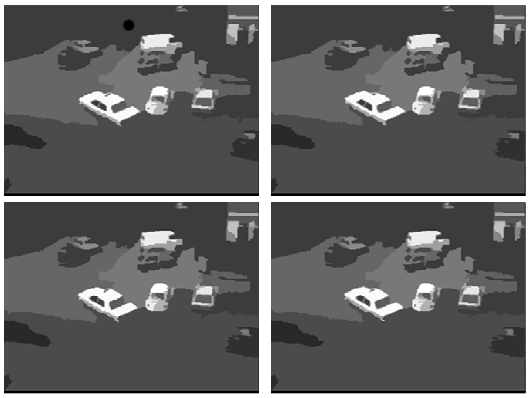


Figure 31: Hamburg taxi video segmentation using the original Mumford Shah functional.

In this case we can observe a good white taxi segmentation vice versa we lose a lot of features in the segmentation of the car on the left and right side of the frames. The reason is that the original functional merge neighboring regions with similar color without considering their movement. So in this case is “simple” to consider the white taxi because it has a color different from the one of the street. Car at left and right side of the frames have a color similar to the background so in this case is difficult to “separate” them by the street.

Figure 32 shows the segmentation result obtain minimizing (4.1):

Figure 32: Hamburg taxi video segmentation using the modified Mumford Shah functional.

In this case we can observe more features about the car that move along the scene because the chosen functional allow to merge region with the same color and the same optical flow vector. So the car has the same color of the street but different optical flow vector.
In particular the latter related pixels that reproduce the street are zero (street is the static background of the scene) and the optical flow vector about the pixel that reproduce the cars are different by zero (cars are the dynamic foreground). This is confirmed by the following the optical flow between frame one and frame two showed in Figure 33.

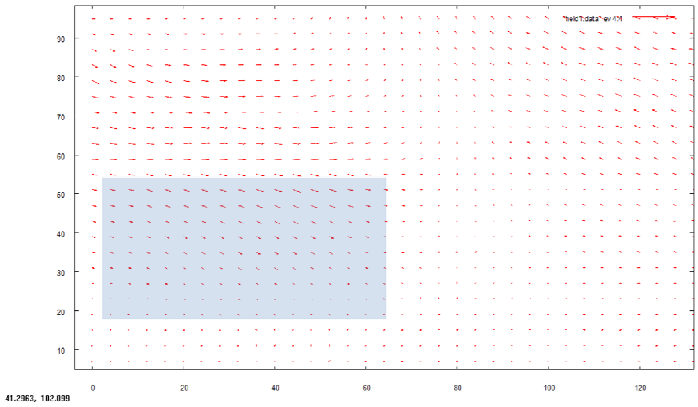


Figure 33: optical flow between frame one and two of the Hamburg taxi video sequence.

The light blue rectangle shows the boundary between the street and the car at the left side of the frame. In this case we have neighboring regions with the same color but different optical flow so we do not merge this regions.

These examples show some video segmentation that is possible to obtain using a video segmentation procedure based on the minimization of a modified version of the Mumford - Shah functional. This procedure use a graph to handle a video sequence. This graph consider a spatial connection between pixels of the same frame and temporal connection between pixels of consecutive frames using the optical flow vector. The minimization of Mumford-Shah functional can be very complex if we consider each possible combination of the graph nodes. This computation becomes easy to do if we take into account a hierarchy of a partitions constructed starting by the nodes of the graph. As we have showed this procedure allows to obtain a good segmentation also if we consider video sequence in which the dynamic foreground has got a similar color to the static background.

References

[1] E. H. Adelson and J.Y.A. Wang. “Representing moving images with layers”. IEEE Trans. on Image Processing, 3(5):625–638, September 1994.

[2] Y. Altunbasak, P. Erhan Eren, and A. Murat Tekalp. “Region-based parametric motion segmentation using color information”. Graphical models and image processing, 60(1):13–23, January 1998.

[3] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. “Axioms and fundamental equations of image processing: Multiscale analysis and P.D.E.”. Archive for Rational Mechanics and Analysis, 16(9):200–257, 1993.

[4] P. Anandan and M.J. Black. “The robust estimation of multiple motions: parametric andpiecewise-smooth flow fields”. Computer Vision and Image Understanding, 63(1):75–104, January 1996.

[5] L. Alvarez, J. Sánchez and J. Weickert. “Reliable estimation of dense optical flow fields with large displacements”. International Journal of Computer Vision, 39(1):41–56, 2000.

