A statistical framework in variational methods of image and video processing problems with high dimensions

Sylvain Boltz

Université de Nice - Sophia Antipolis

École doctorale "Sciences et Technologies de l'Information et de la Communication"

THÈSE

pour obtenir le titre de

Docteur en Sciences

Mention : Automatique, Traitement du Signal et des Images

présentée par

Sylvain Boltz

Équipe d'accueil : CReATIVe - Laboratoire I3S

Un cadre statistique en traitement d'images et vidéos par approche variationnelle avec modélisation haute dimension

Thèse dirigée par Pr. Michel BARLAUD et par Dr. Éric DEBREUVE

soutenue le 9 juillet 2008 devant le jury composé de

Vicent CASELLES	Professeur à l'Université de Pompeu Fabra, (Barcelone)	Rapporteur
Nikos PARAGIOS	Professeur à l'Ecole Centrale (Paris)	Rapporteur
Patrick PÉREZ	Directeur de recherche INRIA (Rennes)	Rapporteur
Éric DEBREUVE	Chargé de recherche CNRS	Examinateur
Pierre COMON	Directeur de recherche CNRS	Examinateur
Frank NIELSEN	Professeur à l'Ecole Polytechnique (Paris)	Examinateur
Michel BARLAUD	Professeur des Universités à Nice-Sophia Antipolis	Directeur de thèse

University of Nice - Sophia Antipolis
Graduate school "Sciences et Technologies de l'Information et de la Communication"

PhD THESIS

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Science

Specialized in Signal and Image Processing

presented by

Sylvain Boltz

prepared at CReATIVe - Laboratoire I3S

A STATISTICAL FRAMEWORK IN VARIATIONAL METHODS OF IMAGE AND VIDEO PROCESSING PROBLEMS WITH HIGH DIMENSIONS

Thesis supervised by Pr. Michel BARLAUD and Dr. Éric DEBREUVE

presented on July, 9th 2008

Vicent CASELLES	Pr. at University Pompeu Fabra, Spain	Reviewer
Nikos PARAGIOS	Pr. at Ecole Centrale Paris, France	Reviewer
Patrick PÉREZ	Research director INRIA Rennes, France	Reviewer
Éric DEBREUVE	Research scientist CNRS	Examiner
Pierre COMON	Research director CNRS	Examiner
Frank NIELSEN	Pr. at Ecole Polytechnique Paris, France	Examiner
Michel BARLAUD	Pr. at University of Nice Sophia-Antipolis	Advisor

Résumé: Cette thèse aborde le traitement d'images et de vidéos sous l'angle variationnel, c'est-à-dire sous forme d'une énergie dont le minimum est atteint pour la solution. La modélisation adoptée pour formaliser le problème et obtenir ces énergies peut être déterministe ou stochastique. Il est connu que la première approche est équivalente à la classe paramétrique de la seconde. Ce constat nous a amenés à faire le choix de la seconde approche a priori plus générale si l'on se débarrasse de l'hypothèse paramétrique. En contrepartie, il s'agit d'tre capable d'exprimer et d'estimer une énergie en fonction des données alors interprétées comme des échantillons d'une variable aléatoire. Ce premier obstacle est classiquement surmonté par l'emploi de méthodes à noyau fixe sur des lois marginales, autrement dit, en supposant les canaux de données indépendants entre eux. Or cet obstacle en cache deux autres : l'inhomogénéité de la répartition des échantillons dans leur espace d'appartenance et leur faible densité dans cet espace. Ces difficultés, ainsi que l'hypothèse d'indépendance mentionnée plus haut, sont d'autant plus pénalisantes que le modèle proposé pour les données est de grande dimension (canaux couleur, mais aussi ajouts d'autres canaux pour prendre en compte les structures locales des images). Au fondement d'estimateurs de mesures statistiques telle que l'entropie, l'idée du kième plus proche voisin permet de résoudre les difficultés évoquées en s'adaptant à la densité locale des données, en considérant les canaux conjointement, et ce quel que soit leur nombre. Dans ce contexte, nous proposons une approche statistique générale inspirée de la théorie de l'information, dédiée aux approches variationnelles car estimant efficacement des énergies en haute dimension, permettant le calcul de leur dérivée et pouvant estimer localement des probabilités. Ce cadre est appliqué aux trois problèmes de traitement d'images ou de vidéos que sont l'estimation de flot optique, le suivi d'objets vidéos et la segmentation. Ce point de vue, en permettant de limiter sinon de s'affranchir du problème de la dimension des données, autorise la définition de nouvelles mesures et lois de probabilités plus adaptées aux images naturelles. Certains travaux en suivi d'objets et en segmentation ont conduit à des implémentations industrielles.

Abstract: This thesis addresses variational formulation of image and video processing problems. This formulation expresses the solution through a minimization of an energy. These energies can be expressed as deterministic or stochastic. The first approach corresponding to the parametric class of the second one. The second class is then more general if we get rid of the parametric assumption. In return, the energy must be expressed as a function of the data considered as random variables. These functions are classically estimated with fixed-sized kernels on marginal distributions of the data, assuming the different channels are independent. These methods have two limitations, the inegal repartition and sparsity of the data in the space. These difficulties, as well as the independence assumption are enhanced when the data of the image are high dimensional (color channels, or other channels describing local patterns of natural images). At the foundation of statistics, the k-th nearest neighbor can solve these difficulties by locally adapting to the repartition of the data and treating the channels jointly. We propose a general statistical framework based on statistics and information theory. This new framework is dedicated to variational problems as it efficiently estimates, high dimensional energies, gradients of these energies and local probabilities. This framework is applied to three problems of image and video processing: optical flow, object tracking and segmentation. This framework circumvents the problem of dimensionality and allows us to introduce new measures and probabilities more adapted to natural images. Some results obtained have been applied in an industrial context.

Contents

Acknowledgments

This thesis was performed in the Creative project at I3S laboratory, under the joint supervision of Eric Debreuve and Michel Barlaud. It was funded by the French department of education and research.

First, I would like to thanks Michel Barlaud and Eric Debreuve for supervising my PhD thesis. I greatly acknowledge them for their wise advices and their involvements in our research.

Next, I would also like to express my deep thanks to Vicent Caselles, Nikos Paragios, Patrick Pérez for having accepted to review my manuscript despite the important amount of work this required. I am also extremely grateful to Pierre Comon, Frank Nielsen for their participation in my jury. I would like to thank all of them for their useful comments.

I would like to thank Stefano Soatto, for accepting me as a postdoc in his team in UCLA.

I would also like to thanks all my lab friends for those great years: Aline, André, Anis, Ariane, Benoît, Cédric, Changy, David, Eric D, Eric W, Francois, Fred, Jérôme, Lao, Laure, Laurent, Mago, Marie, Moger, Muriel, Paul, Ronald, Silvia, Sandrine, Stéphane, Thomas, Vincent, Vincenzo, Wali, Yasmine, and all the forgotten ...

I would like to address additional special thanks to my "real life friends" Changy, Lao, Mago, Thomas, Vincent, Jérôme, V, Rudy, Steven.

Finally, I would like to thank my family : Bernard, Michèle, Nicolas, Lucie, Erwin who supported me for years.

CHAPTER 1

INTRODUCTION

1.1 Context

The general context of this thesis is extraction of information in image and video processing. Multiplications of video cameras in the last years for communication, entertainment (cell phones, web cameras), security (video surveillance) have provided huge amount of data and have raised the need for new challenging applications in the processing of these videos. Applications include, scene understanding, scene compositing, tracking of a target, these operations are part of the field named computer vision. While in some industrial applications it is still reasonable, though very expensive, to manually perform algorithms frame by frame as performed in cinema post-production. In some other industrial applications, for instance processing on large database of historical videos (INA) or of entertainment videos (Youtube), automatic algorithms or semi-automatic algorithms, *i.e* supervised by an operator, to process operations in these videos, are of great interest.

Algorithms of computer vision can be classified in three categories. Low level vision algorithms, mid level vision algorithms, and high level vision algorithms. Low-level vision algorithms process basic operations on image pixels, *e.g:* some pixels are moving in the image plane. Mid-level vision include higher level processing like pixel grouping, *e.g:* some pixels of similar color are moving following a same coherent affine motion. High-level vision is the final stage which gives a semantic meaning to the scene, *e.g:* a vehicle is moving.

In this thesis we focus on low to mid level vision algorithms which means we have poor semantic understanding, poor knowledge or priors about the information to be extracted. The little high-level information required (*e.g.* definition of the object of interest, initial tracking state or segmentation) is defined by an operator.

Low and mid level vision algorithms in video processing treated in this manuscript include optical flow, tracking, and segmentation. Examples in cinema postproduction are shown on Fig. 1.1, first example is scene compositing between two different scenes, second example is a tracking of an head in order to apply color effects on the skin of the actor. More generally, we aim at algorithms which can be formulated as a variational problem: an energy can be defined and minimizing this energy will give a solution to the problem.

Figure 1.1: *Examples of applications: scene compositing (segmentation), color effects (tracking)*

1.2 State of the art

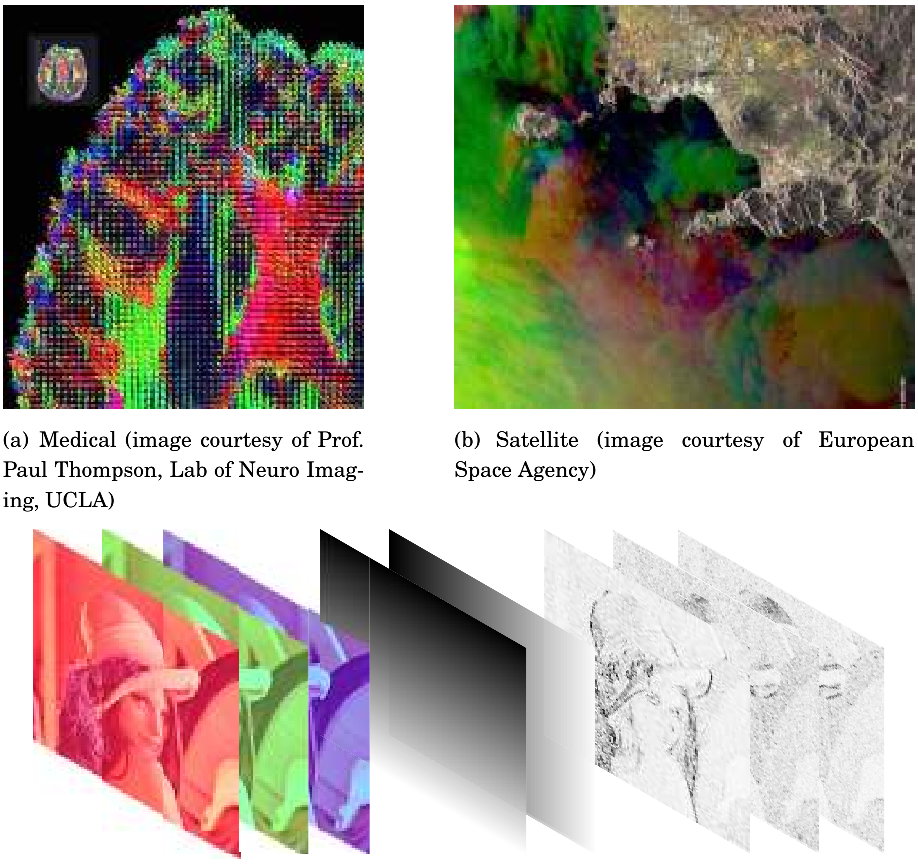
In this thesis the focus is on a large class of variational problems including tracking, motion estimation through optical flow and segmentation. Large class as these methods use different mechanisms in their resolution. Among these methods, some methods can be formulated on the deterministic viewpoint, some other methods can be formulated in the statistical viewpoint. These viewpoints if deterministic usually allow fast derivative-based convergence and if statistic allow flexibility with respect to the model as it accounts for randomness inherently present in the data. These two viewpoints are often tightly related and often end up with the same equations.

For each viewpoint, methods can be classified in two groups. On one hand some methods assume a model for the image and match the model by computing a small set of parameters (*e.g.* parametric PDFs on the statistical viewpoint). On the other hand some methods generate their own model from the image (*e.g.* nonparametric PDFs on the statistical viewpoint). The nonparametric model performs in general better than the parametric model as it adapts to the image statistics, albeit often increasing complexity.

On the statistical viewpoint, nonparametric techniques include Parzen windowing which estimates the probability density function (PDF) efficiently on low dimensional spaces. However, higher dimensional spaces in image processing have to be considered, whether by necessity (neighborhood patches [BCM05], structure tensor [RBD03]) or by nature (high dimensional measures *e.g.* mutual information, or high dimensional acquisition *e.g.* diffusion tensors [AFPA06]). Examples of high dimensional spaces are showed on Fig. 1.2.

Classically, when dealing with nonparametric statistics on vector value data, simplifications are necessary, whether independence between data is assumed, which results in summing terms, weighted by parameters which must be tuned or estimated, whether statistics assume to follow a specific model such as Gaussian or Laplacian where a mean vector and correlation matrix is estimated.

Alternative methods shortcuts PDF estimation with graph constructions techniques able to estimate directly statistical measure as a function of total length of k-nearest neighbors (kNN) graphs or minimum spanning tree (MST) graphs [CH04, GLMI05]. These estimates have proved to be very efficient in high dimensions as they shortcut the PDF estimation process and adapts to the underlying manifold of the data. However the lack of knowledge on its differentiation or its connection with PDF estimates make it unpractical for variational problems. Dealing efficiently with high-dimensional dependent data using statistical flexibility in a derivative-based variational problem is a challenge and is the goal of this thesis.



(a) Medical (image courtesy of Prof. Paul Thompson, Lab of Neuro Imaging, UCLA)

(b) Satellite (image courtesy of European Space Agency)

(c) Artificial features: lattice, wavelet

Figure 1.2: *High-dimensional feature space*

1.3 Contributions of this Thesis

The main contribution of this thesis is a new statistical framework inspired from information theory to estimate measures on high dimensional data for variational problems.

Statistical measures are often used in variational problems for similarity matching between two images or regions (tracking, optical flow) or for computing self similarity matching to detect repetitive patterns (regularization, segmentation). In most cases, similarities are estimated over luminance value statistics which do not model properly natural image statistics. Instead, high dimensional spaces are proposed to integrate pixel positions or small image patches in order to better model image the statistics. However, high dimensionality is a curse for statisticians, as spaces are sparse. Statistical estimates designed for high dimensional data are more complex and do not generally satisfy properties required for integration in variational problems (*e.g.* differentiation, computational efficiency).

This thesis proposes a new unified framework based on simple k-nearest neighbors (kNN) search for variational problems in order to estimate efficiently, energies, gradients of these energies, and local probabilities on high dimensional data. This framework relies on a unification between new estimates, and some estimates well-accepted by the image processing community, into a single kNN framework designed for variational problems. Complexity is isolated in the kNN search and can be treated efficiently (for instance using GPU implementation). This framework allows to explore both lower and higher dimensional spaces in variational problems of computer vision.

We revisited the three initial variational problems using the kNN framework, tracking, optical flow and segmentation as well as contributions which takes advantage of high dimensions. In particular, these new dimensions allow to introduce smooth constraints for image matching, smooth regularization for optical flow and define local probabilities or shape priors for segmentation. In general, the kNN framework allows to define information-theoretic energies on multivariate data in variational problems.

We obtained competitive results in the three variational problems in terms of accuracy and quality. Yet, this thesis had two industrial applications. The tracking algorithm presented in Chapter 6 with GPU implementation of the kNN search is being implemented by a cinema post-production company (Mikros Image). The simplified version of the motion segmentation algorithm is implemented in a H264 video coder by the French national telecom operator (Orange).

1.4 Organization

This manuscript is organized in three parts. First part presents state of the art variational solutions of computer vision classify them and expose their general limitations. Second part presents the theoretical contribution of this thesis: we present a general framework to deal or introduce high dimensions in variational problems. Third part revisits the initial three problems of computer vision.

Part I: Methods and ViewPoints

- **Chapter 2** is a presentation of three classical problem of video processing: region of interest tracking, optical flow, and segmentation as well as three examples of resolution and some remaining challenges.

- **Chapter 3** classifies these methods in a deterministic or a statistical viewpoint and exhibits equivalences between both. The statistical viewpoint will be chosen in the thesis.

- **Chapter 4** presents on the statistical viewpoint, classical nonparametric estimation

Part II: High dimensions in variational framework: a new framework

- **Chapter 5** presents the methodological contribution of this thesis: a new framework to deal with high dimensions in variational problems. This framework unifies PDF estimation methods, statistical measures and derivative approximation in a new framework and explore new possibilities in high dimensions.

Part III: Revisiting three variational problems in this framework

- **Chapter 6** proposes a statistical distance in object tracking with a high dimensional appearance model of the object to handle possible deformations.

- **Chapter 7** proposes a new optical flow unified statistical energy for both data and regularization terms.

- **Chapter 8** proposes a framework for high dimensions in active contour segmentation, the new class of similarity measures is applied for various aspects of segmentation, semi-supervised, semi-supervised with priors, motion segmentation, as well as a simplification for video coding applications.

With the purpose of revisiting some variational problems, let us first present three variational problems, their variational formulation and the different mechanisms involved in their resolution.

Part I

Methods and viewpoints

CHAPTER **2**

THREE PROBLEMS OF IMAGE PROCESSING AND VARIATIONAL SOLUTIONS

The focus is on three problems of image processing, object tracking, optical flow, and segmentation. Each problem can be formulated as variational: an energy, also called cost functional is minimized with respect to the unknowns of the problem in order to find a best solution. Let us write a variational solution $\hat{\theta}$ to unknowns θ as the result of the minimization of an energy function E:

$$\hat{\theta} = \arg\min_{\theta} E(\theta) \tag{2.1}$$

These three problems have been chosen to present various variational problems in image processing. Various as these problems use different mechanisms and tools in their resolutions. The tracking problem aims at recovering the new location of an object defined in a previous frame. The unknowns θ are here the new location (position, scale and eventually orientation), thus there are few unknowns (from 2, translation model, to 8, homography model). It can be solved by various derivative-based or derivative-free algorithms. Motion estimation through optical flow problem aims at recovering the motion field between two consecutive images of a video. The unknowns θ are a motion field (of the same size of the image) which leads to minimization in an higher dimensional space and leads to more complex minimization algorithms, often derivative-based. Finally, the segmentation problem aims at grouping pixels into regions, the unknown θ can be for instance a shape which partitions the image. It is possibly infinite dimensional, as continuous shapes have no structure of vectorial space. Its solutions goes through non-standard derivatives called shape derivatives. Summarizing, each problem is different and needs different mathematical tools for resolution.

Each problem is introduced with some of state-of-the-art methods. A variational formulation is given followed by an example of complete resolution of the problem: from the variational formulation to the numerical scheme used for minimization in order to present all the mechanisms involved in the resolution. Finally some recent improvements in the literature of these problems are given to show actual challenges in these methods.

This chapter is organized as follows: Section 2.1 presents the tracking problem, Section

2.2 presents optical flow, Section 2.3 presents the segmentation problem.

2.1 Tracking

Tracking a region of interest (ROI) in a video is still a challenging task. Various high-level applications rely on tracking, *e.g.*, video indexation, object recognition, video surveillance, cinema post-production. . . .

Following the survey [YJS06], tracking can be classified in three different problems.

- *"Point tracking: objects detected in consecutive frames are represented by points and the association of the points is based on the previous object state which can include object position and motion. This approach requires an external mechanism to detect the objects in every frame.* In this thesis, as objects of interest are assumed to be defined by an operator, no detection step is required and we will skip the class of "Point tracking" methods, an example of a state-of-the-art method in point tracking can be found in [ST94]. Another example [GBDB06] proposed to use the framework further defined in this thesis in a "Point tracking problem", however it will not be presented in this manuscript.

- *"Silhouette Tracking: tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution. Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames."* Following its definition, it can be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames, indeed these methods will be treated as a segmentation application in the related parts of this thesis (introductory Section 2.3 and contribution Chapter 8).

- *"Kernel tracking: kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames. This motion is usually in the form of a parametric transformation such as translation, rotation, and affine."* Finally kernel tracking is the method we will study in this section, an example on Fig. 2.1 shows a first frame of a video where an object of interest is defined. The goal is to recover the object of interest in a frame later in the video, several solutions given by different methods are shown.

2.1.1 Introduction

The problem, referred as kernel tracking, can be defined as follows: an ROI is defined in a reference frame of a video and the purpose is to determine in each subsequent frame the region which best matches the ROI in terms of a given similarity measure. Geometrically speaking, the two regions can be deduced from one another by an apparent motion that one usually restricts to a given model. Two classical similarity measures are the Sum of Squared Differences (SSD) or the Sum of Absolute Differences (SAD) between the reference ROI and a candidate region in a target frame.

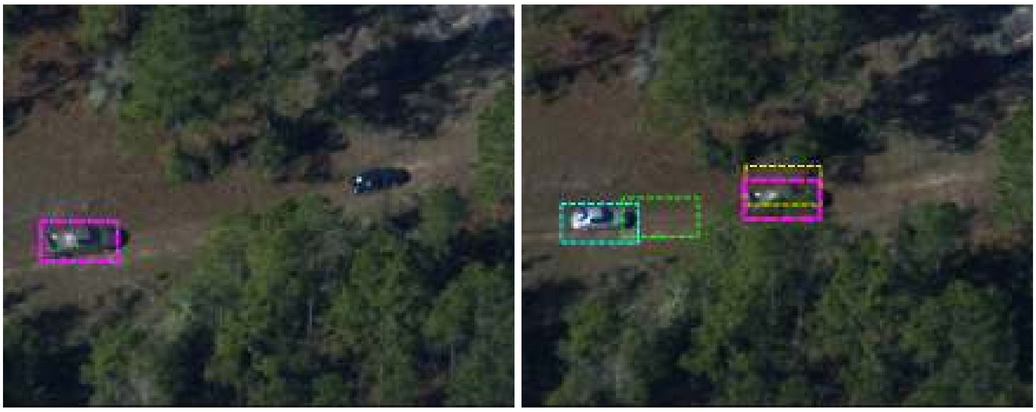


Figure 2.1: *Example on a video from a tracking benchmark [CZT05]. An object of interest is defined on the first frame, and we search for the new position in a frame later in the video: purple algorithm is our method presented in chapter 6. Difficulties: the object is only partially visible, may not have the same size, may have changed appearance . . .*

Similarity measures such as SSD and SAD impose a strict geometric constraint since the underlying residual is computed with a deterministic pixel-to-pixel correspondence between the reference ROI and the target region. In general, this apparent motion follows a rather simple model, so that the estimation of its parameters remains well-posed. Therefore, it is not adapted to complex motions. Moreover, this type of similarity measures corresponds to implicit parametric assumptions on the residual probability density function (PDF) (respectively, Laplacian and Gaussian for the two examples above).