[6] C. Ballester, V. Caselles, L. Garrido and L. I. Munoz. Level Lines Selection with Variational Models for Segmentation and Encoding. Journal of Mathematical Imaging and Vision, Vol. 27, 2007, pp. 5-27.

[7] C. Ballester, V. Caselles, and P. Monasse. “The Tree of Shapes of an image”. ESAIM: Control, Optimization and Calculus of Variations, 9:1–18, 2003.

[8] I Bardera, A. Boada, J. Rigau and M. Sbert. “Medical image segmentation based on mutual information maximization”. In Proceedings of 7th International Conference on Medical Image Computing and Computed Assisted Intervention (MICCAI 2004), 2004.

[9] P. Belhumeur, H. Chen, and D. Jacobs. In search of illumination invariants. In International Conference on Computer Vision and Pattern Recognition, pages 254–261, 2000.

[10] M. Bertero, T.A. Poggio, and V. Torre. Ill-posed problems in early vision. Procedings of the IEEE, 76(8):869–889, August 1988.

[11] M. J. Black. Robust incremental optical flow. PhD thesis, Yale University, Computer Science Dept., sept 1992.
[12] P. Bouthemy and J.M. Odobez. Robust multiresolution estimation of parametric motion models applied to complex scenes. Journal of Visual Communication and Image Representation, 6(4):348–365, December 1995.
[13] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. International Journal of Computer Vision, 10(2):157–182, 1993.
[14] T. Brox , A. Bruhn, S. Didas, N. Papenberg and J. Weickert. High accuracy optical flow computation with theoretical justified warping. International Journal of Computer Vision, to appear, 2005.
[15] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In Computer Vision and Pattern Recognition, pages 41–48. IEEE, 2009.
[16] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. Computer Vision-ECCV 2004, pages 25–36, 2004.
[17] A. Bruhn, C. Feddern, T. Kohlberger, C. Schnoerr and J. Weickert,. “Real-time optic flow computation with variational methods”. In International Conference on Computer Analysis of Images and Patterns, pages 222–229. Springer Verlag, August 2003.
[18] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. International Journal of Computer Vision, 61(3):211–231, 2005.
[19] A. Bugeau and N. Papadakis. “Tracking with occlusions via graph cuts”. IEEE Transactions on Pattern Analysis and Machine intelligence, 2010.
[20] P.Y. Burgi. Motion estimation based on the direction of intensity gradient. Image and Vision Computing, 22(8):637–653, August 2004.
[21] B.M. ter Haar Romeny, editor. Geometry-Driven Diffusion in Computer Vision. Kluwer Academic Publishers, 1994.
[22] J. Canny. A variational approach to edge detection. In National Conference on Artificial Intelligence, pages 54–58, Washington DC, August 1983.

[23] V. Caselles, B. Coll, and J.M. Morel. “Topographic maps and local contrast changesin natural images”. International Journal of Computer Vision, 33(1):5–27, September 1999.
[24] V. Caselles, L. Garrido, M. Kalmoun. “Multilevel optimization as computational methods for dense optical flow”, submitted to SIAM, 2010.
[25] V. Caselles, J.L. Lisani, J.M. Morel, and G. Sapiro. “Shape preserving local contrast enhancement”. In Proceedings of International Conference of Image Processing, pages I:314–xx, 1997.
[26] V. Caselles, J.L. Lisani, J.M. Morel, and G. Sapiro. Shape preserving local histogram modification. IEEE Transactions on Image Processing, 8(2):220, February 1999.
[27] R. Chiariglioni. “Mpeg and multimedia communitations”. IEEE Transactions on Circuits and System for Video Technology, 7:5-18,1997.
[28] P. A. Chou, T. Lookabaugh, and R. M. Gray. “Optimal pruning with applications to treestructured source coding and modeling”. IEEE Trans. Inform. Theory, 35:299–315, 1989.
[29] G. Côté, B. Erol, M. Gallant, and F. Kossentini. H.263+: Video coding at low bit rates. IEEE Transactions on circuits and systems for video technology, 8(7), 1998.
[30] J.L. Cox and D.B. Karron. Digital Morse theory. Manuscript available from http://www.casi.net, 1998.
[31] D. Cremers and S. Soatto. “Motion competition: a variational approach to piecewise parametric motion segmentation”. International Journal of Computer Vision, 62(3):249 – 265, 2005.
[32] C. Davatzikos. “Spatial transformation and registration of brain images using elastically deformable models”. Computer Vision and Image Understanding, 66(2):207–222, May 1997.
[33] M. Droske and M. Rumpf. “A variational approach to non-rigid morphological image registration”. SIAM Journal Applied Mathematics, 64(2):668–687, 2004.
[34] F. Dufaux, F. Moscheni, and A. Lippman. “Spatio-temporal segmentation based on motion and static segmentation”. In IEEE Proc. ICIP’95, volume 1, pages 306–309, October 1995.