An alternative is to adopt a statistical point of view by building a PDF from the ROI and using it as a template to be compared to a target PDF built from a candidate region by means of a similarity measure. Such statistical methods account for randomness and uncertainty in the observations. At the first level of complexity, the PDFs describe the ROI radiometry [CRM00, PHVG02], either in grayscale or color. However, to improve tracking accuracy, later developments tend to show that more information is required than just color. Different cues were then integrated into the ROI PDF template, *e.g.*, recurring to the use of filters such as spatial derivative filters [Low04, BRDW03, BBPW04], Gabor or wavelet filters [PD02b], and temporal filters [BRDW03, BP07].

2.1.2 Tracking formulation

Let I_{ref} and I_{tgt} be, respectively, the reference frame in which the ROI is (user-)defined and the target frame in which the region which best matches the ROI, in terms of a given similarity measure, is to be searched for. This search amounts to finding the geometric transformation Φ , Fig. 2.2, such that

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_1(I_{\text{ref}}(\Omega), I_{\text{tgt}}(\varphi(\Omega))) \tag{2.2}$$

where $\mathfrak{D}_1$ is a similarity measure between two data sets and Ω is the domain of the ROI. Domain Ω is a subset of $\mathbb{R}^2$ or a subset of $\mathbb{N}^2$ in the discrete framework.

For clarity, the reference data set $I_{\text{ref}}(\Omega)$ will be denoted by R and the target data set $I_{\text{tgt}}(\varphi(\Omega))$ will be denoted by T_φ. Thus, $R(i)$ and $T_\varphi(i), i \in \Omega$, represent corresponding samples from their respective regions.

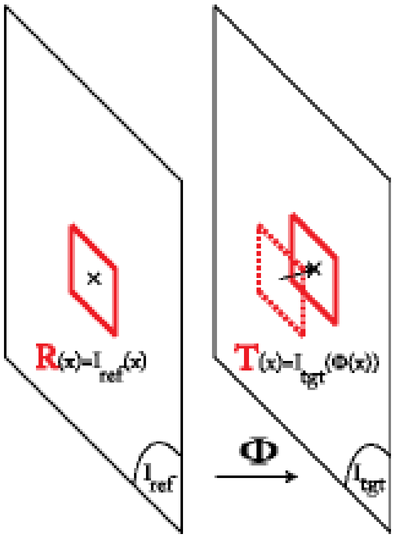


Figure 2.2: *Tracking is finding the geometric transformation of a ROI between two frames*

In order to account for observation uncertainties, the geometric transformation Φ is obtained at the minimum of a similarity measure expressed as a distance between two PDFs

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_2(f_R, f_{T_\varphi}) \tag{2.3}$$

where f_R, respectively f_{T_φ}, is the PDF which generated the samples $\{R(i), i \in \Omega\}$, respectively $\{T_\varphi(i), i \in \Omega\}$. These PDF are unknown and need to be estimated from the samples R and T. These PDF estimates from samples will be noted $\hat{f}_R$ and $\hat{f}_{T_\varphi}$. Whenever appropriate, U will be used as a generic notation for either R or T_φ. Traditionally, $U(i)$ is a triplet of color components in a given color space. Some geometric information can be integrated to this feature vector. Generally speaking, the samples will be regarded as elements of $\mathbb{R}^d$.

The similarity measure $\mathfrak{D}_2$ defined in the Mean-shift tracking problem [CRM00, Com03] is based on the sample estimate of the Bhattacharyya coefficient,

$$E(\varphi) = \sqrt{1 - \rho(\hat{f}_{T_\varphi}, \hat{f}_R)} \tag{2.4}$$

where ρ is the sample estimate of the Bhattacharyya coefficient

$$\rho(\hat{f}_{T_\varphi}, \hat{f}_R) = \int_{\mathbb{R}} \sqrt{\hat{f}_{T_\varphi} \hat{f}_R}. \tag{2.5}$$

PDFs are defined as a weighted histograms, the histogram construction is combined with a Gaussian weighting of the samples according to their distance to the center of the ROI [CRM00, PHVG02]. This Gaussian weighted histogram gives more importance to pixel at the center of the ROIs than pixels on the borders which are less reliable. It can be also seen as a radial layout constraint in the histogram construction.

2.1.3 Resolution: an example

Target localization is then performed by maximizing the Bhattacharyya coefficient (2.4) between the reference distribution f_R and the target distribution f_{T_φ}. This maximization was efficiently performed through a gradient-ascent manner algorithm called mean-shift [CRM00]. The mean-shift procedure is an algorithm that converges to the closest mode in the PDF [FH75].

It can be summarized as follows, let a nonparametric estimate of a PDF of this form

$$\hat{f}_K(\mathbf{x}) = \frac{1}{Nh^d} \sum_{i=1}^{N} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right). \tag{2.6}$$

where x_i are the data, d is the dimension of data, k is a kernel profile (*e.g.* Gaussian), and N is the number of pixels in the region R. The normalized gradient of this PDF leads to a mean-shift expression.

$$\frac{\hat{\nabla} f_K(\mathbf{x})}{\hat{f}_G(\mathbf{x})} = \frac{2}{h^2.C} M_{h,G}(\mathbf{x}) \tag{2.7}$$

where G is another kernel with profile k' depending on kernel K, C is a constant, $M_{h,G}$ is the sample mean shift vector

$$M_{h,G}(\mathbf{x}) = \frac{\sum_{i=1}^{N} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{N} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}. \tag{2.8}$$

The mean-shift expression on x is then a shift between a point x and a weighted mean in a local window. Shifting x sequentially with the mean-shift expression, it converges to the closest mode of the distribution.

This idea is applied in tracking. In order to maximize the Bhattacharyya coefficient (linearized at each step), the location of the target is shifted according to the sample mean shift thus converges toward the closest mode of the PDF. In tracking, this closest mode corresponds to the spatial match between the reference and target distribution.

2.1.4 Challenges

The way to define PDFs (2.5) as weighted histogram has the advantage not to add any dimension to the feature space. However, it lacks generality. Geometry can instead be added directly to the radiometric vector (or any other feature vector), *e.g.*, in the form of the Cartesian coordinates of the pixels of the ROI [EDD03]. Independence between color and geometry cannot be assumed in order to avoid to manipulate high-dimensional PDFs. Indeed, geometry alone, seen as a random variable, follows a uniform distribution whether in the ROI or in the target region and, therefore, brings no information. While considering color and geometry jointly, simplification can still be achieved by approximating the PDFs with parametric laws.

In general, the main difficulties in the kernel tracking problem are to define a template as accurate as possible in order to characterize the object of interest, but flexible enough in order to accounts for deformation of this object. Other challenges are to define a meaningful distance between two templates and minimize it toward the new location of the ROI under various video difficulties (occlusions). This minimization can be performed with several and robust algorithms, for instance complex derivative-free optimization algorithms, as the number of parameters of the transformations φ is in general limited (2 parameters for translation, 1 for scaling in most cases).

This is not the case in an other motion estimation problem, where the number of unknowns is much higher, optical flow.

2.2 Optical flow

Optical flow is the apparent motion of image pixels or regions from one frame to the next frame in a video sequence. The apparent motion representation is a vector field describing the new location of pixels from the first frame in the second frame of the video. An illustration on Fig. 2.3 shows two frames of a video sequence and a color representation of its estimated flow field: each color represents a vector direction. Optical flow is useful in many applications of image processing. In compression, rather than coding images twice, one can code the image and the optical flow field needed to deduce the second image. In computer vision, optical flow can be used for scene understanding, object tracking or segmentation of moving objects.

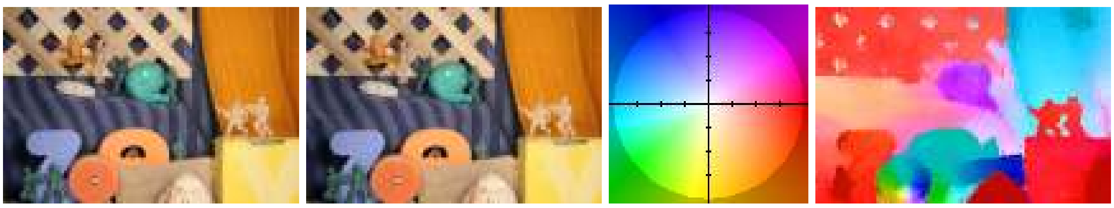


Figure 2.3: *Two frames and color code for representing the flow field direction from an optical flow benchmark [BRS$^+$07], estimated optical flow with our method presented in chapter 7. Challenging problem: non smooth field, diverging fields ...*

Optical flow estimation is a challenging problem for two reasons: on one hand, the visible pixels in the first frame may not be visible in the second frame, this is the occlusion problem. Indeed, some pixels in the second frame may have been hidden in the first frame by another object or because it moved outside of image boundaries. Some other pixels may still be visible in the second frame but may have changed of appearance (luminance value). On the other hand motion vector can only be recovered in the direction of the image gradients, it is ambiguous in other directions. This is the aperture problem: for example in homogeneous zones, many motions vectors lead to pixels with similar appearance value. For these reasons, optical flow estimation still remains a challenging problem.

Let us now present the optical flow estimation problem on a mathematical point of view.

2.2.1 Introduction

In the estimation of a dense flow field (one motion vector per pixel), there are as many unknowns as the number of pixels. Moreover, the solution of this estimation is not unique because of the aperture issue, so the problem has to be constrained. It is ill-posed in the sense of Hadamard, and needs to be regularized. On one hand, some methods keep a dense vector flow field as unknown but impose smooth conditions on the flow field. These methods are called global optical flow computation [HS81, WS01]. On the other hand, local methods constrain the motion vector flow field to follow a parametric motion assumption (*e.g.* translation, affine, homography) on the whole image or on some parts of the image (blocks, regions) [LK81, OB95, WK93]. Finally a method [BWS05] proposed to combine the advantages of the two approaches : a dense flow field like in global methods is combined with the robustness to noise of local methods. Another recent method propose to combine the latter approach with a better model for statistics of the flow field [RB05].

2.2.2 Optical flow formulation

Motion between two frames of a video I_n and I_{n+1} can be computed by minimizing a function of the residual over the image domain D. At a pixel level, making the assumption of brightness constancy, the residual is classically equal to the following residual

$$I_n(\mathrm{m}) - I_{n+1}(\mathrm{m} + \mathrm{v}(\mathrm{m})) \tag{2.9}$$

where $I_n(\mathrm{m})$ is the feature of the image from the n^{th} frame of a video sequence at pixel location m, basically the luminance value, and $\mathrm{v}(\mathrm{m}) = [\mathrm{v}_1(\mathrm{m}), \mathrm{v}_2(\mathrm{m})]$ is the apparent motion between I_n and I_{n+1} at pixel location m (called optical flow). Ideally, this residual is equal to zero up to some noise. The optical flow constraint ϵ_{OFC} is then obtained by a first order Taylor expansion, assuming small displacements relatively to the image derivatives:

$$(I_x(\mathrm{m})\mathrm{v}_1(\mathrm{m}) + I_y(\mathrm{m})\mathrm{v}_2(\mathrm{m}) + I_t(\mathrm{m})). \tag{2.10}$$

Optical flow can be retrieved by minimizing the following quadratic constraint

$$\epsilon_{\mathrm{OFC}}(\mathrm{m}, \mathrm{v}(\mathrm{m})) = (I_x(\mathrm{m})\mathrm{v}_1(\mathrm{m}) + I_y(\mathrm{m})\mathrm{v}_2(\mathrm{m}) + I_t(\mathrm{m}))^2 = \mathrm{w}(\mathrm{m})^T J_0(\nabla_3 I)\mathrm{w}(\mathrm{m})) \tag{2.11}$$

where I_x I_y and I_t represents the spatial and temporal partial derivatives of the image, where w is the spatio-temporal optical flow $\mathrm{w}(\mathrm{m}) = [\mathrm{v}_1(\mathrm{m})\ \mathrm{v}_2(\mathrm{m})\ 1]$ and where $J_0 = \nabla I \nabla I^T$ is a matrix of first order derivatives: $J_{11} = I_x^2$, $J_{12} = I_x I_y \ldots$

In grayscale, this condition provides a single equation for two unknowns (the components of $\mathrm{v}(\mathrm{m})$) and, both in grayscale and color, it is likely that several pixels in $I_{\mathrm{n}+1}$ have the same value than $I_n(\mathrm{m})$. As a consequence, motion estimation problem cannot be solved without additional constraints, the solution needs to be regularized. A possible way to constrain the solution is to suppose that motion is coherent with a chosen model inside a small neighborhood [LK81, OB95, WK93]. For instance, one can assume coherence in a Gaussian window by smoothing the square of optical flow constraint with a Gaussian kernel K_σ.

$$\epsilon_{\mathrm{LK}}(\mathrm{m}, \mathrm{v}(\mathrm{m})) = K_\sigma * \epsilon_{\mathrm{OFC}}(\mathrm{m}, \mathrm{v}(\mathrm{m})) = \mathrm{w}(\mathrm{m})^T J_\sigma(I)\mathrm{w}(\mathrm{m})) \tag{2.12}$$

where $J_\sigma = K_\sigma * \nabla I \nabla I^T$

In order to define a global functional, the optical flow constraint ϵ then sums over D to a cost functional E_{Data}:

$$E_{\mathrm{Data}}(\mathrm{v}) = \int_D \Psi(\epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m})))\mathrm{dm} \tag{2.13}$$

where ϵ is a residual function and can be either ϵ_{OFC} (2.11) or ϵ_{LK} (2.12), where $\Psi(x^2)$ is a non quadratic penaliser.

A more common way to regularize the optical flow field is to impose smooth conditions on the vector field. The regularization term then adds in the cost functional a penalty on the gradient of the vector field.

$$E_{\mathrm{Smooth}}(\mathrm{v}) = \int_D \Psi(|\nabla \mathrm{v}_1(\mathrm{m})|^2 + |\nabla \mathrm{v}_2(\mathrm{m})|^2)\ \mathrm{dm} \tag{2.14}$$

where $\Psi(x^2)$ is again a non quadratic penaliser. In addition, function Ψ has to be chosen carefully to smooth the flow field with edge preservation. This point will be detailed in the next chapter. Finally, motion field v is estimated by minimizing the following energy:

$$\begin{aligned} E(\mathrm{v}) &= E_{\mathrm{Data}}(\mathrm{v}) + \alpha E_{\mathrm{Smooth}}(\mathrm{v}) \\ &= \int_D \Psi(\epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m})))\mathrm{dm} + \int_D \Psi(|\nabla \mathrm{v}_1(\mathrm{m})|^2 + |\nabla \mathrm{v}_2(\mathrm{m})|^2)\ \mathrm{dm} \end{aligned} \tag{2.15}$$

2.2.3 Resolution: an example

Euler Lagrange equations

As images and motion fields are presented as continuous functions. Calculus of variations is used to find the minimum of this functional. The Euler Lagrange equation of energy (2.15) reads

$$(\Psi'_D)(J_{12}(\mathrm{m})\mathrm{v}_1(\mathrm{m}) + J_{12}(\mathrm{m})\mathrm{v}_2(\mathrm{m}) + J_{13}(\mathrm{m})) - \alpha\,\mathrm{div}((\Psi'_S)\nabla\mathrm{v}_1(\mathrm{m})) = 0 \tag{2.16}$$

$$(\Psi'_D)(J_{12}(\mathrm{m})\mathrm{v}_1(\mathrm{m}) + J_{22}(\mathrm{m})\mathrm{v}_2(\mathrm{m}) + J_{23}(\mathrm{m})) - \alpha\,\mathrm{div}((\Psi'_S)\nabla\mathrm{v}_2(\mathrm{m})) = 0 \tag{2.17}$$

where $(\Psi'_D) = \Psi'(\mathrm{w}^T(\mathrm{m})J(\nabla_3 I)(\mathrm{m})\mathrm{w}(\mathrm{m}))$, J depends on the residual term used in the data term, it is J_0 with ϵ_{OFC} (2.11), or J_σ with ϵ_{LK} (2.12), and $\Psi'_S = \Psi'(|\nabla\mathrm{v}_1(\mathrm{m})|^2 + |\nabla\mathrm{v}_2(\mathrm{m})|^2)$

Minimization

Optical flows are in general solved as solutions of linear systems as it converges faster and with more accuracy to a minimum than classical gradient descend. Therefore, a matrix A and a vector B are defined:

$$A.\begin{bmatrix} \mathrm{v}_1 \\ \mathrm{v}_2 \end{bmatrix} = B. \tag{2.18}$$

These two matrices are defined from the Euler Lagrange equations (2.16) and (2.17). Therefore, the derivative equations (2.16) and (2.17) needs to be linearized. Robust functions Ψ (2.13) and (2.14) (and their derivatives) are in general nonlinear and are estimated using fixed point iterations [BBPW04].

Linear system solution and stability

An iterative scheme, Successive Over Relaxation (SOR) (2.19), is then applied to solve the linear system (2.18). SOR has convergence proofs for diagonally dominant matrices.

The scheme at row i and iteration l reads

$$\mathrm{v}_i^{l+1} = (1-w)\mathrm{v}_i^l + w\frac{B(i) - \sum_{j=1}^{j<i} A(i,j)\mathrm{v}_i^{l+1}(j) - \sum_{j=i+1}^{j=n} A(i,j)\mathrm{v}_i^l(j)}{A(i,i)} \tag{2.19}$$

where w is the relaxation parameter $0 < w < 2$, $w = 1$ is the Gauss-Seidel scheme.

The system ends up with:

$$\mathrm{v}_{1i}^{l+1} = (1-w)\mathrm{v}_{1i}^l + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^l_{i\sim j}\mathrm{v}_{1j}^{l+1} + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^l_{i\sim j}\mathrm{v}_{1j}^l - \frac{(\Psi'_D)^l_i}{\alpha}(J_{12\mathrm{i}}\mathrm{v}_{2i}^l + J_{13\mathrm{i}})}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+} (\Psi'_S)^l_{i\sim j} + \frac{(\Psi'_D)^l_i}{\alpha}J_{11\mathrm{i}}} \tag{2.20}$$

$$\mathrm{v}_{2i}^{l+1} = (1-w)\mathrm{v}_{2i}^l + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)^l_{i\sim j}\mathrm{v}_{2j}^{l+1} \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)^l_{i\sim j}\mathrm{v}_{2j}^l - \frac{(\Psi'_D)^l_i}{\alpha}(J_{12\mathrm{i}}\mathrm{v}_{2i}^l + J_{23\mathrm{i}})}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+} (\Psi'_S)^l_{i\sim j} + \frac{(\Psi'_D)^l_i}{\alpha}J_{22\mathrm{i}}} \tag{2.21}$$

where $\mathcal{N}_i^-$ denotes the neighbors j of i with $j < i$ and $\mathcal{N}_i^+$ the neighbors j of i with $j > i$, J_{abi} are the component (a, b) of the structure tensor $J(\nabla_3 I) = \nabla_3 I\nabla_3 I^T$ at some pixel i, ϵ is

ϵ_{OFC}, $(\Psi'_D)^l_i$ and $(\Psi'_S)^l_{i\sim j}$ are discrete diffusity terms and are estimated through fixed point iterations, more details can be found in [BBPW04].

Multi resolution

The classical optical flow formulation contains a first order Taylor expansion to linearize ϵ. Thus, the flow must be small relatively to the image derivatives. A multi resolution scheme is then chosen: the flow is computed on down-sampled versions of the image, respecting the Shannon theorem. Instead of choosing the optical flow at lower resolution to initialize the next resolution $k+1$, we warp the image with the flow at resolution k and we initialize the flow at 0: $\mathrm{v}^{k+1} = \mathrm{v}^k + \mathrm{dv}^k$, where v^k is the optical flow obtained form the lower resolution and used to warp the image, then we are looking for a new flow dv^k. A theoretical justification can be found in [BBPW04]. The scheme at multiresolution level k, iteration l, and row i reads

$$\mathrm{v}_{1i}^{k,l+1} =(1-w)\mathrm{v}_{1i}^{k,l} + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)}(\Psi'_S)^l_{i\sim j}\mathrm{v}_{1j}^{k,l+1} + \sum\limits_{j\in\mathcal{N}^+(i)}(\Psi'_S)^l_{i\sim j}\mathrm{v}_{1j}^{k,l} - \frac{(\Psi'_D)^l_i}{\alpha}(J_{12\mathrm{i}}\mathrm{v}_{2i}^{k,l} + J_{13\mathrm{i}})}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+}(\Psi'_S)^l_{i\sim j} + \frac{(\Psi'_D)^l_i}{\alpha}J_{11\mathrm{i}}} \tag{2.22}$$

$$\mathrm{v}_{2i}^{k,l+1} =(1-w)\mathrm{v}_{2i}^{k,l} + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)}(\Psi'_S)^l_{i\sim j}\mathrm{v}_{2j}^{k,l+1} \sum\limits_{j\in\mathcal{N}^+(i)}(\Psi'_S)^l_{i\sim j}\mathrm{v}_{2j}^{k,l} - \frac{(\Psi'_D)^l_i}{\alpha}(J_{12\mathrm{i}}\mathrm{v}_{2i}^{k,l} + J_{23\mathrm{i}})}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+}(\Psi'_S)^l_{i\sim j} + \frac{(\Psi'_D)^l_i}{\alpha}J_{22\mathrm{i}}} \tag{2.23}$$

This scheme converges after several iterations to a realistic and smooth flow field.

2.2.4 Challenges

On the data term, two kinds of methods have recently enhanced the quality of optical flow estimation.

The first one is based on the idea to constrain motion to be constant within a small neighborhood. Bruhn *et al* [BWS05] minimize the quadratic form $\mathrm{w}^T(\mathrm{m}).J_\sigma(\nabla_3 I(\mathrm{m})).\mathrm{w}(\mathrm{m})$ where $\mathrm{w}(\mathrm{m}) = [\mathrm{v}(\mathrm{m})\, 1]$, $J_{\sigma'}(\nabla f) = K_{\sigma'} * (\nabla_3 I\, \nabla_3 I^T)$ is the structure tensor [BWS05]. This solution is an integration of local methods [LK81] in global functionals [HS81]

The second one [BBPW04] is to combine other constraints than just brightness consistency. The authors have added gradient consistency. The image features I are now multi dimensional $(I, \nabla I)$ and the different components in the data energy (2.13) are added using different weightings.

Other works [RB05] modified the regularization term. They studied the statistics of optical flow fields and they deduced a prior that captures the rich statistics of optical flow patches (3×3 or 5×5).