[35] J.L. Dugelay and H. Sanson. Differential methods for the identification of 2D and 3D motionmodels in image sequences. Image Communication, 7:105–127, September 1995.
[36] W. Enkelmann. “Investigations of multigrid algorithms for the estimation of optical flow fieldsin image sequences”. Comput. Vision Graph. Image Process., 43(2):150–177, 1988.
[37] C. Fiorio. “A topologically consistent representation for image analysis: the frontiers topological graph”. In Proceedings of the 6th Conference of Discrete Geometry for Computationial Imagery, Lyon, France, 1996.
[38] C. Fiorio. Border map: A topological representation for nd image analysis. In Proceedings of Conference of Discrete Geometry for Computationial Imagery, pages 242–257, Marne la Vall´ee, France, March 1999.
[39] C. Fiorio. “Topological operators on the frontiers topological graph”. In Proceedings of Conference of Discrete Geometry for Computationial Imagery, pages 207–217, Marne la Vall´ee, France, March 1999.
[40] L. Garrido and P. Salembier, “Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval”. IEEE Transactions on Image Processing 9(4), pp. 561–576, 2000.
[41] M. Gangnet, J.C. Herv´e, T. Pudet, and J.M. Van Thong. Incremental computation of planar maps. Digital Pattern Recognition Letters, (5), 1989.
[42] F. Guichard and L. Rudin. Velocity estimation from images sequence and application to superresolution. In IEEE Proc. ICIP’99, volume 3, pages 527–531, October 1999.
[43] L. Guigues, “Mod`eles Multi-´Echelles pour la Segmentation d’Images”. PhD thesis, Universit´e de Cergy-Pontoise, 2003.
[44] W. Hackbusch. Multi-grid Methods and Applications. Springer-Verlag: Berlin, 1985.
[45] B. Horn and B Schunck. Determining optical flow. Artificial Intellience, 17:185–203, 1981.
[46] P. J. Huber. Robust statistics. John Wiley, New York, 1981.
[47] R.A. Hummel. “Representations based on zero-crossings in scale-space”. In Proceedings of the IEEE International Conference of Computer Vision and Pattern Recognition, pages 204–209, 1986.

[48] K. Kanade and B. Lucas. An iterative image registration technique with an application to stereo vision. Procedings Seventh International Joint Conference on Artificial Intelligence, pages 674–679, august 1981.
[49] K. Kanatani. Group-Theoretical Methods in Image Understanding. Springer-Verlag, 1990.
[50] A.C. Kak, Y. Kim and A. M. Martinez. “Robust motion estimation under varying illumination”. Image and Vision Computing, 23(4):365–375, 2005.
[51] J.J. Koenderink. “The structure of images. Biological Cybernetics”, 50:363–370, 1984.
[52] G. Koepfler, C. Lopez and J.M. Morel. A multiscale algorithm for image segmentation by variational method. SIAM J. Numer. Anal, 31:282–299, 1994.
[53] J. Konrad and C. Stiller. Estimating motion in image sequences. IEEE Signal Processing Magazine, 16(4):70–91, July 1999.
[54] V.A. Kovalevsky. Finite topology as applied to image analysis. Computer Vision, Graphics and Image Processing, 46(2):141–161, May 1989.
[55] A.S. Kronrod. On functions of two variables. Uspehi Mathematical Sciences, 5(35), 1950. (in Russian).
[56] S.-H. Lai. “Computation of optical flow under non-uniform brightness variations”. Pattern Recognition Letters, 25(8):885–892, February 2004.
[57] P. Lienhardt. “Topological methods for boundary representation: A survey”. Computer Aided Design, 23(1):59–81, 1989.
[58] J.B.A. Maintz, J.P.W. Pluim and M.A. Viergever. Mutual-information-based registration of medical images. IEEE Trans. on Medical Imaging, 22(8):986–998, August 2003.
[59] S. Mallat. A Wavelet Tour of Signal Processing. Academic Press, New York, 1998.
[60] D. Marr. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. W.H. Freeman and Co., 1982.
[61] D. Marr and E.C. Hildreth. “Theory of edge detection”. Proceedings of the Royal Society of London, B-207:187–217, 1980.