Numerical schemes involved here are complex due to the number of unknowns, they use efficient numerical solvers with proofs of convergences under some conditions on the matrix (diagonal dominance). These methods are numerically complex and need in general an accurate knowledge of derivatives of the energies in variational problems. Last variational problem presented is segmentation.

2.3 Segmentation

2.3.1 Introduction

Segmentation problem is partitioning the image into semantic objects. The segmentation problem is reduced in this thesis to the extraction from the background of one or several objects of interests in an image or in a video sequence. By extraction, we mean that not only the localization or bounding box of the object has to be recovered but the mask of the object (a matrix of same size as the image containing 1 if a specific pixel belongs to the object or 0 if the pixel does not belong to the object). The most common application of segmentation is scene compositing: the object is extracted for its original background and pasted on a new background. The object or region of interest has no general definition and is application dependent. In cinema, the object of interest can be an actor filmed in a studio, extracted and pasted into an outside scene. In medical imaging, object of interest can be a tumor and one must watch its evolution in size and in color.

Segmentation without priors on the object to find, is still a very challenging task. A more realistic solution is to let an operator introduces priors about the object to segment. These priors may be the location of the object, its color properties, its shape. An example is show on Fig. 2.4, a user gives a box where he believes the object of interest is and the algorithm extract the object based on this information.

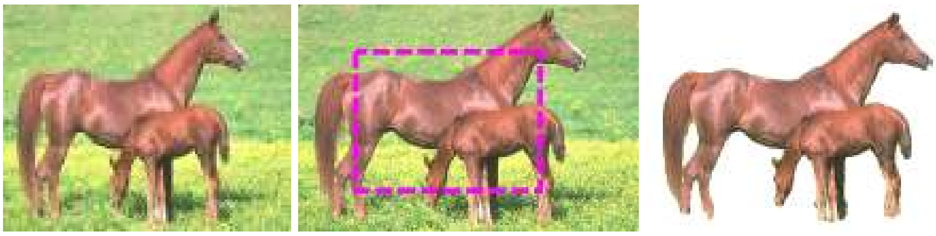


Figure 2.4: *Example on an image from a segmentation benchmark [MFTM01]. There is a prior about the location of the object defined by a dashed box (middle image). The result of the segmentation algorithm presented in chapter 8 is then able to extract the object of interest from the image. Difficulties: color histogram of object and background overlaps, object is non uniform in luminance...*

2.3.2 Segmentation formulation

Segmentation methods can be classified in two categories. The first category detects high-gradients or gradient discontinuities in the image. These methods are called contour-based segmentation. One can for instance threshold the norm of the gradient or use more complex filters (Sobel, Prewitt, Canny-Deriche filter). The second category is region-based segmentation and computes region characteristics inside partitions. Methods can be for instance thresholding, mean-shift analysis, region growing, watersheds or clustering (for instance k-means).

The framework, used for contour-based and region-based segmentation, is active contours: it initializes a deformable contour and computes an evolution force V to deform the contour to a solution in order to minimize an energy. This is illustrated on Fig. 2.5.

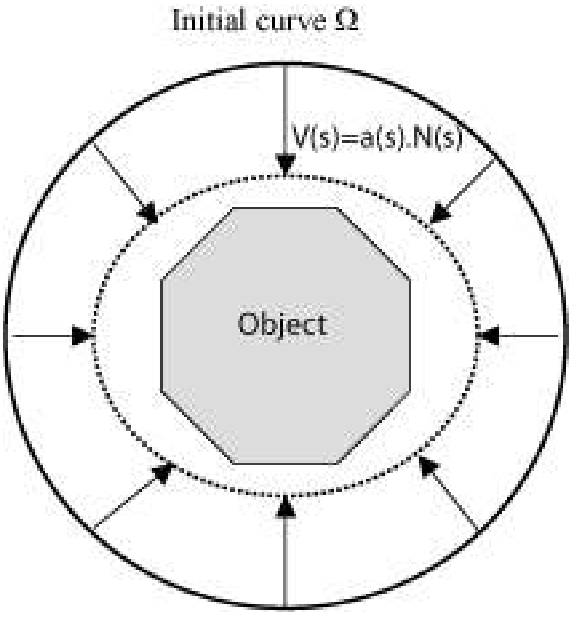


Figure 2.5: *Active contour evolution: a deformable contour is initialized and evolves in order to minimize an energy*

Contour-based active contours

A contour-based energy minimizes a function over the contour Γ.

$$E_{\text{boundary}}(\Gamma) = \int_{\Gamma} k(s)\ \mathrm{d}s \tag{2.24}$$

where $k(s)$ is a function called descriptor which has to be minimized over the contour.

Kass *et al.* [KWT87] proposed the first snake model composed of three terms. The two first terms are rigidity terms which constrain the solutions of the contour. These terms are regularization terms. The last term is the data term, it maximizes the gradient value over the contour.

$$E^{Snake} = \alpha \int_0^1 |\Gamma'(s)|^2\ \mathrm{d}s + \beta \int_0^1 |\Gamma''(s)|^2\ \mathrm{d}s - \lambda \int_0^1 |\nabla I(\Gamma(s))|\ \mathrm{d}s \tag{2.25}$$

where α, β and λ are some weights which must be manually tuned.

Caselles *et al.* [CKS97] reformulated the problem as finding the curve of minimal geodesic length in a Riemannian space whose metric is induced by image gradients. This reformulation cancels the second rigidity term and added a decreasing function g of the gradient.

$$E^{Geodesic} = \int_0^1 |\Gamma'(s)|^2\ \mathrm{d}s + \alpha \int_0^1 g(|\nabla I(\Gamma(s))|)^2\ \mathrm{d}s \tag{2.26}$$

Region-based active contours

Rather than minimizing a function of the gradient of the contour boundary, region-based active contours focus on the inner region Ω of contour Γ.

$$E_{\text{region}}(\Gamma) = \int_{\Omega} k(\mathrm{m}, \Omega)\ \mathrm{dm} \tag{2.27}$$

where $k(\mathrm{m}, \Omega)$ is a region descriptor.

In practice, this term is also combined with a boundary-based constraint, in particular, as a means to obtain smooth contours

$$E(\Gamma) = E_{\text{region}} + E_{\text{boundary}}. \tag{2.28}$$

Among the first region-based energies, Mumford and Shah [MS89] have defined a functional with two unknowns, a piecewise smooth representation u of an image I, for instance an ideal noiseless version of the image. They defined the following functional:

$$E^{MS}(u,\Gamma) = \int_{\Omega} (I(\mathrm{m}) - u(\mathrm{m}))^2 \,\mathrm{dm} + \int_{\Omega\backslash\Gamma} |\nabla u(\mathrm{m})|^2 \,\mathrm{dm} + \nu|\Gamma| \tag{2.29}$$

Assuming the representation of the image u is piecewise constant and equal to the mean μ over the region Ω Chan et Vese [CV01] have proposed the following energy

$$E^{CV}(u,\Gamma) = \int_{\Omega} (I(\mathrm{m}) - \mu(\Omega))^2 \,\mathrm{dm} + \int_{\overline{\Omega}} (I(\mathrm{m}) - \mu(\overline{\Omega}))^2 \,\mathrm{dm} + \nu|\Gamma| \tag{2.30}$$

The regularization term on ∇u vanishes to zero as u is constant in $\Omega\backslash\Gamma$. The shape Γ that minimizes the Mumford Shah energy is the null shape. In order to avoid this problem, a region competition algorithm can be defined, minimizing an integral on the region Ω plus an integral on the background $\overline{\Omega}$..

Statistical formulation in active contours were introduced by a Bayesian formulation [ZY96]. The Bayes rule writes the probability of an unknown θ given an observation I:

$$p(\theta|I) = \frac{p(I|\theta)}{p(I)} p(\theta) \tag{2.31}$$

$$p(\theta|I) \propto p(I|\theta)p(\theta) \tag{2.32}$$

where the observation I is the image, and where θ is a partition of this image into two regions $\theta = \{\Omega, \overline{\Omega}\}$.

This Bayesian formulation considers the object and the background as separated random variables:

$$p(I|\theta) = p(I|\{\Omega,\overline{\Omega}\}) = \prod_{\mathrm{m}\in\Omega} p(I(\mathrm{m})|\Omega) \prod_{\mathrm{m}\in\overline{\Omega}} p(I(\mathrm{m})|\overline{\Omega}) \tag{2.33}$$

as each pixel is considered as an independent realization of a random variable inside each region. $p(\theta)$ is a simple prior on image partition as a penalty on contour length:

$$p(\theta) = p(\{\Omega,\overline{\Omega}\}) = e^{-\nu|\Gamma|} \tag{2.34}$$

Maximizing the log probability of a partition given an image $p(\{\Omega,\overline{\Omega}\}|I)$ is equivalent to maximize the log probability of an image knowing its partition plus the log probability of this partition. The energy finally writes

$$E^{B}(\Gamma) = -\int_{\Omega} \log(f_{\Omega}(I(\mathrm{m}))) \,\mathrm{dm} - \int_{\overline{\Omega}} \log(f_{\overline{\Omega}}(I(\mathrm{m}))) \,\mathrm{dm} + \nu|\Gamma| \tag{2.35}$$

where $f_{\Omega}(I(\mathrm{m})) = p(I(\mathrm{m})|\Omega)$ is the probability that one pixel $I(\mathrm{m})$ belongs to a region Ω equal to the probability of this pixel in the distribution f estimated over Ω. Choosing for f_{Ω} a Gaussian estimate, the formulation of this energy is equivalent to (2.30). Links between deterministic and statistical methods will be treated in details in next chapter.

Geodesic active regions model [PD02b] extended Bayesian formulations with a contour-based term

$$E^{GAR}(\Gamma) = -\int_{\Omega} \log(f_{\Omega}(I(\mathrm{m})))\ \mathrm{dm} - \int_{\overline{\Omega}} \log(f_{\overline{\Omega}}(I(\mathrm{m})))\ \mathrm{dm} + \nu \int_0^1 |\Gamma'(s)| g(.)\ \mathrm{d}s \tag{2.36}$$

Finally, more recent approaches relaxed the underlying Gaussian assumption on the PDF and estimates it with a nonparametric model. Nonparametric estimation as the energy are based on information theory measures, mutual information [KFY$^+$05] or entropy [HSD$^+$06], estimated through Parzen windowing techniques

$$E^{NP}(\Gamma) = -\int_{\Omega} \log(f_{\Omega}(I(\mathrm{m})))\ \mathrm{dm} - \int_{\overline{\Omega}} \log(f_{\overline{\Omega}}(I(\mathrm{m})))\ \mathrm{dm} + \nu|\Gamma| \tag{2.37}$$

with

$$f_{\Omega}(u) = \frac{1}{|\Omega|} \int_{\Omega} K_h(u - I(\mathrm{m}))\ \mathrm{dm}. \tag{2.38}$$

where K_h is a Gaussian kernel of standard deviation h.

2.3.3 Resolution: an example

In this section we will focus on the general derivation of a Bayesian segmentation problem with nonparametric distributions as formulated in (2.37).

Shape derivative

Ignoring the region competition term, which derivative is similar, the energy (2.37) can be simply written as

$$E(\Gamma) = -\frac{1}{|\Omega|} \int_{\Omega} \log f_{\Omega}(I(\mathrm{m}))\ \mathrm{dm} \tag{2.39}$$

where the distribution function f_{Ω} is estimated non parametrically over Ω

$$f_{\Omega}(u) = \frac{1}{|\Omega|} \int_{\Omega} K_h(u - I(\mathrm{m}))\ \mathrm{dm} \tag{2.40}$$

The definition of the shape derivative of (2.39) is based on a domain transformation T whose amplitude continuously depends on a parameter τ such that $T(\Omega, \tau = 0)$ is equal to Ω and $T(\Omega, \tau)$ is equal to $\Omega(\tau)$ [DZ01, HR04, JBBA03, ABFJB03]. Functions of Ω, or Γ, can then be rewritten as functions of τ. In this context, the shape derivative of

$$E(\Gamma) = \int_{\Omega} G(\Gamma, \mathrm{m})\ \mathrm{dm} \tag{2.41}$$

is equal to

$$\mathrm{d}E(\Gamma, F) = \frac{\mathrm{d}E}{\mathrm{d}\tau}(\tau = 0) = \int_{\Omega} \frac{\partial G}{\partial \tau}(\tau = 0, \mathrm{m})\ \mathrm{dm} - \int_{\Gamma} G(\Gamma, s)\ N(s) \cdot F(s)\ \mathrm{d}s \tag{2.42}$$

where F is a vector field defined on Γ and linked to T, s is the arclength parameter of Γ, $G(\Gamma, s)$ is a short notation for $G(\Gamma, \Gamma(s))$, and N is the inward unit normal of Γ.

Detailed in appendix E, the shape derivative of (2.39) is equal to

$$\begin{aligned} \mathrm{d}E(\Gamma, F) \quad = \quad & \frac{1}{|\Omega|} \int_{\Gamma} \Bigg(E(\Gamma) - 1 + \log f_{\Omega}(I(s)) \\ & + \frac{1}{|\Omega|} \int_{\Omega} \frac{K_{\sigma}(I(s) - I(\mathrm{m}))}{f_{\Omega}(I(\mathrm{m}))}\ \mathrm{dm} \Bigg)\ N(s) \cdot F(s)\ \mathrm{d}s\ . \end{aligned} \tag{2.43}$$

The shape derivative (2.43) has the following form

$$\mathrm{d}E(\Gamma, F) = \int_{\Gamma} ((\alpha(s))\ N(s)) \cdot F(s)\ \mathrm{d}s = \langle \alpha\ N, F \rangle \tag{2.44}$$

where $\langle , \rangle$ is the L^2-inner product on Γ. Therefore, $\alpha\ N$ is, by definition, the gradient of (2.39) at Γ associated with this inner product.

Based on the notion of gradient defined in (2.42), energy (2.39) can be minimized using a steepest descent procedure in the space of contours. The following contour evolution process is known as the active contour technique [CKS97, HR04]: an initial contour[1] is iteratively deformed in the opposite direction of the gradient until a convergence condition is met. The evolution equation of the active contour is written as follows

$$\begin{cases} \Gamma(\tau = 0) = \Gamma_0 \\ \dfrac{\partial \Gamma}{\partial \tau} = (\alpha^c - \alpha)\ N \end{cases} \tag{2.45}$$

where τ is the evolution parameter and α^c has the same expression as α but is evaluated on $\overline{\Omega}$. The convergence condition is $\alpha^c - \alpha = 0$.

Implementation: explicit versus implicit representations

The explicit representation is the original formulation in active contours [KWT87], the active contour is parametrized by a polygon or a spline driven by control points. The evolution consists in applying a force (2.45) to the control points and rebuilding the spline.

Implicit representations were popularized by level sets [OS88]. Rather than considering explicitly the curve as a spline, the active contour is a zero level of a function of higher dimension. A common choice for this function of higher dimension is the signed distance function to the active contour.

These two representations of active contours have different benefits and drawbacks. The explicit representation (splines) are computationally efficient as the evolution force is only computed and applied on a few points of the curve, however topological changes like curves auto intersection must be controlled by an additional algorithm [PBBU05]. The implicit representation of active contours naturally adapts to topological changes, however, the evolution force has now to be computed on all the points of the image or in a narrow band around the evolving contour.

2.3.4 Challenges

Recent works [RBD03] proposed to add additional cues, such as the structure tensor, in the probability model for the region. Additional shape priors [RP02, CKS03] were also introduced, for instance as a distance to some reference shape. Other cues can be extracted from motion [CS05, BHD$^+$07]. For instance we defined an energy as a joint entropy between a spatial term E_s and a temporal term E_t. The spatial term is basically an entropy on image luminance, and the temporal term is on a residual e_n from the motion $\mathrm{v}(\Gamma)$ estimated of the inner region defined by contour Γ.

$$\begin{aligned} E(\Gamma) &= -\frac{1}{|\Omega|} \int_{\Omega} \log f_t(e_n(v(\Gamma), \mathrm{m}))\ \mathrm{dm} - \frac{1}{|\Omega|} \int_{\Omega} \log f_s(I_n(\mathrm{m}))\ \mathrm{dm} \\ &= E_t(\Gamma) + E_s(\Gamma) \end{aligned} \tag{2.46}$$

[1] For example, a user-defined contour.

This energy will be presented in Chapter 8.

2.4 Conclusion

In this chapter, three different variational problems have been presented with three different resolutions. For example, tracking has a low number of unknowns, allowing a lot of flexibility in the possibilities of resolution. In particular, statistical methods are largely used in state of the art. On the other hand, optical flow has a high number of unknowns and its resolution often goes through derivative knowledge of the energy (Euler Lagrange equations if the problem is formulated in the continuous domain). Deterministic energies with knowledge of derivatives and convergence guaranties are then largely preferred in the literature to statistical energies. This comparison deterministic-statistical will be discussed in the following chapter.

Another important point often encountered in these three variational problems is dimensionality. Recent methods in the literature often integrate other features than just RGB color. In order to keep generality, the notation $u(\mathrm{m})$ for features will be preferred to $I(\mathrm{m})$ for image luminance.

CHAPTER 3

DETERMINISTIC OR STATISTICAL VIEWPOINTS

The previous detailed approaches can be classified in two different viewpoints: deterministic and statistical. The energy can be formulated directly as a function of the data U (deterministic viewpoint)

$$\hat{\theta} = \arg\min_{\theta} E(U, \theta) \tag{3.1}$$

or the energy is formulated as a function of a PDF built on the data f_U (statistical viewpoint)

$$\hat{\theta} = \arg\min_{\theta} E(U, f_U, \theta). \tag{3.2}$$

For each viewpoint, methods can be classified in two groups. On one hand some methods assume a model for the image and match the model by computing a small set of parameters (*e.g.* parametric PDFs on the statistical viewpoint). On the other hand some methods generate their own model from the image (*e.g.* nonparametric PDFs on the statistical viewpoint). The latter methods are more accurate in the general case as they do not assume a particular model.

These two viewpoint exhibit strong connections, in the rest of this thesis the statistical viewpoint will be preferred, as nonparametric statistics are largely studied.

This chapter is organized as follows: Section 3.1 presents parametric methods where model is assumed and is adapted to the image only through the setting of a small set of parameters. Section 3.2 presents nonparametric methods which learn their model from image.

3.1 Penalizing functions and parametric statistics

3.1.1 Deterministic viewpoint: penalizing functions

An energy in a variational problem is defined as a (positive) penalizing function on a quantity which is desired to be close to zero at the solution of the variational problem (and equal to zero in the ideal case).

An example of penalizing functions is in optical flow (2.15) where the optical flow constraint is desired to be zero (brightness consistency assumption). Another example with the square function in segmentation is the Chan et Vese functional:

$$E^{CV}(\Gamma) = \int_{\Omega} (u(\mathrm{m}) - \mu(\Omega))^2 \,\mathrm{dm} + \int_{\overline{\Omega}} (u(\mathrm{m}) - \mu(\overline{\Omega}))^2 \,\mathrm{d}x + \nu|\Gamma| \qquad (3.3)$$

This energy aims at finding a partition of the image into two piecewise constant regions. Thus, it is expressed as a penalizing square function of the difference between a pixel feature and the mean feature over its region. An ideal case is an image composed of two connected piecewise constant regions. If Γ is the boundary between these two regions, the first two terms would be equal to zero (the third is a constraint to promote smooth boundaries).

In the three image processing problems considered in the latter chapter, the classical penalizing function is the square penalizer $\varphi(x) = x^2$ [HS81, CV01], essentially because it is strictly convex and differentiable.

In some cases, function $\varphi(x) = x^2$ is abandoned for two main reasons: first, used in a data term φ is not robust to outliers, this reason is quite intuitive: the square function computed on a single outlier will dramatically increase the global score of the energy. Second, used in a regularization term, for instance a penalizing function on optical flow variations, $\varphi(|\nabla \mathrm{v}|)$ will not respect image discontinuities and will blur contours. This reason is not straight forward to understand and we show an example of Fig. 3.1. We show a discrete 1-D example of a function $\theta = \mathrm{v}$ with an edge. The classical regularization term is $\sum \varphi(|\nabla \mathrm{v}|)$, if the φ function is chosen to be the square function, the score on this discontinuity of 4 would be 16, if the discontinuity is smoothed, the score would be 4. The square function will favor smooth contours. Replacing the square function by the absolute function, the discontinuity score would be 4 on both examples. The absolute function will not favor smooth contours or sharped contours. In the resulting diffusion equations, this behavior will be seen as isotropic diffusion (non edge preserving) versus anisotropic diffusion (edge preserving).

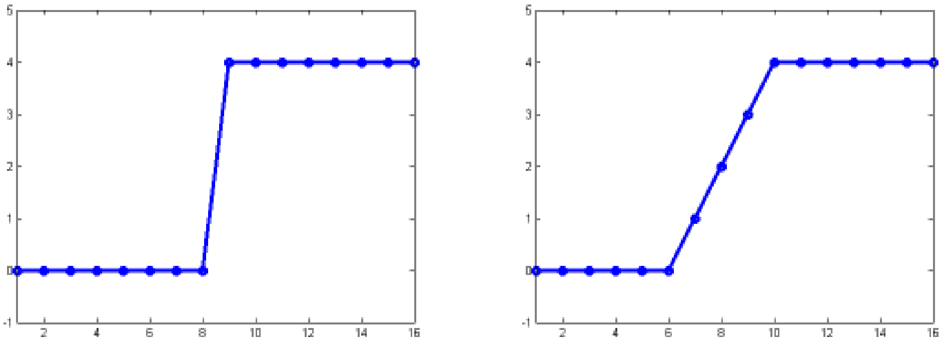


Figure 3.1: *Sharped or smoothed contours: a discrete 1-D example. Score of the function $\sum \varphi(|\nabla \mathrm{v}|)$ on discontinuity with the square function on sharped contour: 16 and on smooth contours: 4. The square function promotes smooth contours. Score of the absolute function on sharped contours: 4 and on smooth contours 4. The absolute function does not promote smooth or sharped contours.*

Other robust functions are shown Fig. 3.2, they ideally follow these properties: they must minimize effects of outliers, be differentiable and preserve discontinuities [BA96].

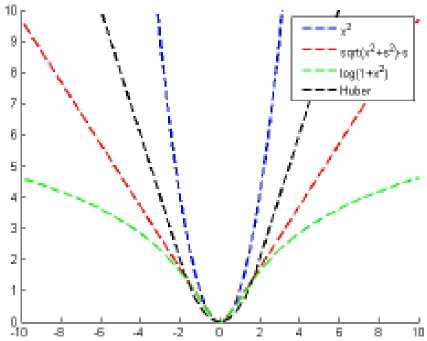


Figure 3.2: *Some robust functions used in the literature*

3.1.2 Statistical viewpoint: parametric distributions

Historically, Maximum likelihood links φ-functions and parametric distributions.

Let $\{u(\mathrm{m};\theta) \mid \mathrm{m} \in D\}$ be some values on the domain D of an image observed under some unknown θ. Let us assume they are independent observations of a random variable X_U following some parametric distribution $f_U(u) = a\ \exp(-|(u-\mu)/b|^\alpha)$, where μ is the mean of the distribution, where a is a normalization constant such as f_U integrates to 1, b is a scaling factor, and α is the shape of the distribution. Choosing $\alpha = 1$ is the Laplacian distribution, $\alpha = 2$ is the Gaussian distribution, ... The joint probability density function of these n independent random variables is

$$f(u_1, \dots, u_n; \theta) = \prod_{\mathrm{m}\in D} f_U(u(\mathrm{m};\theta)) \tag{3.4}$$

$$= a^n \prod_{\mathrm{m}\in D} \exp\Big(-|(u(\mathrm{m};\theta)-\mu)/b|^\alpha\Big), \tag{3.5}$$

The likelihood of a given observation U given θ writes:

$$L(\theta) = K \exp\left(-\sum_{\mathrm{m}\in D} \big|(u(\mathrm{m};\theta)-\mu)/b\big|^\alpha\Big)\right), \tag{3.6}$$

with some constant $K > 0$.

Maximizing the likelihood is then equivalent to maximizing the log likelihood as the log is a monotonous increasing function.

$$LL(\theta) = -\sum_{m\in D} \Big|(u(\mathrm{m};\theta)-\mu)/b\Big|^\alpha + \log K, \tag{3.7}$$

Maximizing the $\log$ likelihood based on a parametric distribution $exp(-\varphi)$ is then equivalent to minimizing a penalizing function φ (Gaussian distribution and square function for $\alpha = 2$, Laplacian distribution and absolute value for $\alpha = 1$).

Let us develop an example on the segmentation problem. As showed in Section (2.3), Bayesian segmentation problem writes

$$E^B(\Gamma) = -\int_\Omega \log(f_\Omega(u(\mathrm{m})))\ \mathrm{dm} - \int_{\overline{\Omega}} \log(f_{\overline{\Omega}}(u(\mathrm{m})))\ \mathrm{dm} + \nu|\Gamma| \tag{3.8}$$

Let us now assume that f follows a monovariate Gaussian parametric distribution

$$E^B(\Gamma) = \int_\Omega \frac{1}{2}\log(2\pi\sigma^2(\Omega)) + \Big(\frac{u(\mathrm{m}) - \mu(\Omega)}{\sqrt{2}\sigma(\Omega)}\Big)^2 \,\mathrm{dm} + \int_{\overline{\Omega}} \frac{1}{2}\log(2\pi\sigma^2(\overline{\Omega})) + \Big(\frac{u(\mathrm{m}) - \mu(\overline{\Omega})}{\sqrt{2}\sigma(\overline{\Omega})}\Big)^2 \,\mathrm{dm} + \nu|\Gamma| \tag{3.9}$$

Choosing a constant standard deviation $\sigma = \sqrt{0.5}$ this energy is equivalent to Chan et Vese (3.3) up to some constant. Actually, many deterministic methods using penalizing functions (*e.g.* absolute value) are equivalent to statistical methods with parametric statistics (*e.g.* Laplacian distribution). This equivalence also applies for nonlocal filters and nonparametric statistics.

3.2 Non local filters and statistical measures

We have presented methods where a model is chosen and fitted on a image through few parameter setting (parameters of the robust penalizing function or of the statistical distribution). Alternative methods let the image generate its own model. These methods can again be classified into deterministic and statistical view points.

3.2.1 Deterministic viewpoint: nonlocal filters

Neighboring filters have been introduced in [Yar85]. The idea is the following: let us consider a noisy observation, in order to find the original value of a pixel u_i, one can search all the close values in the image (disregarding their geographic location) and average over these values. Thus, the image generates its own model from non local self-similarities.

Later improvements constraint the search of these close values in a spatial neighborhood [SB97]. The resulting debluring filter is then a mean over values, close in value $u(\mathrm{m}) - u(\mathrm{n})$ and in position $\mathrm{m} - \mathrm{n}$.

$$SNF(u(\mathrm{m})) = \frac{1}{C(\mathrm{m})} \int_D u(\mathrm{n}) e^{-\frac{|\mathrm{m}-\mathrm{n}|}{\sigma^2}} e^{-\frac{|u(\mathrm{m})-u(\mathrm{n})|}{h^2}} \,\mathrm{dn} \tag{3.10}$$

where σ is a spatial filtering parameter, h is a feature filtering parameter, and D is the image domain.

The nonlocal means algorithm follows the same idea, although self-similarities are not computed between pixel values but between neighboring blocks [BCM05]

$$NL(u(\mathrm{m})) = \frac{1}{Z(\mathrm{m})} \int_D u(\mathrm{n}) e^{-\frac{|\mathrm{m}-\mathrm{n}|}{\sigma^2}} e^{-\frac{|u(\mathcal{N}(\mathrm{m}))-u(\mathcal{N}(\mathrm{n}))|}{h^2}} \,\mathrm{dn} \tag{3.11}$$

where $\mathcal{N}(\mathrm{m})$ is a spatial neighborhood around m, where $Z(\mathrm{m})$ is a normalizing constant.

3.2.2 Statistical viewpoint: non-parametric distributions

On the statistical viewpoint, the model does not follow a parametric distribution but again adapts to the data. A nonparametric distribution (*e.g.* histogram) is generated from the image and this distribution is used as model. Examples of nonparametric estimation of densities are numerous in image processing and some of them were presented along Chapter 2. In tracking the Mean-shift algorithm [CRM00] builds a nonparametric distribution over a joint location-color model of the object. In segmentation, nonparametric active contours [KFY$^+$05, HSD$^+$06] build color distributions of pixels inside and outside the active contour. In image denoising, the UINTA model [AW06] computes a distribution over image patches and averaged to perform denoising. The same model was also applied to segmentation [ATW06]. The UINTA model writes

$$u(\mathrm{m}) = \arg\min_{u(\mathrm{m})} H(u(\mathrm{m})|u(\mathcal{N}(\mathrm{m}))) \tag{3.12}$$

where H is a conditional entropy estimated non parametrically in the space of neighborhoods $\mathcal{N}(\mathrm{m})$.

Connections between deterministic and statistical viewpoints can again be made for instance in image denoising. The nonlocal denoising algorithm detailed in [BCM05] which learns image model through similarities of patches connects to the UINTA denoising algorithm which learns a distribution of image patches [AW06].

The statistical viewpoint will be chosen in this thesis, as it is somehow more general and extensively studied for image-generated models.

The critical problem is the construction of the image-generated model. An image offers a limited number of samples to efficiently estimate a model. Moreover, these image-generated models are often defined locally, decreasing the number of available samples, increasing sparsity, leading to poor model representation and performance drop. Let us present the model construction in the statistical viewpoint: nonparametric estimation.

CHAPTER 4

CLASSICAL NONPARAMETRIC ESTIMATION

The ”nonparametric” label is not directly related to the number of parameters to estimate a probability density function (PDF). In fact, in most of the ”nonparametric” PDF estimates, there are parameters and as we will see in this section, some of them are critical. Actually, the label ”nonparametric” means that the density estimate can asymptotically approximate any continuous density. Consequently, the estimated PDF cannot be described in terms of a small number of parameters, as opposed to, say, a Gaussian distribution defined by its mean and variance.

4.1 PDF estimation

Let a sample $U = \{u_1, \ldots, u_n\}$ be independent random observations of a random variable X_U taking values in a range of values Ω_U, The size of the sample is noted $|U|$. As the actual PDF f_U of the random variable X_U is unknown it is estimated empirically $\hat{f}_U$ from the sample U.

4.1.1 Histogram estimation of densities

The most common way to estimate densities nonparametrically in many domains including image processing is through histogram construction. The support of the distribution (estimated for instance between the min and the max of the points in the sample) is divided into cells C_h of size h also called bin width and of volume V_h. This construction is natural in image processing as the samples (usually considered as the luminance values at each pixels) are already sampled on a discrete grid. A natural choice to divide the space of possible values into cells is then the discrete grid of possible luminance values: usually $[1..255]$.

An histogram estimate of the density is constant within a cell and is given by the ratio between the number of samples falling into the cell divided to the total number of samples, relatively to the cell volume V_h.

$$\hat{f}_U(u) = \frac{k(C_h(u))}{|U|.V_h} \tag{4.1}$$

where $k(C_h(u))$ is the number of samples falling into the same cell $C_h(u)$ than u. Histogram construction has only one parameter, the bin width h. Albeit this estimator is non smooth at cell boundaries, it is labeled nonparametric as it converges to the actual continuous density f when the number of samples tends to infinity and the bin width h must tend to zero.

4.1.2 Kernel estimation of densities

In order to obtain smoother estimates of densities, a general class of density estimate called kernel density estimate was introduced by Rosenblatt [Ros56] and Parzen [Par62].

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} K_h\Big(u - u_i\Big) \tag{4.2}$$

where K_h is a kernel function of size h. This kernel function must follow conditions like be centered at 0, must integrate to 1, must have first moments vanishing followed by a non vanishing moment (called the order of the kernel). Upon the kernels satisfying these conditions, one can think of Epanechnikov, Gaussian $\mathcal{N}(0,1)$ or Uniform kernel $U(0,1)$. When using an Uniform kernel, the density estimate, firstly introduced in the literature [FH51], is intuitive: it approximates the density at sample point u with the relative number of samples $k(u)/|U|$ falling into the open ball of volume V_h centered on u (also called window or Parzen window).

$$\hat{f}_U(u) = \frac{k(u)}{|U|.V_h}. \tag{4.3}$$

where $|U|$ is the number of sample points in U. The main difference with histogram construction (4.1) is that the kernels are placed over the samples, whereas in the histogram construction, the kernels have fixed position over the centers of the cells.

h is the size of the window, also called bandwidth. Using a uniform kernel, h is basically the radius of the window, whereas using infinite support kernels such as Gaussians, h is the standard deviation of the kernel. The choice for this bandwidth h is critical. On one hand, it must be large enough for small sized samples in order that windows contain enough samples to accurately estimate the density. On the other hand, the bandwidth must be small enough to capture details of the PDF. Additionally, the bandwidth must depend on the sample size $|U|$. Indeed, as the density must approximate any continuous densities, the bandwidth must tend to zero as the sample size tends to infinity.

Several methods exist to estimate an optimal h in the sense of some criteria for example the square difference between the estimated distribution and the actual distribution (MISE), these optimal estimates will be presented later in Section 4.1.4.

4.1.3 The multivariate case

In the multivariate case, the samples are not scalars but vectors. The kernel density estimate $\hat{f}_U$ uses multivariate d-dimensional kernels K_H of bandwidth H (covariance matrix $d \times d$, in the Gaussian case).

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} K_H\Big(u - u_i\Big) \tag{4.4}$$

If the multivariate kernel is separable, it can be written as a product of univariate kernels

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} \prod_{j=1}^{d} K_{h_j}\Big(u(j) - u_i(j)\Big) \tag{4.5}$$

For instance, in the Gaussian kernel case :

$$\hat{f}_U(u) = \frac{1}{n(2\pi)^{d/2}|\Sigma|^{1/2}} \sum_{u_i \in U} \exp\Big[-\frac{1}{2}(u-u_i)^t \Sigma^{-1}(u-u_i)\Big] \tag{4.6}$$

Σ is a $d \times d$ covariance matrix.

4.1.4 Bandwidth selection

As mentioned before, the critical parameter is the bandwidth h or bin-size of the histogram. Both in practice and in theory, the bin width must tend to zero when the number of samples tends to infinity. Larger bandwidth will capture overall structure while smaller bandwidth will get finer structure. Many techniques can make the selection of this parameter automatic, one can assume the PDF to be Gaussian, and finds the optimal h in the sense of some error criteria like Integrated Mean Squarre Error (MISE) or Asymptotic Mean Square error (AMISE).

These automatic bandwidth selection are called plug-in rules:

- an histogram bandwidth selection rule is

$$\hat{h} = 3.5\hat{\sigma_U}|U|^{-1/3} \tag{4.7}$$

- a Parzen bandwidth selection rule is the rule-of-thumb [Sil86]

$$\hat{h} = 1.06\hat{\sigma_U}|U|^{-1/5}. \tag{4.8}$$

  As variance increases when the distribution is not unimodal a better estimate of the bandwidth is defined with the interquartile range $\hat{p}$

$$\hat{h} = 0.9 \min\left(\frac{\hat{p}}{1.34}, \hat{\sigma_U}\right)|U|^{-1/5} \tag{4.9}$$

  where $\hat{\sigma}_U$ is the empirical standard deviation of the samples U and $\hat{p}$ is the interquartile range (difference between the third quartile and the first quartile). The 1.34 constant is the interquartile range of a univariate normal kernel, standard deviation 1, it acts as a normalization weighting.

- The multivariate rule of thumb depends on the dimension and reads

$$\hat{h} = \hat{\sigma_U}|U|^{-\frac{1}{d+4}}. \tag{4.10}$$

One can note that plug-in rules (4.8) and (4.9) are contradictory, as the constants in front do not match. Actually, there are a lot of plug-in rules in the literature which give various results depending on the hypothesis on the actual distribution, on the kernel choice for the density estimate, on the error criterion, ... Loosely speaking, most of plug-in rules for unidimensional kernel estimates (4.8) and (4.9), have in common the decrease of the bandwidth with respect to the number of samples at a exponential rate of $-1/5$, the constant in front varies between 0.5 and 3

Another method for automatic bandwidth selection is the double kernel estimates, which estimates the density with two different kernels (for instance Gaussian and Epanechnikov) and tries to find the bandwidth which minimizes the distance between these two estimates. Other methods use cross validation.

Cross validation techniques or double kernel methods estimate the bandwidth h through an external minimization procedure. These methods will not be considered in numerical studies as they are more computationally intensive and not suitable for our image processing applications as the density estimates are often used in an iterative loop.

Let us now make an illustration of the importance of the bandwidth on Fig.4.1, in the univariate case, and Fig. 4.2, in the bivariate case. One can already observe that bandwidth selection problem seems to get worse in high dimensions.

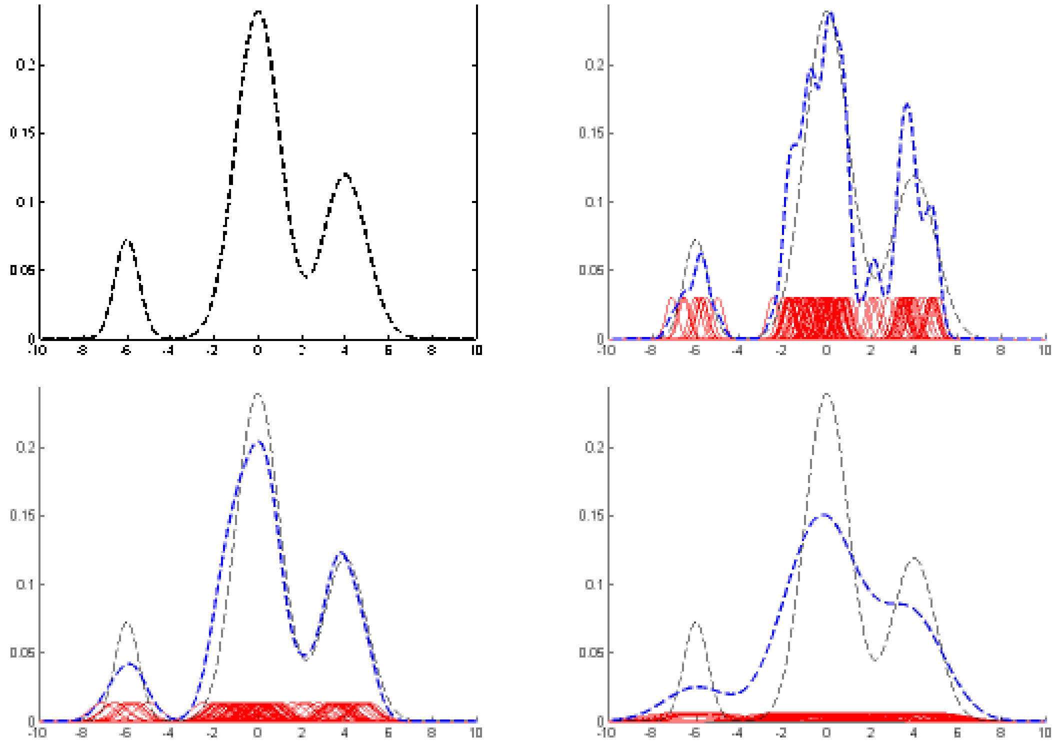


Figure 4.1: *Kernel density estimation performance on a 1-D Gaussian mixtures for different bandwidth. Actual distribution is in black, kernel density estimate is in blue and kernels are in red. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with $0.2h$, PDF estimated with h, PDF estimated with $5h$.*

The PDFs being estimated, some statistical measures can be evaluated on these PDFs, and particularly entropy-based measures.

4.2 Entropy-based measures

4.2.1 Definition and estimation of entropy

Entropy is a concept larger than information theory and statistics. In information theory, it has been introduced by Shannon as a code length in bits necessary to encode a variable. In general, entropy measures the uncertainty in the values of random observations. Let X_U be a random variable taking values in a range of values Ω_U with probability density function f_U. Shannon entropy H is defined as follows: let us consider a discrete random variable, for instance we consider this discrete random variable as a quantization of a continuous random

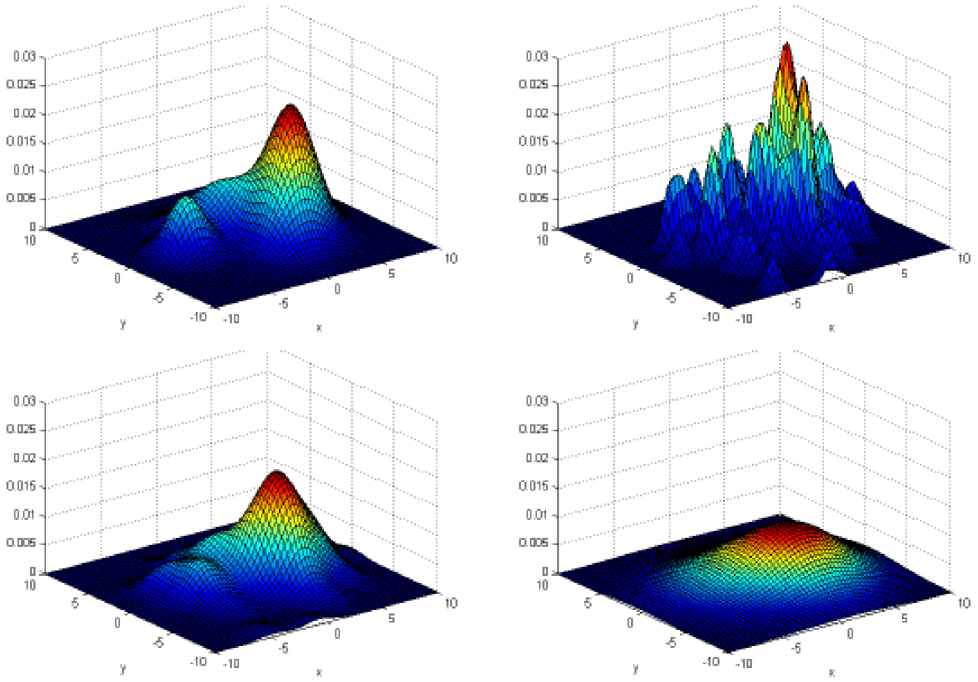


Figure 4.2: *Multivariate kernel density estimation performance on a 2-D Gaussian mixtures for different bandwidth. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with 0.2h, PDF estimated with h, PDF estimated with* $5h$

variable quantized with a step equal to Δ. In the discrete case, p_U is the probability mass function. Shannon entropy writes

$$H^{\Delta}(X_U) = -\sum_{i \in \Omega_U} p_U(i) \log p_U(i) \tag{4.11}$$

where $p_U(i)$ is the probability mass function of a value. It is expressed as the number of occurrence $k(i)$ of i in $U : p_U(i) = k(i)/|U|$.

Shannon entropy has been extended to continuous random variables with the definition of differential entropy. The differential entropy of the random variable X_U is defined by

$$H(X_U) = -\int_{\Omega_U} f_U(u) \log f_U(u) \mathrm{d}u \tag{4.12}$$

measured in nats, if $\log_2$ is used instead of $\log$, the unit is the bit.

According to [CT91], the link between the differential entropy and discrete entropy is

$$\begin{aligned} H^{\Delta}(X_U) &= -\sum_{i \in \Omega_U} p_U(i) \log p_U(i) \\ &= -\sum_{i \in \Omega_U} f_U(i).\Delta \log f_U(i).\Delta \end{aligned}$$

Since $\sum f_U(i)\Delta$ approximates $\int f_U(i)$ as $\Delta \to 0$, they deduced

$$H^{\Delta}(X_U) + \log \Delta \to H(X_u), \text{as } \Delta \to 0 \tag{4.13}$$

$\log \Delta$ is equal to the number of bits required to code the resulting discrete variable, *e.g.* if the quantization step is $\Delta = 1/256$, $-log_2\Delta = 8bits$ is the number of bits necessary to code 256 symbols.

Estimation of entropy on a finite sample U of a random variable X_U: u_1,u_2,...,u_n can be performed with different estimates

- Integral estimate:
  The integral estimate of entropy consists in reducing the support of Ω_U to non zero values and perform numerical integration.

- Resubstitution estimate:
  Another estimate proposes to write entropy H as the expected value of the log density function and to empirically estimate this expectation.

$$\begin{aligned} H(X_U) &= -E_U[\log f_U] && (4.14) \\ &= -\frac{1}{|U|}\sum_{u_i \in U} \log f_U(u_i) && (4.15) \end{aligned}$$

  Ahmad-Lin [AL76] used this estimate of entropy where f is estimated by kernel density estimate.

- Other estimates of entropy:
  Another estimate is the splitting data estimate: the sample is splitted into two subsamples, one is used to construct a kernel density estimate and the second is used for empirical estimation of H as in the resubstitution estimates. This kind of estimate has also been studied in image processing applications [VW97].

4.2.2 Estimation of entropy-based measures

Entropy estimation being presented one can use the estimator presented above for measures derived from entropy. For instance using cross entropy: Kullback-Leibler divergence $D_{\mathrm{KL}}(X_T, X_R)$ or using joint entropy: Mutual information $D_{\mathrm{MI}}(X_T, X_R)$.