[62] G. Matheron. Random Sets and Integral Geometry. John Wiley, N.Y., 1975.
[63] E. Mémin and P. Pérez. A multigrid approach for hierarchical motion estimation. In Processing Sixth International Conference on Computer Vision, pages 933–938, January 1998.
[64] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. International Journal of Computer Vision, 46(2):129–155, February 2002.
[65] Y. Meyer. Wavelets: Algorithms and Applications. SIAM, Philadelphia, 1993.
[66] H. Miike, T. Sakurai and L. Zhang. “Detection of motion fields under spatio-temporal nonuniform illumination”. Image Vision Comput., 17(3-4):309–320, 1999.
[67] J. Milnor. Morse Theory. Number Study 51 in Annals of Mathematics Studies. Princeton University Press, 1969.
[68] P. Monasse. “Morphological representation of Digital Images and Application to Registration”. PhD thesis, Université Paris IX-Dauphine, June 30, 2000.
[69] P. Monasse and G. Guichard. “Fast computation of a contrast invariant image representation”. IEEE Transactions on Image Processing, 9:860–872, 2000.
[70] J.M. Morel and S. Solimini. Variational Methods in Image Processing. Birkhauser, 1994.
[71] D. Mumford and J. Shah. “Optimal approximations by piecewise smooth functions and variational problems”. Communications on Pure and Applied Mathematics, XLII(5):577–685, 1988.
[72] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure Applied Math, 42:577–685, 1989.
[73] L. I. Munoz, “Image Segmentation and Compression using The Tree of Shapes of an Image. Motion Estimation”. PhD thesis, Universitat Pompeu Fabra, 2005.
[74] H.-H. Nagel andW. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:565–593, 1986.
[75] M.H.A. Newman. “Elements of the Topology of Plane Sets of Points”. Dover, 1992.

[76] M. Nitzberg and D. Mumford. “The 2.1-D sketch”. In Proceedings of the 3d International Conference on Computer Vision, pages 138–144, Osaka, Japan, 1990.
[77] A. Nomura. Spatio-temporal optimization method for determining motion vector fields under non-stationary illumination. Image Vision Comput., 18(12):939–950, 2000.
[78] M. Pardas and P. Salembier. “3d morphological segmentation and motion estimation for images sequences”. Signal Processing, 38(2):31–41, September 1994.
[79] A. Rosenfeld. Adjacency in digital pictures. Information and Control, 26, 1974.
[80] J. Serra. “Image Analysis and Mathematical Morphology”. Academic Press, New York, 1982.
[81] J. Serra. “Introduction to mathematical morphology”. Computer Vision, Graphics and Image Processing, 35(3):283–305, September 1986.
[82] C. Schnörr. “Determining optical flow for irregular domains by minimizing quadratic functional of a certain class”. International Journal of Computer Vision, 6(1):25–38, 1991.
[83] C. Schnörr and J. Weickert. “Variational image motion computation: Theoretical framework, problems and perspectives”. In DAGM-Symposium, pages 476–488, 2000.
[84] C. Schnörr and J.Weickert. “Variational optic flow computation with a spatio-temporal smoothness constraint”. Journal of mathematical imaging and vision, 14(3):245–255, May 2001.
[85] T. Sikora. “The mpeg-7 visual standard for content description - an overview”. IEEE Transactions on Circuits and Systems for Video Technology, 11:696–702, 2001.
[86] C. H. Stewart. “Robust parameter estimation in computer vision”. SIAM Rev., 41(3):513–537,1999.
[87] A.M. Tekalp. “Digital Video Processing”. Prentice-Hall, 1995.
[88] J. Weickert. “On discontinuity-preserving optic flow”. In Computer Vision and Mobile Robotics Workshop, pages 115–122, Santorini, 1998. Springer Verlag.
[89] M. Wertheimer. Untersuchungen zur Lehre der Gestalt, ii. Psychologische Forschung, (4):301–350, 1923.

[90] A.P. Witkin. “Scale-space filtering”. In International Joint Conference on Artificial Intelligence, pages 1019–1022, Karlsruhe, 1983.