Let us consider two random variables X_R, resp. X_T, following a distribution f_R, resp. f_T. Let us consider that both variables X_R and X_T have the same support Ω. The Kullback-Leibler divergence is the information gain.

$$\begin{aligned} D_{\mathrm{KL}}(X_T, X_R) &= \int_\Omega f_T(u)\ \log \frac{f_T(u)}{f_R(u)}\ \mathrm{d}u \\ &= -H(X_T) + H_\times(X_T, X_R) && (4.16) \end{aligned}$$

where $H_\times(X_T, X_R)$ is the cross entropy between two random variables X_T and X_R.

The mutual information is defined as

$$\begin{aligned} D_{\mathrm{MI}}(X_T, X_R) &= \int_\Omega f_{T,R}(u)\ \log \frac{f_{T,R}(u)}{f_R(u) f_T(u)}\ \mathrm{d}u \\ &= H(X_T) + H(X_R) - H(X_T, X_R) && (4.17) \end{aligned}$$

where $H(X_T, X_R)$ is the joint entropy between two random variables X_T and X_R.

4.3 Conclusion

In this chapter, we have presented classical nonparametric estimation. The critical parameter in kernel density estimation is the bandwidth or standard deviation of the kernels. In particular, this deviation is hard to select as it should be large enough to take into account sparsity of the data but small enough to capture peaks of distribution. Things get worse in more than one dimension as sparsity of the data increases. This problem is informally called the curse of dimensionality. As the dimension of the data space increases, the space sampling gets sparser. Therefore, fewer samples fall into the Parzen windows centered on each sample, making the PDF estimation less reliable. Dilating the Parzen window does not solve this problem since it leads to over-smoothing the PDF. In a way, the limitations of the Parzen method come from the fixed window size: the method cannot adapt to the local sample density. This problem will lead to poor estimation of PDFs and to poor estimation of statistical measures estimated from these PDFs.

Part II

High-dimensions in variational problems: a new framework

CHAPTER 5

HIGH-DIMENSIONS AND NONPARAMETRIC STATISTICAL FRAMEWORK

While high dimensionality is a curse for statisticians, recent methods in the three variational problems defined in Chapter 2 often pointed higher dimensional data as a major way of improvements. These higher dimensional data can be for instance color channels, but also other information such as geometric layout or image gradients. As a reminder, a (statistical) variational formulation of a problem in image processing writes

$$\hat{\theta} = \arg\min_{\theta} E(U, f_U, \theta) \tag{5.1}$$

The data values extracted from the images U are vector valued and its associated PDF f_U is high dimensional.

Due to nonparametric estimation problems in high dimensions mentioned in last chapter, high-dimensionality is often circumvented by assuming data are parametric distributed, or treated non parametrically by assuming independence between features, or by choosing a large Parzen window size to overcome sparsity of the data thus oversmoothing the PDFs. Reduction dimension techniques, *i.e.* projecting high dimensional spaces on lower dimensional spaces, is not satisfactory either. In fact, these techniques even if decreasing the number of dimensions, do not reduce dimensions to 1 (depending of the algorithm of reduction and of the data).

Throughout the three variational presented in Chapter 2, a nonparametric energy able to deal with low to higher dimensional space without making assumptions on the underlying PDF is of great interest. Such an energy should satisfy various properties in order to be used in practice in variational problems. In object tracking, a high dimensional nonparametric energy without knowledge on the underlying PDF and non-differentiable is acceptable as these steps can be avoided with gradient-free optimization algorithms (diamond search [ZM00], particle filter [PHVG02]).

However, in some variational formulations such as segmentation, an explicit knowledge of the PDF is necessary to compute the gradient (the active contour evolves by computing a probability on sample points on the contour). Moreover, in some other variational formulations such as optical flow, the number of unknowns encourages to use a gradient-based resolution and thus to use a differentiable statistical estimate.

Summarizing, a statistical framework proposed for variational problems must

1. be efficient to estimate nonparametric energies $E(U, f_U, \theta)$ in high dimensions with few and sparse data, in order to estimate energies in variational problems
2. have a PDF interpretation f_U, in particular for applications in segmentation
3. be almost everywhere differentiable and its gradient $\nabla_\theta E(U, f_U, \theta)$ can be approximated, in particular for applications in optical flow.

Last but not least is computational efficiency as it is to be used inside numerical schemes (with possible high number of iterations).

5.1 Methodological contributions of this thesis

The methodological contributions rely on new general framework, called k-th nearest neighbor (kNN) framework, to solve variational problems defined with information theory on high dimensional data.

This thesis propose a new class of entropy-based measure estimates and their connections with locally density functions estimates as well as approximations of their gradients. We exhibit links between some well-known estimates (kNN estimate of PDF, kNN Mean-shift) and new ones (kNN estimate of entropy) in the scope of applications in variational problems. These estimates were proposed in very different contexts and applications and a unification was, to the best of our knownledge, never proposed in the literature.

1. The kNN estimate of entropy was introduced in the statistical community by [KL87, GLMI05, LPS05] and, is to the best of our knowledge, our contribution in the image processing community [BWD$^+$06, BDB07]. Other closely related kNN estimates of entropy were also introduced [CH04].
2. The kNN Mean-shift was introduced in the literature by [FH75] and in the image processing community by [GSM03, Com03]. However, it was used as a clustering algorithm and not as a general tool to differentiate entropy-based energies.
3. The kNN estimate of PDF was introduced in the statistical community by [LQ65] and in the image processing community by [Com03, MP04, TPJ05]. This PDF estimate will establish the link between the different kNN estimates.

This thesis links all these estimates as a general kNN framework to solve variational problems.

These estimates are adapted from low to high dimensions and for variational problems as it can estimate energies as well as their gradient. An example of performance of this estimate w.r.t the number of samples and dimension of the data is shown on Fig. 5.1, a more complete study is in Appendix A. This good behavior in high dimensions will allow us to generalize variational problems to high dimensional spaces, in particular image spaces better adapted to capture natural image statistics.

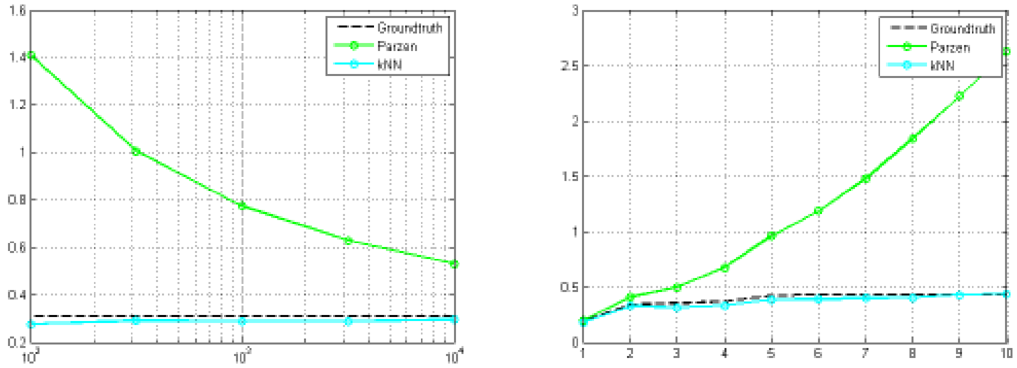


Figure 5.1: *kNN framework performance example: comparison of Kullback-Leibler divergence estimation between Gaussians using Parzen and kNN. Theoretical value: dashed black line; Parzen-based estimation: green line; kNN-based estimation: light blue line. (in lexicographical order) Fixed sample size (1000) and varying dimension; Fixed dimension (5) and varying sample size.*

Last but not least, as these estimates are implemented on graphic cards (GPU), it is computationally efficient. This point is important for variational problems as the resolution often goes through iterative schemes which required a lot of energy and gradient computations.

Finally, this thesis proposes the following methodological contributions for variational problems

- the definition of variational problems based on information theory,
- the generalization to high dimensional data in order to capture natural image statistics,
- an efficient statistical and variational framework to deal with these measures between high dimensional data

Section 5.2 presents the generalization to high dimensional feature space and its utility in image processing, classical nonparametric estimation , presented in Chapter 4, suffers from high dimensions from its fixed size bandwidth. Let us now present in Section 5.3 some locally adaptive PDF estimates and how they connect to a new class of efficient statistical measures estimates in Section 5.4, as well as approximations of their gradients in Section 5.5.

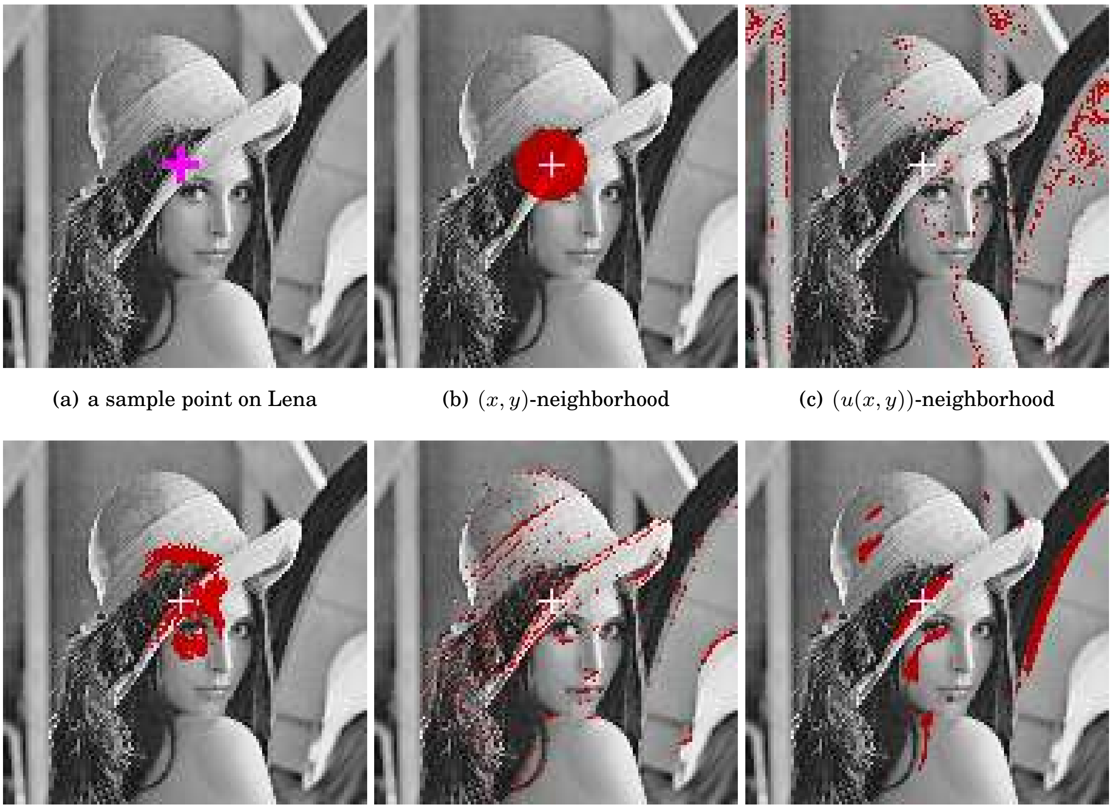
5.2 New possibilities with high dimensions

High dimensionality in nonparametric estimation starts from dimensions 3 or 4 as classical nonparametric estimate performances drop [TS92]. High dimensionality with this meaning thus starts with RGB color channels, however higher dimensions are also considered.

High dimensions in image processing are present by nature or by necessity, by nature as the images acquired from the acquisition process are often vector-valued and can be high dimensional: satellite imaging, medical imaging, for instance diffusion tensor imaging or high angular resolution diffusion imaging. By nature also because the similarity measure can introduce high dimensions such as mutual information or geometric constraints. By necessity as it is sometimes required to add cue information in the data such as spatial filters structure tensor [RBD03], gabor filters [PD02b], wavelets coefficients, patches of images neighborhoods, or temporal filters such as motion[BRDW03, BP07]. Indeed adding dimensions was pointed as a challenge and a major way of improvements in the literature of the three mentioned variational problems: tracking [EDD03, BP07], optical flow [BBPW04, BWS05, RB05] and segmentation [RBD03, BRDW03] but also in other problems such as denoising [BCM05, AW06].

In general, high dimensions are required to better characterize a sample point (pixel) on an image. A pixel can be characterized as a luminance value, a position on the image, a texture (considering a patch surrounding the point).

As an example let us consider an image, a pixel located at position (x, y) in the image has the feature value $u(x, y)$, for example luminance or RGB color values. An illustration on the 500 nearest neighbors of a sample point on image "Lena" on Fig. 5.2(a) for various spaces is presented. One canonical choice of space is the coordinates of the images x and y, thus the nearest neighbors in this space would be the spatial neighbors Fig. 5.2(b). Another choice will consider similar pixels with very different feature values u, disregarding its spatial location Fig. 5.2(c). A compromise is to consider the higher dimensional space of joint feature and spatial neighbors $(x, y, u(x, y))$ Fig. 5.2(d) with some weighting between both.



(a) a sample point on Lena

(b) (x, y)-neighborhood

(c) $(u(x, y))$-neighborhood

(d) $(x, y, u(x, y))$-neighborhood

(e) $(x, y, u(x, y), \nabla u(x, y))$-neighborhood

(f) $(u(x, y), u(x{+}1, y), u(x, y{+}1), ..., x, y)$ 9x9 patch-neighborhood

Figure 5.2: *Various neighborhoods types on Lena*

One can validate this idea of high-dimensional space with its asymptotic behavior. Let us consider the weighting α between spatial information and feature value $(\alpha x, \alpha y, u(x, y))$. As α tends to infinity the nearest neighbors of $(\alpha x, \alpha y, u(x, y))$ are the spatial neighbors of (x, y)

Fig. 5.2(b). As α tends to 0, the spatial features have no influence and the space is the feature space Fig. 5.2(c). A reasonable choice is $\alpha = 1$, a compromise between these two classical spaces Fig. 5.2(d). One other possibility to better describe the sample is to add a gradient information $(x, y, u(x, y), \nabla u(x, y))$ Fig. 5.2(e). Pixels in the neighborhood will have a gradient in the same directions than the sample point pixel (towards the bottom right of the picture).

This high dimensional space also captures local patterns and structure and it models spatial-features interactions. This idea also follows the same behavior as nonlocal filters, presented in Section 3.2, where self-similarities are searched in feature $|u(\mathrm{n}) - u(\mathrm{m})|$ and spatial $|\mathrm{n} - \mathrm{m}|$ spaces

$$SNF(\mathrm{m}) = \frac{1}{C(\mathrm{m})} \int_D u(\mathrm{n}) e^{-\frac{|\mathrm{n}-\mathrm{m}|}{\sigma^2}} e^{-\frac{|u(\mathrm{n})-u(\mathrm{m})|}{h^2}} \,\mathrm{dn} \tag{5.2}$$

An extension would be naturally to extend this idea to neighboring patches neighboring patches with spatial constraints Fig. 5.2(f) as performed in image denoising [BCM05]. These kinds of spaces have not been studied in this thesis and are described in the perspectives Chapter 10 of this thesis.

The utility of high dimensions in image processing being motivated, let us now develop on nonparametric estimation dedicated to high dimensions.

5.3 Probability density function estimates

Regarding the Parzen estimate, the fixed bandwidth has the default to oversmooth peaks and undersmooth tails of distributions. This problem gets worse in high-dimensions as large bandwidth must be chosen to properly estimate PDF tails (present in majority in high dimensions), and small bandwidth must be chosen to properly estimate PDF peaks (rare but very informative). Nor of these solutions are satisfactory as large bandwidth will oversmooth PDF peaks while small bandwidth will undersmooth PDF tails. In order to overcome this problem, we present two classes of locally adaptive PDF estimates: balloon estimate and sample point estimate.

Let us remind the classical multivariate Parzen PDF estimate

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} K_H(u - u_i) \tag{5.3}$$

where K is a kernel function of bandwidth H. As a reminder of last Chapter 4, K can be a multivariate Gaussian and H is its $d \times d$ covariance matrix.

Illustrations on fig 5.3 shows the fixed bandwidth Parzen estimate and two locally adaptive estimates. The balloon estimate will be detailed in Section 5.3.1 and the sample point estimate will be detailed in Section 5.3.2.

5.3.1 Balloon estimate

The balloon estimate can be written in a similar form by replacing the fixed size kernel bandwidth H by a varying kernel bandwidth $H(u)$ (illustrated on Fig. 5.3(b), estimation point u is a dashed line).

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} K_{H(u)}(u - u_i) \tag{5.4}$$

where K is a kernel which must follow the same properties as the kernel in classical density estimate. Looking at point u, this estimator still places a fixed-size kernel $H(u)$ on each sample point u_i, but size of the kernels placed on each sample points changes with u.

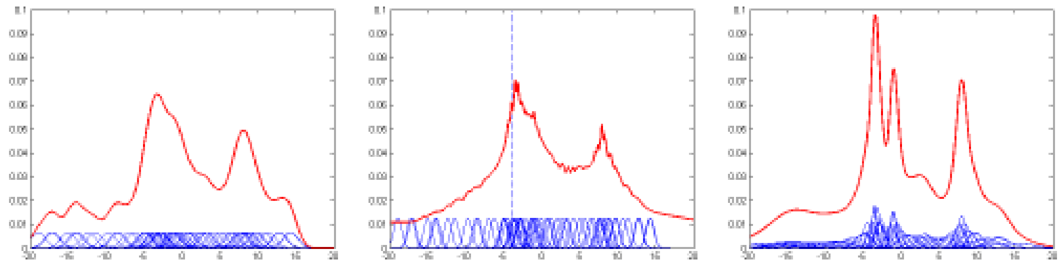
One can choose for K a uniform kernel $U(0,1)$ on the unit sphere. The kernel size variability is driven by $H(u) = \rho_k(U,u).I_d$, where $\rho_k(U,u)$ be the distance from u to its k-th nearest neighbor (kNN) among the data set U, and I_d is the $d \times d$ identity matrix. This estimate was early presented in [LQ65]:

$$\hat{f}_U(u) = \frac{k}{|U|.V_k} = \frac{k}{|U|.v_d\rho_k^d(U,u)} \tag{5.5}$$

where V_k is the ball volume of radius the distance to the k-th nearest neighbor $\rho_k(U,u)$ equal to $v_d.\rho_k^d(U,u)$: the volume v_d of the unit ball in $\mathbb{R}^d$ times the distance to the k-th nearest neighbor powered to dimension d.

An intuitive comparison between balloon and Parzen estimates can be made. In the Parzen estimate (4.3), the density of U at sample u is related to the number of samples falling into a window of fixed size h centered on the sample. The kNN balloon estimate (5.5) is the *dual* approach: the density is related to the size of the window h necessary to include its k nearest neighbors. The bandwidth "inflates" until it contains k points in the window, gaining the name of balloon estimate.

Although the distance is usually computed in the Euclidean sense, other distances can be used. Let us remind that the distances are between points $\mathbb{R}^d$, where d is the dimension of



(a) Fixed (b) Balloon (c) Sample-point

Figure 5.3: *Comparison of kernel-based PDF estimates: fixed-kernels and two local adaptive kernels balloon kernels and sample point kernels*

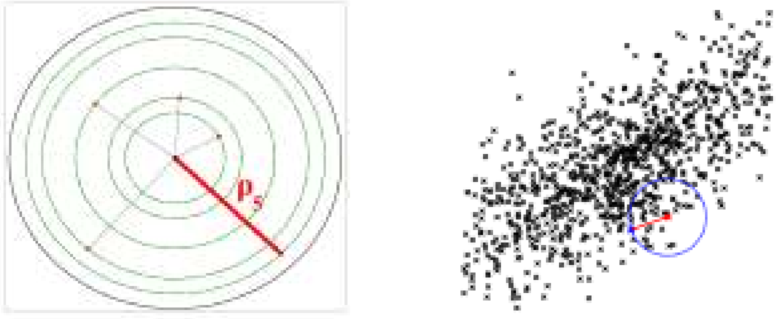


Figure 5.4: *The size of the kernel is equal to the distance to ρ_k the k-th nearest neighbor. Left: k=5, Right: illustration on a point cloud, estimation of the probability with k=50.*

the data.

One of the main advantages of balloon estimates, a part from its locally adaptive property, is that the choice of k is much less critical than the choice of the window size in the Parzen estimate. Actually, when the balloon approach is used for parameter estimation, k must be larger than the number of parameters and such that $k/|U|$ tends towards zero when both k and $|U|$ tend toward infinity. A typical choice is $k = \sqrt{|U|}$.

However, this estimator does not integrate to one, in particular in one dimension, outside of the support defined by sample U, it has a decreasing speed of $1/\mathrm{x}$ which yields high order bias in the tails. However, it behaves better in high-dimension, typically for $d > 3$ [TS92].

A second method to introduce local adaptivity in the kernel density estimate is the sample point estimate

5.3.2 Sample-point estimate

This estimator can also be written as a kernel density estimate (5.3), where the local adaptivity depends on the sample point $H(u_i)$ (illustrated on Fig. 5.3(c))

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in U} K_{H(u_i)}(u - u_i). \tag{5.6}$$

This estimator still places a kernel on each sample point, but the variable size of the kernels placed on each sample points, but fixed for any estimate point u, gaining the name of sample point estimate. For comparison, the balloon estimate at point u uses the same kernel on all the sample points u_i, but this unique kernel size is changing with u.

A typical choice for choosing the bandwidth variability is $H(u_i) \propto f(u_i)^{-1/d} I_d$. Choosing the Loftsgaarden estimate [LQ65] to estimate f, this choice is equivalent to take $H(u_i) = \rho_k(U, u_i)$ the distance to the k-th nearest neighbor. Other works suggest to take the Abramson square rule [Abr82], $H(u_i) = h.f(u_i)^{-1/2}$, regardless of the dimension d.

In a simple form, choosing for kernel the uniform density and for $H(u_i)$ the distance to the k-th nearest neighbor to u_i, the kNN sample point expression writes

$$\hat{f}_U(u) = \frac{1}{|U|} \sum_{u_i \in \mathcal{N}_{SP}(u)} \frac{1}{v_d \; \rho_k^d(U, u_i)}. \tag{5.7}$$

where $\mathcal{N}_{SP}(u) = \{u_i \in U : \; \|u - u_i\| \leq \rho_k(U, u_i)\}$ is the kNN sample point window: the set of points which allow u in their k first nearest neighbors.

If we compare the two density estimates the main advantage of balloon over sample point estimate is that it has the simplest PDF expression based on kNN (5.5), it is never equal to zero and it is efficient in high dimensions [TS92]. The main drawback is that it does not integrate to one, thus can lead to bias in the statistical measure estimates. However, unbiased expression of these measures will be derived in the next section.

5.4 Statistical measure estimates

These estimates of density are well known and currently used in statistics and in image processing [Com03, TPJ05]. However, we propose to use the balloon estimate in its simplest kNN form (5.5), to define a class of efficient entropy-based measures estimates.

5.4.1 Estimation of entropy

The Ahmad-Lin estimate [AL76] is an empirical estimate of entropy where the distribution is estimated non-parametrically using kernel density estimation.

$$\hat{H}_{\mathrm{AL}}(X_U) = -E_U[\log \hat{f}_U(u)] = -\frac{1}{|U|}\sum_{u_i \in U} \log \hat{f}_U(u_i) \tag{5.8}$$

where $\hat{f}_U$ is estimated through kernel density estimation (5.3) of actual PDF f_U. Approximation (5.8) converges in mean to the differential entropy of U. Note that an entropy estimate following the same fashion has later been proposed in [VW97]. However, this estimate proposed to split the data points, one half pour the PDF estimate, one half for the empirical entropy estimate. This Ahmad-Lin estimate allows to estimate entropy and the PDF with all the samples available.

Plugging the kNN form of balloon estimate (*i.e.* with uniform kernels (5.5), [LQ65]), in the Ahmad-Lin estimate (5.8), we have

$$\begin{aligned}
H(X_U) &= -\int_{\Omega_U} f_U(u) \log f_U(u) \mathrm{d}u \\
&= -E_U\left[\log f_U(u)\right] \\
\hat{H}_{\mathrm{AL}}(X_U) &= -\frac{1}{|U|}\sum_{u_i \in U} \log \hat{f}_U(u_i) \\
\hat{H}_{\mathrm{kNN-Biased}}(X_U) &\overset{\mathrm{kNN}}{=} -\frac{1}{|U|}\sum_{u_i \in U} \log \frac{k}{|U| v_d \rho_k(U, u_i)^d} \\
&= \log(v_d|U|) - \log(k) + d\ \mu_U(\log \rho_k(U))
\end{aligned} \tag{5.9}$$

where $\mu_U(a)$ is the mean of a for all the values it takes over the sample set U

$$\mu_U(a(u)) = \frac{1}{|U|}\sum_{u_i \in U} a(u_i). \tag{5.10}$$

Informally, the main term in estimate (5.9) is equal to the mean of the log-distances to the k-th nearest neighbor of each sample (Fig. 5.5).

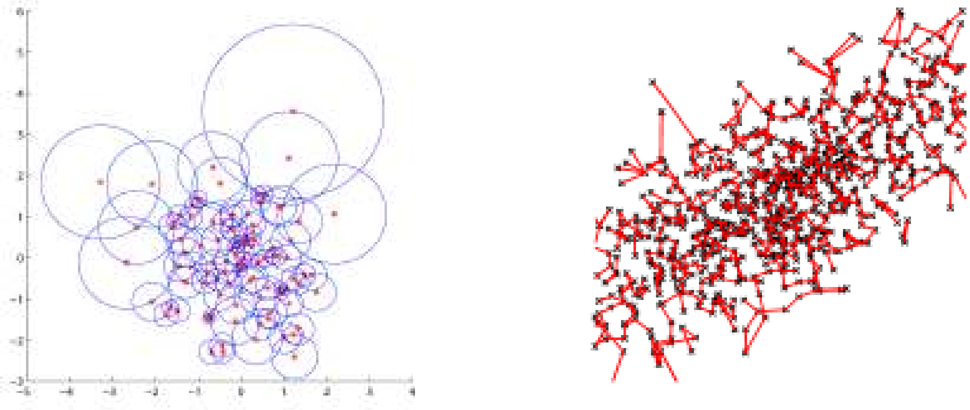


Figure 5.5: *Entropy is a function of the mean of* log*-distances to the* k*-th nearest neighbor. Left: k=2, Right: illustration on a point cloud k=3.*

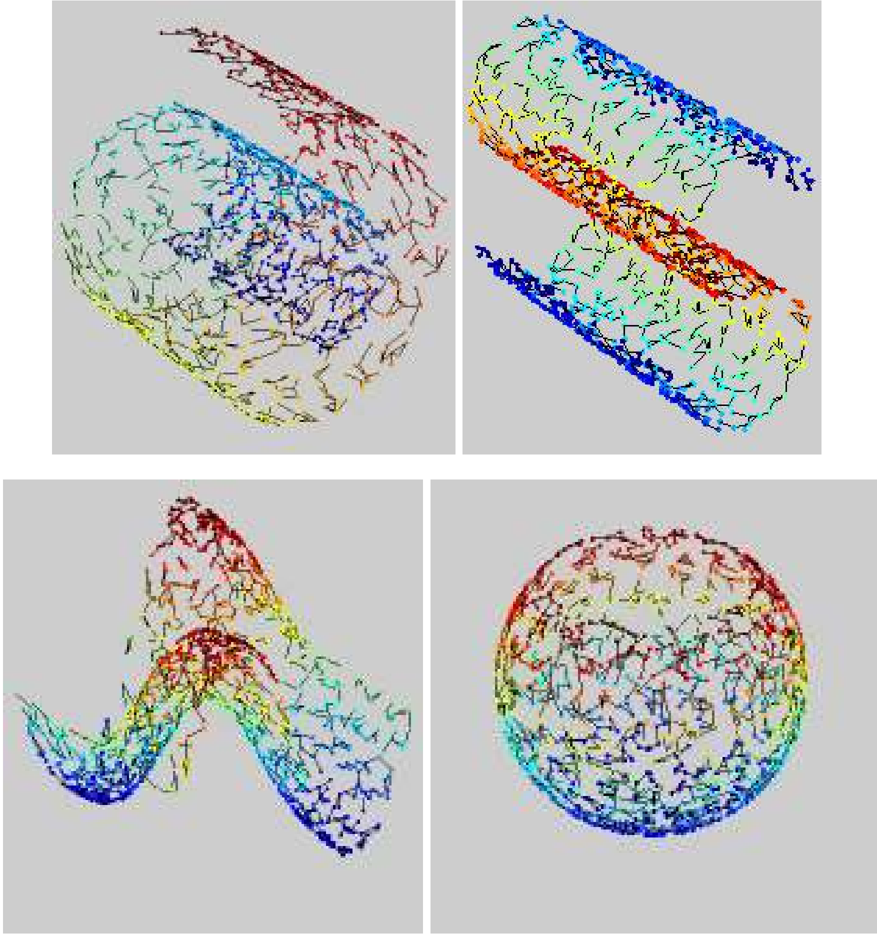


Figure 5.6: *Entropy is a function of the length of the k-nearest neighbor graph (k=3 on this example, each point is connected to its 3-nearest neighbor), this graph follows the underlying manifold as shown on some examples*

Examples of k-nearest neighbors graphs are showed on some point clouds (Fig. 5.6). Visually, the graphs appear to follow the underlying manifold.

Moreover, one can note that expression (5.9) does not estimate explicitly the PDF f_U, entropy is estimated directly from the data. This estimator, relying on an efficient locally adaptive density function estimate is thus well adapted for high-dimensions.

However, this estimate of entropy is biased, the reason being that the balloon estimate does not integrate to 1. Yet, we connect it with the Leonenko estimate of entropy [KL87, GLMI05, PLS07]. They defined a consistent and unbiased entropy estimator. First proposed for $k = 1$ [KL87]. Their work was extended to $k > 1$ with a proof of consistency under weak conditions on the underlying PDF [GLMI05]:

$$\hat{H}_{\mathrm{kNN}}(X_U) = \log(v_d\ (|U| - 1)) - \psi(k) + d\ \mu_U(\log \rho_k(U, u)) \tag{5.11}$$

where ψ is the digamma function Γ'/Γ. One can note the bias correction term $\psi(k)$ replacing $\log k$ and that $\psi(k)$ tends to $\log k$ when k tends to infinity. Another difference is the $\log(|U|-1)$ instead of $\log |U|$. It comes from, whether or not, the k-th nearest neighbor of a sample point includes or excludes the sample point itself. The bias problem of (5.9) is then only present for low number of samples $|U|$ (as typically $k = \sqrt{|U|}$) and vanishes when the number of samples grows.

The estimate (5.11) will show very good performance in both low and high dimensional spaces, on Gaussian-distributed data (see Appendix A) and on natural image data, and will be the entropy estimate used in this manuscript.

5.4.2 Estimation of entropy-based measures

This type of differential entropy estimate can be extended to any entropy-based measures: Kullback-Leibler divergence, Mutual information, ...

kNN estimation of Kullback-Leibler divergence

In a similar way, an estimate of Kullback-Leibler divergence using a kNN balloon estimation of density can be written as

$$\begin{aligned}
\hat{\mathfrak{D}}_{\mathrm{KL}}(X_T \| X_R) &= \int_{\Omega_T} f_T(t) \log \frac{f_T(t)}{f_R(t)} \mathrm{d}t \\
&= E_T \left[\frac{\log f_T(t)}{\log f_R(t)} \right] \\
&= \frac{1}{|T|} \sum_{t_i \in T} \log \hat{f}_T(t_i) - \log \hat{f}_R(t_i) \\
&\overset{\mathrm{kNN}}{=} \frac{1}{|T|} \sum_{t_i \in T} \log \frac{k}{|T| v_d \rho_k(T, t_i)^d} - \log \frac{k}{|R| v_d \rho_k(R, t_i)^d} \\
&= \log \frac{|R|}{|T|} + d\, \mu_T(\log \rho_k(R, t)) \\
&\quad -d\, \mu_T(\log \rho_k(T, t)). \qquad (5.12)
\end{aligned}$$

This Kullback-Leibler divergence estimator can be also be retrieved by decomposing it into a sum of two entropies, the classical differential entropy and the cross entropy:

$$\begin{aligned}
\hat{\mathfrak{D}}_{\mathrm{KL}}(X_T \| X_R) &= \int_{\Omega_T \cup \Omega_R} f_T(t) \log \frac{f_T(t)}{f_R(t)} \mathrm{d}t \\
&= \int_{\Omega_T} f_T(t) \log f_T(t) \mathrm{d}t - \int_{\Omega_T \cup \Omega_R} f_T(t) \log f_R(t) \mathrm{d}t \\
&= -H(X_T) + H^{\times}(X_T, X_R) \qquad (5.13)
\end{aligned}$$

kNN estimation of cross entropy

A similar expression can be then retrieved using the Leonenko estimate of entropy (5.11). Cross entropy (also called relative entropy or likelihood) of two sample sets R and T can be estimated by

$$\hat{H}^{\times}_{\mathrm{kNN}}(X_T, X_R) = \log(v_d |R|) - \psi(k) + d\, \mu_T(\log \rho_k(R)). \qquad (5.14)$$

Note again that estimate (5.14) does not depend on any PDF and that its main term is the mean of the log-distances to the k-th nearest neighbor among the samples of R of each sample of T. Since a sample t_i of T does not belong to R, the search for the k-th nearest neighbor *excluding* t_i itself does not in fact exclude any sample of R. This is why $|R|$ appears in (5.14) whereas $|T| - 1$ appears in (5.11).

The expression of Kullback-Leibler can be retreived in (5.12).

$$\begin{aligned}\hat{\mathfrak{D}}_{\mathrm{KL}}(X_T \| X_R) &= H^{\times}(X_T, X_R) - H(X_T) && (5.15)\\ &\overset{\mathrm{kNN}}{=} \log \frac{|R|}{|T|-1} + d\ \mu_T(\log \rho_k(R,t)) \\ &\quad -d\ \mu_T(\log \rho_k(T,t)). && (5.16)\end{aligned}$$

It has also been proven that this estimator is consistent and asymptotically unbiased [PLS07]. The classical estimate of Kullback-Leibler divergence has many numerical instabilities arising from the division by 0 or log of 0, for instance when the two PDFs have different supports. Implicitly relying on a balloon estimate of PDF, these numerical instabilities are not present in this kNN estimate as the PDFs are never equal to zero but are vanishing at a decreasing speed of $1/$x.

This estimate will also show great performance in both low and high dimensional spaces, on Gaussian-distributed data (see Appendix A) and on natural image data, and will be the Kullback-Leibler estimate used in this thesis. Finally, one could derive other entropy-based measures, but this will not be used in this manuscript and can be pointed as perspectives.

kNN estimation of Jensen-Shannon divergence

Kullback-Leibler divergence is not symmetric, an alternative is the Jensen-Shannon divergence: let $M = T + R$ be a mixture of the two distributions, Jensen-Shannon divergence writes:

$$\begin{aligned}\mathfrak{D}_{\mathrm{JS}}(X_T \| X_R) &= \frac{1}{2}\mathfrak{D}_{\mathrm{KL}}(T \| M) + \frac{1}{2}\mathfrak{D}_{\mathrm{KL}}(R \| M)\\ &\overset{\mathrm{kNN}}{=} \frac{1}{2}(\log \frac{(|M|-1)(|M|-1)}{(|T|-1)(|R|-1)} + d\ \mu_T(\log \rho_k(M,t)) + d\ \mu_R(\log \rho_k(M,t))\\ &- \quad d\ \mu_T(\log \rho_k(T,t)) - d\ \mu_R(\log \rho_k(R,r)))\\ &\overset{\mathrm{kNN}}{=} \frac{1}{2}(\log C + d\ \mu_M(\log \rho_k(M,m)) - d\ \mu_T(\log \rho_k(T,t)) - d\ \mu_R(\log \rho_k(R,r))). && (5.17)\end{aligned}$$

where $|M| = |R| + |T|$ and $C = \frac{(|M|-1)(|M|-1)}{(|T|-1)(|R|-1)}$

We can repeat the same process on estimation of mutual information. Mutual information can be decomposed in a sum of joint entropy and differential entropy.

kNN estimation of joint entropy

Let U be the joint system of the two random variables $\{T, R\}$ of dimension $2d$. Using the same framework, one can estimate the joint entropy as :

$$\hat{H}^{+}_{\mathrm{kNN}}(X_T, X_R) = \log(v_{2\mathrm{d}}|R|) - \psi(k) + 2d\ \mu_U(\log \rho_k(U,u)). \tag{5.18}$$

kNN estimation of Mutual Information

Mutual information can be written as

$$\begin{aligned}I(X_T, X_R) &= H(X_T) + H(X_R) - H(X_T, X_R)\\ &\overset{\mathrm{kNN}}{=} \hat{H}_{\mathrm{kNN}}(X_T) + \hat{H}_{\mathrm{kNN}}(X_R) - \hat{H}^{+}_{\mathrm{kNN}}(X_T, X_R) && (5.19)\end{aligned}$$

A study showing that these estimates outperform classical Parzen estimates on both univariate and multivariate Gaussian mixtures is available in Appendix A.

5.5 Statistical measures differentiation using mean-shift

The new class of similarity measures introduced in the last section must be differentiated in order to be used in practice in gradient-based variational problems, in particular optical flow (Section 2.2). As these measures are entropy-based, the difficulty is to estimate the gradient $\nabla \log \hat{f}_U(.) = \nabla \hat{f}_U(.)/\hat{f}_U(.)$, also called the normalized gradient of a PDF.

The kNN PDF estimates $\hat{f}_U$ are differentiable almost everywhere. As an illustration let us choose $k = 1$, the PDF estimate is differentiable everywhere, except on the Voronoi diagram of the points, as the nearest neighbor changes. The Voronoi diagram is an hypersurface in the chosen space of measure zero.

Mean-shift is a method to efficiently estimate this normalized gradient while classical estimates fail in general when f vanishes (division by zero). Fukunaga [FH75] remarked that one can write a simple expression of $\nabla f/f$ by combining two different kernel density estimates for f. The approach focused on a Epanechnikov kernels as their gradients are Uniform kernels and thus leads to a simple and intuitive normalized gradient estimation, detailed in Section 5.5.1. It was latter generalized to the general class of kernels and applied in image analysis by Comaniciu [CRM00, Com03], detailed in Section 5.5.2.

5.5.1 Fukunaga approach

Let f be a classical Parzen kernel density estimate, it can be written:

$$\hat{f}(u) = \frac{1}{|U|.h^d} \sum_{u_\mathrm{n} \in U} K\left(\frac{u - u_\mathrm{n}}{h}\right) \tag{5.20}$$

where K is a kernel function, h is the bandwidth of the kernel. We consider the bandwidth to be fixed-size and will extend to variable bandwidth later in this chapter. The gradient of the kernel density estimates reads

$$\nabla_u \hat{f}(u) = \frac{1}{|U|.h^{d+1}} \sum_{u_\mathrm{n} \in U} \nabla K\left(\frac{u - u_\mathrm{n}}{h}\right) \tag{5.21}$$

Fukunaga [FH75] remarked that using a simple Epachnikov kernel:

$$K(X) = \begin{cases} c(1 - XX^T) & \text{if } 0 \leq X^T X \leq 1 \\ 0 & \text{if } X^T X > 1 \end{cases} \tag{5.22}$$

where c is a normalization constant, the gradient expression simply writes

$$\nabla \hat{f}(u) = \left(\frac{k(u)}{|U|.V_h}\right) \frac{d+2}{h^2} M_h(u) \tag{5.23}$$

where $V_h = v_d.h^d$ is the volume of the Parzen window of size h in dimension d, d is the dimension of the features, $k(u)$ is the number of samples falling in the Parzen window of u, and where M_h is the sample mean shift:

$$M_h(u) = \frac{1}{k(u)} \sum_{u_\mathrm{n} \in \mathcal{N}_h(u)} (u_\mathrm{n} - u) = [\mu_h(u) - u] \tag{5.24}$$

where $\mathcal{N}_h(u)$ is the Parzen window centered on u, $\mu_h(u)$ is a mean over the samples falling into the Parzen window $\mathcal{N}_h(u)$ centered on u.

One can recognize in (5.23), the Parzen approximation $\frac{k(u)}{|U|.V_h}$ of f with Uniform kernels (4.3). Thus, one can obtain a simple normalized gradient expression mixing two different kernel density estimates, one with Epanechnikov kernels, the second with Uniform kernels:

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = \frac{d+2}{h^2} M_h(u). \tag{5.25}$$

Informally speaking, the normalized gradient of a PDF at some point can be estimated by computing the shift between this point and the mean over the samples falling into the Parzen window centered on this point. The sample mean-shift (5.24) is here computed using a Parzen fixed-size bandwidth estimate of f but Fukunaga also remarked that mean-shift expression is also valid for variable bandwidth (kNN window). An example of mean-shift in a kNN window is presented on Fig. 5.7.

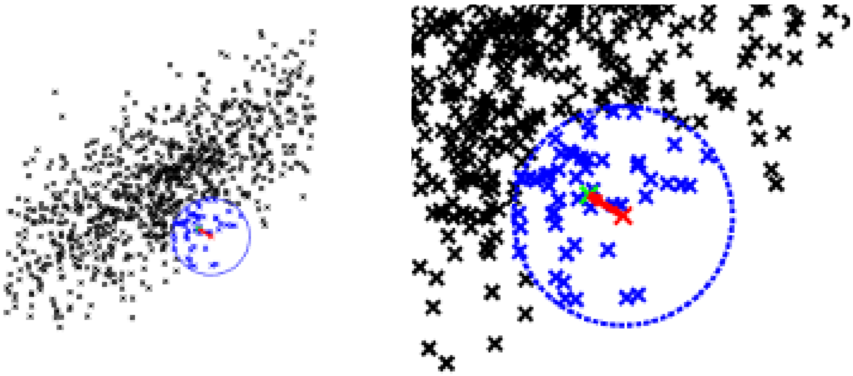


Figure 5.7: *Left: Illustration of Mean-Shift at a sample point on a point cloud, expressed as the shift (red) between a point and the centroid (green) in its kNN window (blue), right: Zoom on the Mean-Shift*

Using a kNN estimate of (5.24), sample mean-shift writes

$$M_k(u) = \frac{1}{k} \sum_{u_{\mathrm{n}} \in \mathcal{N}_k(\mathrm{u})} (u_{\mathrm{n}} - u) = [\mu_k(u) - u] \tag{5.26}$$

where k is for k-th nearest neighbor (equal to the number of u_{n} falling in the kNN window of u), $\mathcal{N}_k(u)$ the kNN window centered on u *i.e.* containing the k nearest neighbor of u, $\mu_k(u)$ is a mean over the k samples falling into the kNN window centered on u

Finally, the kNN mean-shift estimates reads

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = \frac{d+2}{\rho_k^2(U,u)} [\mu_k(u) - u] \tag{5.27}$$

Summarizing, Fukunaga remarked that mixing two different kernel density estimates, one with Epanechnikov kernels, the other with Uniform kernels, one can obtain a simple expression of the normalized gradient. Generalization has been done in [Com03], where the authors remarked that this trick is valid for a general class of kernels K. Kernel K has a profile k [1] and its normalized gradient expression can be simplified by mixing it with a kernel density estimate G of kernel profile g, where $g(x) = -k'(x)$. This work confirmed what Fukunaga found as the gradient of the Epanechnikov kernel profile is the Uniform kernel profile. Let us present this generalization.

[1] $K(x) = k(\|x\|^2)$

5.5.2 Generalized Approach

variable bandwidth mean-shift with any differentiable kernels

In dimension d, a kernel density estimate can be written as

$$\nabla \hat{f}(u) = \frac{1}{|U|} \sum_{u_{\mathrm{n}} \in U} k\left(\left\|\frac{u - u_{\mathrm{n}}}{h_{\mathrm{n}}}\right\|^2\right) \tag{5.28}$$

where k is a kernel profile, h_{n} is the bandwidth which depends on sample point u_{n} (sample point estimate) and can also depend on sample u (balloon estimate). The generalized Mean-shift expression writes:

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = (d+2)h^{-1}(u)M(u) \tag{5.29}$$

with

$$M(u) = h(u) \sum_{u_i \in U} w_i h_i^{-2} u_i - u \tag{5.30}$$

$$h^{-1}(u) = \sum_{u_i \in U} w_i(u) h_i^{-2} \tag{5.31}$$

and

$$w_i(u) = \frac{h_i^{-d} g\left(\left\|\frac{u-u_i}{h_i}\right\|^2\right)}{\sum_{u_{\mathrm{n}} \in U} h_{\mathrm{n}}^{-d} g\left(\left\|\frac{u-u_{\mathrm{n}}}{h_{\mathrm{n}}}\right\|^2\right)} \tag{5.32}$$

where g is a kernel profile derivated from k, $g(x) = -k'(x)$.

Restriction to Epanechnikov kernels

Restricting k to Epanechnikov profile as performed by [FH75]:

$$k(x) = \begin{cases} 1 - x & \text{if } 0 < x < 1 \\ 0 & \text{if } x > 1 \end{cases} \tag{5.33}$$

the gradient profile $g(x)$ is piecewise constant.

Let us introduce generality on the variability of the bandwidth. The three different neighborhoods, introduced and compared in Section 5.3, corresponding to fixed and variable bandwidth PDF estimates, read

- The fixed-size neighborhood

$$\mathcal{N}_F(U, u) = \{u_{\mathrm{n}} \in U : \; \|u - u_n\| \leq h\} \tag{5.34}$$

- The kNN balloon neighborhood

$$\mathcal{N}_B(U, u) = \{u_{\mathrm{n}} \in U : \; \|u - u_n\| \leq \rho_k(U, u)\} \tag{5.35}$$

  if there is no ambiguity about what is the data set U, $\mathcal{N}_B(U, u)$ will be simply noted $\mathcal{N}_B(u)$

- The kNN sample point neighborhood

$$\mathcal{N}_{SP}(U, u) = \{u_{\mathrm{n}} \in U : \; \|u - u_n\| \leq \rho_k(U, u_n)\} \tag{5.36}$$

The mean shift (5.29) simplifies to:

$$w_i(u) = \begin{cases} \frac{\rho_k(U,u_i)^{-d}}{\sum_{u_{\mathrm{n}} \in \mathcal{N}(U,u)} \rho_k(U,u_{\mathrm{n}})^{-d}} & \text{if } i \in \mathcal{N}(U,u) \\ 0 & 0 \text{ otherwise} \end{cases} \tag{5.37}$$

where $\mathcal{N}$ can be any of the three above mentioned neighborhoods,

$$h^{-1}(u) = \frac{\sum_{u_{\mathrm{n}} \in \mathcal{N}(u)} \rho_k(U,u_n)^{-(d+2)}}{\sum_{u_{\mathrm{n}} \in \mathcal{N}(u)}, \rho_k(U,u_{\mathrm{n}})^{-d}} \tag{5.38}$$

$$M(u) = \frac{\sum_{u_{\mathrm{n}} \in \mathcal{N}(u)} \rho_k(U,u_n)^{-(d+2)} u_{\mathrm{n}}}{\sum_{u_{\mathrm{n}} \in \mathcal{N}(u)}, \rho_k(U,u_{\mathrm{n}})^{-(d+2)}} - u \tag{5.39}$$

Using the corresponding neighborhood, we retrieve the expression of the original fixed-bandwidth mean-shift (5.25)

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = \frac{d+2}{h^2} . \frac{1}{|\mathcal{N}_F(U,u)|} \sum_{u_{\mathrm{n}} \in \mathcal{N}_F(U,u)} (u_{\mathrm{n}} - u) \tag{5.40}$$

as well as the balloon mean-shift (5.27)

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = \frac{d+2}{\rho_k^2(U,u)} . \frac{1}{k} \sum_{u_{\mathrm{n}} \in \mathcal{N}_B(U,u)} (u_{\mathrm{n}} - u) = \frac{d+2}{d_k^2(u)} . [\mu_B(u) - u] \tag{5.41}$$

where $d_k(u)$ is a short notation for $\rho_k(U,u)$ and $\mu_B(u)$ is a short notation for the mean in a balloon neighborhood around u (of radius $d_k(u)$) containing the k-th nearest neighbors of u. This expression will demonstrate good performance on Gaussian mixtures and clustering applications (Appendix A.4), it will be the expression used in this manuscript and called kNN mean-shift.

Finally the expression of sample point mean-shift reads

$$\frac{\nabla \hat{f}(u)}{\hat{f}(u)} = (d+2) \frac{\sum_{u_{\mathrm{n}} \in \mathcal{N}_{SP}(u)} \rho_k(U,u_n)^{-(d+2)}}{\sum_{u_{\mathrm{n}} \in \mathcal{N}_{SP}(U,u)} \rho_k(U,u_{\mathrm{n}})^{-d}} \left(\frac{\sum_{u_{\mathrm{n}} \in \mathcal{N}_{SP}(U,u)} \rho_k(U,u_n)^{-(d+2)} u_{\mathrm{n}}}{\sum_{u_{\mathrm{n}} \in \mathcal{N}_{SP}(U,u)} \rho_k(U,u_{\mathrm{n}})^{-(d+2)}} - u \right) \tag{5.42}$$

Experiences on various Gaussian mixtures are presented in Appendix A.4. In particular, (balloon) kNN mean-shift converges faster than Parzen mean-shift. It is also never equal to zero, which is an advantage when trying to classify a point far from the modes of the distributions, as it always indicates a direction while Parzen and sample-point mean-shift would be stuck.

5.6 Conclusion

In this chapter, we derived a new class of statistical measure estimate to deal with variational problems. These statistical measures are efficient in high dimensions, have a locally adaptive PDF interpretation, and its gradient can also be approximated. Two estimates of local density were compared along this chapter, balloon and sample point. Both for simplicity of the expressions and non-zero tails leading to interesting properties for mean-shift and clustering (see Appendix A.4), we will focus on the balloon estimate.

Even if this balloon adaptive density estimate does not preform well in one dimension against Parzen estimates, it behaves better in higher dimensions [TS92]. Moreover, even if

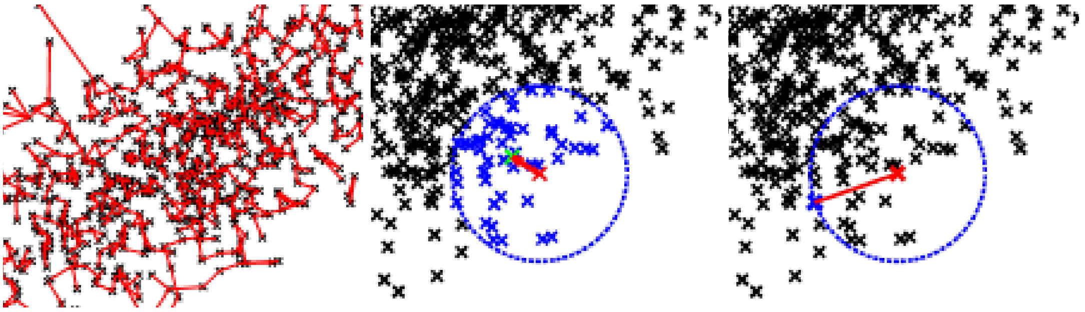


Figure 5.8: *kNN framework (from left to right): Entropy of a point cloud as the mean of log kNN distances, gradient of Entropy at a sample point as a kNN Mean-Shift, probability of a sample point as the distance to its kNN*

this estimate is not efficient in 1-D, its associated statistical measure estimate outperforms the natural Parzen-based entropy estimate (see Appendix A.4). In higher dimensions, the difference between kNN and Parzen estimation increases as Parzen reaches its limitation. Experimental results show that even on various test on Gaussian mixtures in Appendix A, which is a favorable case for Parzen estimate with Gaussian kernel shapes, the new statistical measure estimates is performing better. This is very encouraging for image processing applications as the data are more complex than Gaussian distributions and may be higher-dimensional.

Finally, complexity in this framework relies on the k-th nearest neighbor search. Efficient tools allow to estimate nearest neighbors quickly. These methods use k-D trees, one of the most famous is called Approximate Nearest Neighborhood (ANN) [AM93]. ANN proposes to quickly estimate nearest neighbors with k-d tree construction. We also used a library proposed by [GDB08] to compute nearest neighbors on graphical processing unit (GPU) in order to speed up the computations.

Summarizing, the main result of this chapter is a new kNN framework (Fig. 5.8) which

1. estimates efficiently information theory measures such as
   - differential entropy: kNN entropy estimate (5.11)
   - Kullback-Leibler divergence: kNN Kullback estimate (5.16)
2. comes from (balloon) locally adaptive PDF (5.4), kNN PDF estimate (5.5)
3. approximates measure and PDF gradients through variable bandwidth mean-shift, kNN mean-shift estimate (5.41).

Along this thesis we will mainly use the high dimensional space $(x, y, u(x, y))$ as a space to define image probabilities Fig. 5.2(d). The 500 nearest neighbors of a sample point on this figure are the neighbors used to compute statistics in the kNN framework (PDF estimate, statistical measure estimate, mean-shift estimate). This space is simply computed with appending dimensions to the feature values as showed on Fig 5.9.

Appending "geometric" dimensions can better model image distributions as it captures local patterns. PDF estimation is then called nonlocal, it builds statistics over all the image but using a spatial constraint.

High dimensional feature space, which statistics are estimated with the kNN framework, will revisit the three variational problem, in tracking to define smooth matching as presented

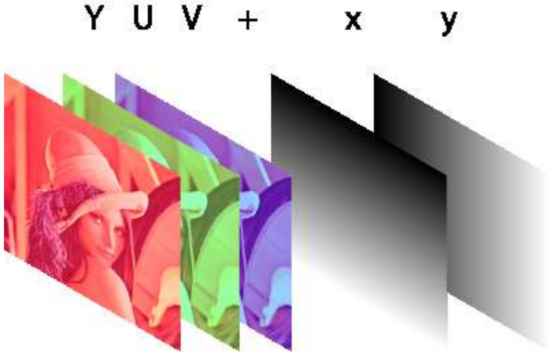


Figure 5.9: *High dimensional feature space to capture local patterns in image structure*

in this chapter, in optical flow, to define adaptive regularization, and in segmentation to define local statistics for segmentation.

Part III

Revisiting three variational problems in the kNN framework

CHAPTER **6**

TRACKING

This chapter will revisit kernel tracking algorithms, with high dimensions. Distance between high dimensional templates of the data will be able to model and take into account geometric deformations. These distances will be efficiently estimated using the kNN framework and minimized using either derivative-free and gradient descent algorithms. This chapter follows the general introduction written in Section 2.1.

6.1 Introduction

Tracking a region of interest (ROI) in a video is still a challenging task. Various high-level applications rely on tracking, *e.g.*, motion picture indexation, object recognition, video surveillance, audiovisual post-production... The problem can be defined as follows: an ROI is defined in a reference frame and the purpose is to determine in each subsequent frame the region which best matches the ROI in terms of a given similarity measure. Geometrically speaking, the two regions can be deduced from one another by an apparent motion that one usually restrict to a given model. Two classical similarity measures are the Sum of Squared Differences (SSD) or the Sum of Absolute Differences (SAD) between the reference ROI and a candidate region in a target frame.

6.1.1 A statistical approach for tracking

Similarity measures such as SSD and SAD impose a strict geometric constraint since the underlying residual is computed with a deterministic pixel-to-pixel correspondence between the reference ROI and the target region. In general, this apparent motion follows a rather simple model so that the estimation of its parameters remains well-posed. Therefore, it is not adapted to complex motions. Moreover, this type of similarity measures correspond to implicit parametric assumptions on the residual probability density function (PDF) (respectively, Laplacian and Gaussian for the two examples above). A solution is to relax this assumption by using nonparametric estimate on the residual.

An alternative is to adopt another statistical point of view by building a PDF from the ROI and using it as a template to be compared to a target PDF built from a candidate region by means of a similarity measure. Such statistical methods account for randomness and uncertainty in the observations. At the first level of complexity, the PDFs describe the ROI radiometry [CRM00, PHVG02], either in grayscale or color. However, to improve tracking accuracy, later developments tend to show that more information is required than just color. Different cues were then integrated into the ROI PDF template, *e.g.*, recurring to the use of filters such as spatial derivative filters [Low04, BRDW03, BBPW04], Gabor or wavelet filters [PD02b], and temporal filters [BRDW03, BP07]. A review of methods based on this framework proposed for segmentation was recently carried out [CRD07].

While this increase of knowledge improves accuracy, the combination of cues leads to high-dimensional PDFs. There exist efficient [Sco92, Ihl] and fast [YDGD03] methods to estimate multivariate PDFs using Parzen windowing. However, due to the fixed cardinality of the data set, a limitation known as the curse of dimensionality [Sco92] appears: as the dimension of the domain of definition of the PDFs gets higher, the domain sampling gets sparser. One can think of dilating the Parzen window [BP07] so as to ensure that it will enclose enough samples. However, the resulting PDF is oversmoothed. Another standard solution is to assume independence between the different cues in order to bring out low-dimension marginal laws [BRDW03] and/or make some parametric assumptions on the PDFs [EDD03]. While these solutions may be satisfactory in some cases, we will discuss in Section 6.1.2 why they are inappropriate for tracking.

6.1.2 High-dimensional feature space for tracking

To define a suitable high-dimensional feature space is to make a trade-off between decreasing the number of cues, thus being less exposed to the curse of dimensionality, and increasing the amount of (relevant) information about the ROI to ensure a reliable tracking. Therefore, features should be chosen carefully as opposed to adding as many features as possible.

The combination of color and geometry proved to be efficient for tracking. This will be referred to as a soft geometric constraint since the geometric correspondence between the ROI and the target region will be imposed by a similarity measure between PDFs. In some works, spatial information has been added by means of a Gaussian weighting of the samples according to their distance to the center of the ROI [CRM00, PHVG02]. This weighting can be seen as a radial layout constraint. This approach has the advantage not to add any dimension to the feature space. However, it lacks generality. Geometry can instead be added directly to the radiometric vector (or any other feature vector), *e.g.*, in the form of the Cartesian coordinates of the pixels of the ROI [EDD03]. Independence between color and geometry cannot be assumed in order to avoid to manipulate high-dimensional PDFs. Indeed, geometry alone, seen as a random variable, follows a uniform distribution whether in the ROI or in the target region and, therefore, brings no information. While considering color and geometry jointly, simplification can still be achieved by approximating the PDFs with parametric laws. Nevertheless, fully data-driven nonparametric PDF estimation was advantageously applied to segmentation [KFY$^+$05, HSD$^+$06].

6.1.3 Proposed approach

We propose to use the k-th nearest neighbor (kNN) framework in order to be able to handle the components of a high-dimensional feature vector jointly, non-parametrically, and to work in a locally adaptive manner in the feature space, thus avoiding under or oversmoothing in processing the data set. Although kNN PDF estimators were proposed a long time ago [FH51, LQ65], they did not received much attention since they were known to be biased [TS92, Sai02]. Recently though, corrective terms have been derived to cancel the bias and led to consistent kNN-based statistical measures such as entropy [KL87, GLMI05]. Moreover, even if the kNN PDF estimator is only adapted to high dimensions [TS92], the resulting entropy estimator appears to be accurate in both low and high dimensions.

In this context, the Kullback-Leibler divergence between high-dimensional PDFs will be suggested as a similarity measure for tracking. The divergence will be expressed directly from the samples and, therefore, its computation does not require explicit estimation of the underlying PDFs. This divergence estimator being well-adapted to high dimensions, it can be used in an extended radiometric/geometric feature space [BDB07].

The chapter is organized as follows: Section 6.2 first provides some notations and motivates the framework used for tracking, Section 6.3 motivates the choice of high dimensional space and how it can handle deformations; Section 6.4 details the ROI tracking algorithm; Finally, Sections 6.5 and 6.6 provide some results and comments for several standard sequences.

6.2 Tracking formulation

6.2.1 Problem statement

Let I_{ref} and I_{tgt} be, respectively, the reference frame in which the ROI is (user-)defined and the target frame in which the region that best matches the ROI, in terms of a given similarity measure, is searched for. This search amounts to finding the geometric transformation Φ such that

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_1(I_{\text{ref}}(\Omega), I_{\text{tgt}}(\varphi(\Omega))) \tag{6.1}$$

where $\mathfrak{D}_1$ is a similarity measure between two data sets and Ω is the domain of the ROI. Domain Ω is a subset of $\mathbb{R}^2$ or a subset of $\mathbb{N}^2$ in the discrete framework. This search is illustrated on Fig. 6.1.

For clarity, the reference data set $I_{\text{ref}}(\Omega)$ will be denoted by R and the target data set $I_{\text{tgt}}(\varphi(\Omega))$ will be denoted by T_φ. Thus, $R(i)$ and $T_\varphi(i), i \in \Omega$, represent corresponding samples from their respective region.

Since a statistical approach was preferred over the deterministic formulation (6.1), the geometric transformation Φ is instead expressed as

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_2(f_R, f_{T_\varphi}) \tag{6.2}$$

where f_R, respectively f_T, is the PDF which generated the samples $\{R(i), i \in \Omega\}$, respectively $\{T_\varphi(i), i \in \Omega\}$. Whenever appropriate, U will be used as a generic notation for either R or T_φ. Traditionally, $U(i)$ is a triplet of color components in a given color space. From Section 6.3 on, some geometric information will be appended to this feature vector. Generally speaking, the samples will be regarded as elements of $\mathbb{R}^d$.

6.2.2 Kullback-Leibler divergence

Several measures have been introduced to quantify the disagreement between two PDFs. In this chapter, the Kullback-Leibler divergence $\mathfrak{D}_{\text{KL}}$, or information gain, has been chosen. The

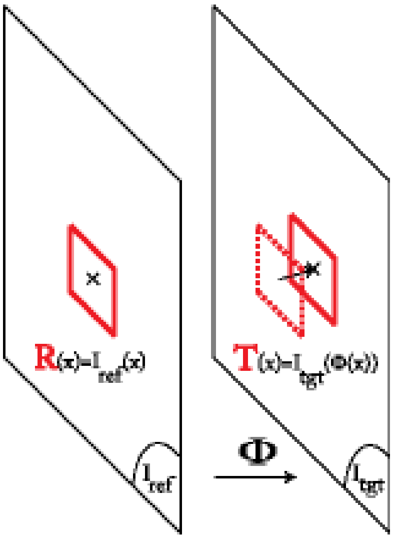


Figure 6.1: *Image matching example: find the transformation that maps a region of an image to another*

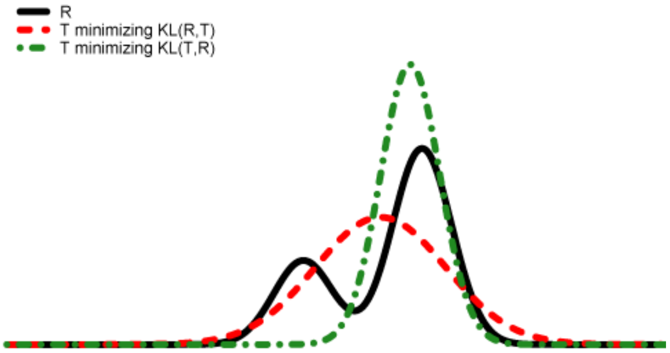


Figure 6.2: *Order of the arguments of the Kullback-Leibler divergence: zero-forcing and zero-avoiding solutions (Image courtesy of Pierre Dangauthier, Ph.D, E-Motion project, INRIA Rhone Alpes / LIG, Grenoble, France).*

discussion below about the order of the arguments of the divergence[1] explains what motivated this choice.

Let us reformulate the problem this way: f_R is a reference PDF and the best approximation f_{T_φ} of it must be found. Minimizing $\mathfrak{D}_{\mathrm{KL}}(T,R)$ leads to a so-called zero-forcing solution [Min05]: wherever f_R is close to zero, the solution is strongly encouraged to be close to zero as well. As a consequence, f_{T_φ} "focuses" on the dominant mode of f_R, thus underestimating the variance of f_R. This solution is also called exclusive since it can exclude some parts of f_R. Minimizing $\mathfrak{D}_{\mathrm{KL}}(R,T)$ leads to a so-called zero-avoiding solution [Min05]: the solution is encouraged to cover the whole support of f_R. As a consequence, f_{T_φ} usually overestimates the variance of f_R (see Fig. 6.2).

Various works proposed symmetric versions of the Kullback-Leibler divergence, *e.g.*, J-divergence and Jensen-Shannon divergence [Lin91]. Nevertheless, for tracking, $\mathfrak{D}_{\mathrm{KL}}(T,R)$ seems to be the appropriate choice. Indeed, f_{T_φ} can never be identical to f_R due to noise,

[1] Remember that the Kullback-Leibler divergence is not symmetric.

occlusion, motion blur, and the fact that a frame is a projection onto a two-dimensional plane of a three-dimensional scene. However, both should have the same main modes if they correspond to the same object. Thus, the zero-forcing divergence enforces a relevant behavior in trying to "align" the main modes of the PDFs. By the way, it follows the same philosophy as the Bhattacharya distance, a measure widely used for tracking since a Mean-Shift-based implementation has been proposed [CRM00].

Using the efficient estimate presented in Chapter 5, the Kullback-Leibler divergence is equal to

$$\begin{aligned} \mathfrak{D}_{\mathrm{KL}}(T_\varphi, R) &= H^\times(T_\varphi, R) - H(T_\varphi) && (6.3) \\ &\overset{\mathrm{kNN}}{=} \log \frac{|R|}{|T_\varphi| - 1} + d\ \mu_{T_\varphi}(\log \rho_k(R)) \\ &\quad -d\ \mu_{T_\varphi}(\log \rho_k(T_\varphi)). && (6.4) \end{aligned}$$

It has been proven that this estimator is consistent and asymptotically unbiased [GLMI05, KL87]. The choice of feature space for R and T has now to be carried out.

6.3 Similarity-based tracking: handling geometry

As noted earlier, the radiometric feature vector will be enriched with geometry. Radiometry allows to check if the ROI and the target region have similar colors, and geometry allows to check with a given degree of strictness if these colors appear at the same location in the regions. For comparison purposes (see Section 6.5), let us describe levels of strictness.

6.3.1 Classical similarity measures

Geometry-free similarity measures

The similarity measure between the ROI and the target region can be based solely on radiometry. Classically, it can be a distance between color histograms or, similarly, PDFs. The knowledge of where a given color was present within the region is lost. For example, let us mention the Bhattacharya distance [CRM00, PHVG02]

$$\mathfrak{D}_{\mathrm{BHA}}(T_\varphi, R) = \int_{\mathbb{R}^d} \sqrt{f_R(s)\ f_{T_\varphi}(s)}\ \mathrm{d}s \qquad (6.5)$$

where d is equal to three if all color components are used. The Kullback-Leibler divergence on geometry-free PDFs will also be tested in Section 6.5.

Not accounting for the knowledge of where a given color was present in the region allows to be more flexible regarding the geometric transformation φ between the ROI and the target region. However, it increases the number of potential matches and then the risk for the tracking to fail after a few frames. This can be avoided by using a geometry-aware similarity measure.

Similarity measures with strict geometry

Geometry can be involved by means of a motion model (*i.e.*, a constraint on φ) used to compute a pointwise residual between the ROI and a candidate region. A function of the residual can serve as a similarity measure: classically, the SSD or functions used in robust estimation [BA96] such as the SAD. The geometric constraint being strictly defined by the motion model, these measures might be less efficient if the model is not coherent with the actual motion. Indeed, this might generate too many outliers in the residual, including in the framework of robust estimation. Moreover, even if the model is globally coherent with the actual motion, the choice of the function of the residual is implicitly linked to an assumption on the PDF of the residual, *e.g.*, Gaussian for SSD or Laplacian for SAD. This might not be valid in case of occlusion for example.

To fix the ideas, let us assume that $|T_\varphi| = |R|$ and let us define the following notations

$$\mathfrak{D}_{\text{SSD}}(T_\varphi, R) = \sum_{i \in \Omega} (T_\varphi(i) - R(i))^2 \tag{6.6}$$

and

$$\mathfrak{D}_{\text{SAD}}(T_\varphi, R) = \sum_{i \in \Omega} \phi(T_\varphi(i) - R(i)) \tag{6.7}$$

where ϕ can be either the absolute value or a smooth approximation of it, *e.g.*, $\phi(x) = \sqrt{x^2 + \epsilon^2} - \epsilon$ [WS01].

6.3.2 Matching with uncertainties on appearance: entropy

Classical strict geometry similarity measure $\mathfrak{D}_1(R - T_\varphi)$ in image matching use the SSD or SAD for criteria $\mathfrak{D}_1$. The SSD estimator is fully efficient only in the case of normally distributed data, see e.g. [GM95].

The Gaussian assumption is not appropriate in image matching problems. Here we consider instead minimizing the entropy of the estimation error. The properties of the measure of entropy motivate its use as an estimation criterion: entropy is a convex function of the transformation φ that coincides locally asymptotically with the likelihood at its optimum [WTP05]. This suggests that an estimator minimizing the entropy of the errors should be efficient. Also, the shift-invariance property of the entropy of a density yields some robustness to outliers, e.g. unexpected patches of pixels in an image, and makes it insensitive to a global variation of illuminance between the two regions. In the present context we consider the residual on region $R - T_\varphi$, viewed as realizations of a random variable. Knowledge (or estimation) of the probability mass function (in the discrete case) or of the probability density function (in the continuous case) of these errors is required in order to compute the entropy of the block. A minimum-entropy approach for discrete data would require a large number of data points (pixels) in order to obtain an accurate enough estimation of the distribution, which makes it unsuitable for matching purposes. Instead we consider turning the residual into an image of continuously distributed pixel values. A uniform noise $\mathcal{U}(-.5; .5)$ is added to the residual $R - T_\varphi$ (this choice is arbitrary, other types of distributions could be used). In this context, a common PDF estimation procedure consists in using Parzen windowing technique, which provides a smooth estimate that can be plugged into an empirical expression of the Shannon entropy. Minimum-entropy estimation gives a consistent estimator of the parameters in a regression model with unknown distribution of the observations errors, see [WTP05].

The minimum-entropy estimator of the transformation φ that minimizes the Ahmad-type plug-in entropy estimate [AL76] is defined by

$$\Phi = \arg\min_{\varphi} H(I_{\text{ref}}(R) - I_{\text{tgt}}(T_\varphi)) \tag{6.8}$$

with

$$H(U) = -\frac{1}{|U|} \sum_{u_i \in U} \log \hat{f}_U(u_i)$$

Here, $\hat{f}_U$ denotes a kernel density estimate, estimated on samples $|U|$, which bandwidth is set either by a data-driven procedure [Sco92, DL00, BD94] or by the user. In dimensions larger than 1, e.g. for color images, kernel estimation techniques rapidly become inefficient when the dimension increases. The main difficulty lies in the choice of the kernel bandwidth: due to the curse of dimensionality, the bandwidth of a multivariate kernel must be large enough to take a sufficient number of data points into account, which causes oversmoothing. This leads to a degradation of the performances of the estimator as defined by (6.8) for data having dimension 2 or 3 and for samples of reasonable size. For 3D (i.e. color) images, the alternative of using a product of univariate kernels still remains relatively computationally costly and does not perform well enough in general. New entropy estimators defined in Chapter 5 efficiently circumvents these problems.

This robust strict geometry image matching model was successfully applied in applications where there is good confidence on the transformation model: motion estimation between two following frames including Block Matching [BWD$^+$06] and variational optical flow [BDB08]. The latter application will be detailed in Chapter 7. Robustness of this method against various types of noise and comparisons to classical strict geometry measures are presented in Appendix B.

Strict geometric approaches are robust if there is good confidence on the transformation model. However, as deformations become too important[2] these methods reach their limitations as the residual is of high energy. On the other hand, methods based on divergences between radiometric PDF templates (6.2) are flexible with respect to the transformation model, but not discriminative as many regions yield the same PDF.

6.3.3 Matching with uncertainties on position and appearance: divergence

An ideal measure should allow both flexibility on the appearance (color values) but also flexibility on the pixel positions as the transformation model may be incorrect. On one hand, the strict geometric constraint can be softened, *e.g.*, by cascading a strict geometry approach and a radiometric approach [VBPB07], or by relaxing parametric assumptions on the data. On the other hand , constraints on non-geometric measures can be integrated, by adding geometry to the PDF-based approach, *i.e.*, by defining a joint radiometric/geometric PDF [EDD03, BDB07].

On one viewpoint, it adds uncertainty in the transformation model in a strict geometry method, on another viewpoint, it imposes some geometric constraints in a non-geometric model. The PDFs of the templates R and T now contain some geometry information describing the position of pixels. This method will be called soft geometric.

Soft geometry model

A joint radiometric/geometric PDF is defined [EDD03, BDB07]. Formally, the PDF f_U corresponding to the sample set $\{U(i), i \in \Omega\}$ is replaced with the PDF $f_{U,i}$ corresponding to the sample set $\{(U(i), i), i \in \Omega\}$. Therefore, the color+geometry feature space is equal to $\mathbb{R}^5$. In general, i can be any couple of independent spatial coordinates. For the ROI tracking application presented here, Cartesian coordinates (x, y) relative to the center of mass of the ROI

[2] for instance, between the first frame of a video and a frame 10 seconds later, assuming the reference region is still visible

or of the target region seem adapted as shown on Fig. 6.3.

The sample set R and T are now enriched with geometry and are in dimension 5. the corresponding PDFs f_R and f_T are also in dimension 5 and we choose as similarity measure the Kullback-Leibler divergence [3]. The high dimensional feature space is handled by the kNN framework defined in Chapter 5 using estimate (6.4).

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_{\mathrm{KL}}(f_R, f_{T_\varphi}) \tag{6.9}$$

This choice is motivated by the asymptotic behavior of this space : suppose we need to estimate the distance of the point $T(s), \alpha.s$ in a reference set $(R(i), \alpha.i)$ where α is a weighting constant. As α tends to infinity, the closest point to $T(s)$ is $R(s)$ as there is an infinite penalty on uncertainty pixel position. The measures is strict as the nearest neighbor search is now $\|T(s) - R(s)\|$. As α tends to zero, the geometric features have no influence in the nearest neighbor search, leading the measure to be classical PDF non-geometric matching. A reasonable choice for α, in this paper we choose $\alpha = 1$ is a soft-geometry compromise between non-geometric and strict-geometric matching.

Geometric features

Several geometric feature choices are possible depending on the application. Matching rectangular regions, the canonical choice is to choose Cartesian coordinates. Matching circular regions, the canonical choice is polar coordinates. Matching deformable and complex shapes, more complex choices will be introduced. Basic choices for similarity measures are illustrated on Fig. 6.3.

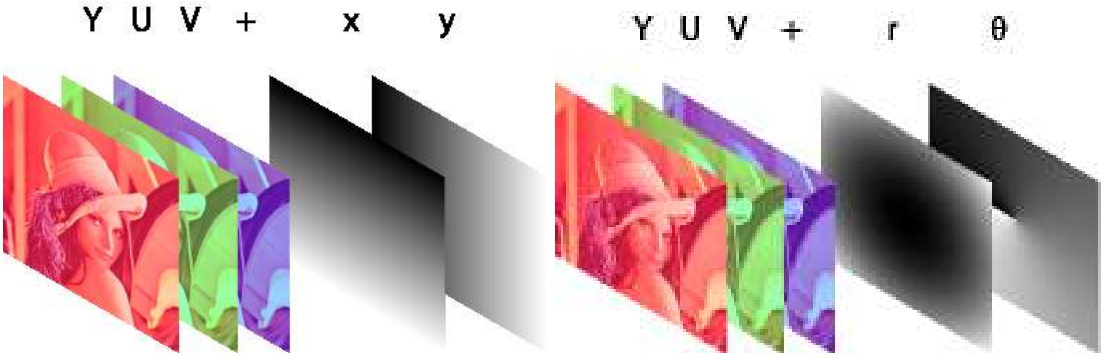


Figure 6.3: *High dimensional feature to add uncertainty on position: add lattice information to observations: Cartesian or polar coordinates*

6.4 Tracking algorithm

6.4.1 The main steps

As a reminder, R and T_φ are the following sample sets

$$\left\{ \begin{array}{lcl} R & = & \{(I_{\mathrm{ref}}(i), i), i \in \Omega\} \\ T_\varphi & = & \{(I_{\mathrm{tgt}}(i), i), i \in \varphi(\Omega)\} \end{array} \right. \tag{6.10}$$

[3] this choice was motivated in section 6.2.2

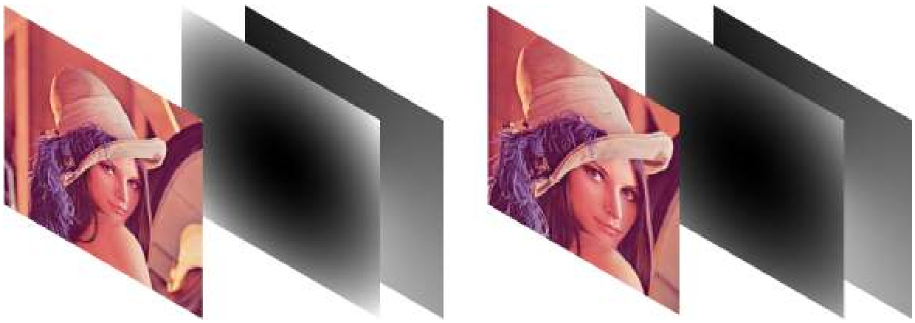


Figure 6.4: *zoom factor is retreived by aligning geometric components, no interpolation of the radiometric features is used*

where i represents some Cartesian coordinates (x, y) relative to the center of mass of the corresponding region (Ω for R and $\varphi(\Omega)$ for T_φ), and φ is a geometric transformation representing the motion of the ROI between the reference frame and the target frame. Then, we propose to perform tracking by minimizing the kNN Kullback-Leibler divergence (6.4) between f_R and f_{T_φ} with respect to φ, or actually a set of parameters defining φ

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_{\mathrm{KL}}(T_\varphi, R). \tag{6.11}$$

The chosen motion model is "translation+scaling"

$$\begin{aligned} \varphi(i) &= i + M(i)\, p & (6.12)\\ &= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x & 1 & 0 \\ y & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha - 1 \\ u \\ v \end{bmatrix} & (6.13) \end{aligned}$$

where α is the scaling factor and (u, v) is the translation. The main steps of the tracking algorithm are

In this algorithm, λ is a test set of scaling factors. For the experiments of Section 6.5, λ was chosen equal to $\{0.98, 0.99, 1, 1.01, 1.02\}$ or $\{1\}$ depending on whether scaling was taken into account or not. One nice property about the geometric features is that one can match two regions at different scales without interpolating colors, as shown on Fig. 6.4, zoom is simply recovered by interpolating geometric features (basic multiplication) in order to align both radiometric and geometric features.

6.4.2 Mean-Shift-based gradient descent using a kNN implementation

The derivative of the Kullback-Leibler divergence reads

$$\nabla_\varphi \mathfrak{D}_{\mathrm{KL}}(T_\varphi, R) = \sum_{s \in T_\varphi} \mathcal{D}_s(T_\varphi) \left(\frac{\nabla f_R(s)}{f_R(s)} - \frac{\nabla f_{T_\varphi}(s)}{f_{T_\varphi}(s)} + \frac{1}{|T_\varphi|} \sum_{t \in T_\varphi} \frac{\nabla K_h(t-s)}{f_{T_\varphi}(t)} \right). \tag{6.14}$$

Estimation (6.4) being defined in the kNN framework, one can approximates its derivative

Table 6.1: *Tracking algorithm.*

1. Set the parameters
   - Neighboring order: $k \overset{e.g.}{\leftarrow} 3$
   - Spatial weight: $\delta \overset{e.g.}{\leftarrow} 1$
   - Scaling factors: $\lambda \overset{e.g.}{\leftarrow} \{0.98, 0.99, 1, 1.01, 1.02\}$
   - Radiometric function: $U(i) \overset{e.g.}{=} I(i)$
2. Manually select an ROI Ω in the reference frame I_{ref}
   - (a) Let $i_R = (x_R, y_R)$ be the normalized Cartesian coordinate system relative to Ω
     Perform either 2b or 2c depending on the minimization strategy (see below)
   - (b) *Either:* Set $R_\alpha = \{(I_{\text{ref}}(i_R), \alpha\delta i_R),\ i_R \in \Omega\}$ for all $\alpha \in \lambda$
   - (c) *Or:* Set $R = \{(I_{\text{ref}}(i_R), \delta i_R),\ i_R \in \Omega\}$
3. Let φ be the triplet (α, u, v) equal to $(1, 0, 0)$ initially
4. For each remaining frame I_{tgt} taken sequentially
   - (a) Let $i_T = (x_T, y_T)$ be the normalized Cartesian coordinate system relative to $\varphi(\Omega)$
     Perform minimization using either strategy 4b or strategy 4c
   - (b) *Either:* Perform a series of minimizations as follows
     - i. For each $\beta \in \lambda$
       - Determine the translation (m, n) such that

         $$(m, n) = \arg\min_{(a,b)} \mathfrak{D}_{\text{KL}}(T_{(a,b)}, R_\beta)$$

         where $T_{(a,b)} = \{I_{\text{tgt}}(i_T + (a, b)), \delta i_T),\ i_T \in \varphi(\Omega)\}$
       - Let $\mathfrak{D}_\beta$ be equal to $\mathfrak{D}_{\text{KL}}(T_{(m,n)}, R_\beta)$
     - ii. Determine the triplet $(\tilde{\beta}, \tilde{m}, \tilde{n})$ that gave the lowest $\mathfrak{D}_\beta$ among the $|\lambda|$ loops of 4(b)i
   - (c) *Or:* Perform a gradient descent in (α, u, v) (see Section 6.4.2) to determine the triplet $(\tilde{\beta}, \tilde{m}, \tilde{n})$ that minimizes $\mathfrak{D}_{\text{KL}}(T_{(m,n)}, R_\beta)$ where R_β is obtained by multiplying the geometry stored in R by β
   - (d) $\varphi = (\alpha, u, v) \leftarrow (\alpha\,\tilde{\beta}, u + \tilde{m}, v + \tilde{n})$
   - (e) $\varphi_{\text{tgt}} \leftarrow \varphi$

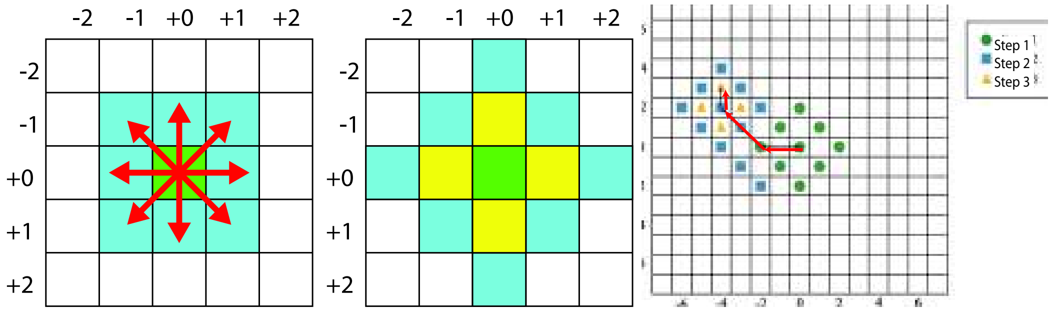


Figure 6.5: *On the left, gradient-descent search: motion is initialized at green position, the gradient direction is computed and projected on one of the* 8 *vectors, next motion is then at blue position. On the Middle, the diamond descent: motion is initialized at green position, the* 8 *blue positions are tested, at convergence the* 4 *yellow positions are tested. On the right, both algorithms are repeated on a grid until the distance is not decreasing anymore.*

using Mean-Shift.

$$\nabla_\varphi \mathfrak{D}_{\mathrm{KL}}(T_\varphi, R) = -\frac{1}{k\,|T_\varphi|}\sum_{s\in T_\varphi}\mathcal{D}_s(T_\varphi)\left(\frac{d+2}{\rho_k^2(R,s)}\sum_{t\in\mathcal{N}_B(R,s)}(t-s) - \frac{d+2}{\rho_k^2(T_\varphi,s)}\sum_{t\in\mathcal{N}_B(T_\varphi,s)}(t-s) - \sum_{\substack{t\in T_\varphi\\ |t-s|=\rho_k(T_\varphi,t)}}\frac{t-s}{\rho_k(T_\varphi,t)}\right) \tag{6.15}$$

where $\mathcal{D}_s(T_\varphi)$ is a $3\times d$-matrix involving frame gradients, $\mathcal{N}_B(\cdot,s)$ are balloon neighborhoods (5.35) of radius $\rho_k(\cdot,s)$ centered at sample s, and K_σ is a kernel of bandwidth σ. (The complete definitions of these terms are given along the development leading to derivative (6.15) presented in Appendix C.) As a consequence, the ROI tracking could be solved by gradient descent in the space of the parameters (α, u, v). However, the sensitivity of the similarity measure with respect to the scaling α is much higher than the sensitivity with respect to translation. In practice, this can lead to undesirable convergence behaviors such as finding a match in the target frame at a scale different from the scale of the reference ROI (*i.e.*, the reference could be matched to a region much larger or much smaller). Therefore, a procedure based on a series of minimizations will be preferred (see Section 6.4.3).

6.4.3 Series of minimizations

The minimization of (6.4) with respect to $\varphi = (\alpha, u, v)$ can be performed by a series of minimizations in (u, v) at α fixed, as illustrated in the algorithm of Section 6.4.1. This decoupling allows to confine α to a reasonable interval, *e.g.*, $[0.98, 1.02]$. The minimizations in (u, v) can be achieved by a gradient descent setting the α-component of (6.15) to zero. For computational considerations, they can be performed using a suboptimal search procedure such as the diamond search [ZM00]. Naturally, more sophisticated search techniques such as particle filters [PHVG02][4], also known as sequential Monte Carlo methods, can be used. Fig. 6.5 presents the two chosen descent possibilities: gradient-guided search (positions are tested following the gradient direction) or diamond search (all directions are searched for).

[4] These methods are particularly efficient in case of total occlusion of the target on several frames.

6.5 Experimental results

6.5.1 Distances comparison

In this part we show the differences between strict geometric methods, non-geometric methods and soft geometric methods. As an illustration, we compare in a first step some similarity measures between two consecutive frames of sequence “Football”. In a second step, we compare some similarity measures between two frames which are not consecutive as shown on Fig. 6.6.

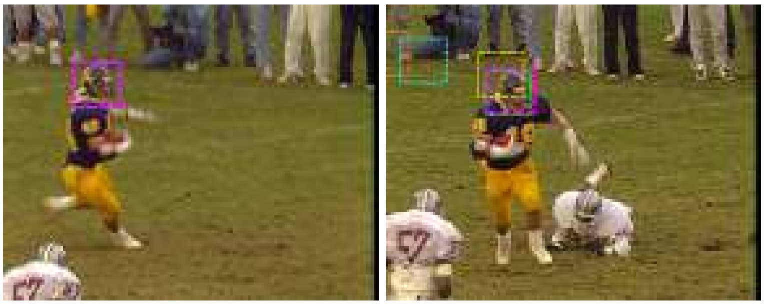


Figure 6.6: *Sequence football, left frame 1 , right frame 20: matching of the region of interest (head of the player) is difficult as deformations are important, several algorithms fail*

We compare 8 similarity measures: 2 strict geometric measures, SSD, SAD; 2 strict geometric measure with uncertainties on appearance (entropy of the residual (6.8)) Pz-H (entropy of the residual estimated with Parzen) and kNN-H (entropy of the residual estimated with kNN); 2 non-geometric methods Pz-KL (Kullback-Leibler divergence between radiometric PDFs estimated with Parzen) and kNN-KL (Kullback-Leibler divergence between radiometric PDFs estimated with kNN); 2 soft geometric methods with uncertainties on appearance and position Pz-KL-G (Kullback-Leibler divergence between joint radiometric and geometric PDFs estimated with Parzen) kNN-KL-G, (Kullback-Leibler divergence between joint radiometric and geometric PDFs estimated with kNN).

The distance between the reference ROI in frame 1 and candidate regions in frame 2 on Fig. 6.7 resp., in frame 20 on Fig. 6.8 was computed as a function of the translation parameters for these 8 similarity measures.

We plot the criteria value in a dashed-box 12×12 window around the true position. The black spot at the center of the plane represents the correct motion, the two first measures SSD and SAD are not convex around the solution. The two measures based on entropy Pz-H and kNN-H are very effective on small deformations (Fig. 6.7), measures with no geometry Pz-KL and kNN-KL are not enough discriminative around the solution. Finally the soft geometric methods are performing well. On larger deformations Fig. 6.8, when the head is turning and motion blur corrupts the transformation model, strict geometry measures are either biased or lead to highly non convex functionals. Soft geometric measures Pz-KL-G and kNN-KL-G perform well on both small and large deformations and seem strictly convex in a large window around the minimum. This property is interesting for the convergence of optimization algorithms (diamond search or gradient descent in our case). Moreover, one can already see that soft geometric measures estimated with Parzen, Pz-KL-G, are more “flat” around the minimum, due to PDF oversmoothing, and the solution is less accurate than with kNN-KL-G.

As a conclusion, strict geometric methods based on entropy of the residual Pz-H and kNN-H are adapted when deformations w.r.t the transformation model are small. In fact we will use these measures when estimating motion between two consecutive frames, for instance in optical flow Chapter 7. Nevertheless, in a tracking application where large deformations are possible between two templates, we will use soft geometric distances estimated accurately with kNN estimate (kNN-KL-G).

6.5.2 Tracking algorithm setup

The proposed kNN-based algorithm presented in Section 6.4 will be referred to as kNN-KL-G where KL stands for Kullback-Leibler and G stands for geometry. It was compared to four other trackers: *(i)* a geometry-free version of the proposed method (kNN-KL), *(ii)* a version of the proposed method where the kNN expression (6.4) of the divergence was replaced with an estimation based on Parzen windowing[5] (Pz-KL-G), *(iii)* an SAD version of the algorithm described in Section 6.4.1 (*i.e.*, replacing the Kullback-Leibler divergence in step 4(b) by energy (6.7)), and *(vi)* a Mean-Shift-based tracker whose implementation is publicly available [CZT05].

Note that in these comparisons, we focused on the pros and cons of the different similarity measures and their approximations. To try to avoid "corruption" of the results by other methodological aspects, we kept the tracking algorithm simple, purposely setting aside improvements such as reference update and motion prediction. Moreover, for a fair comparison between all these methods, the experimental setup of the above-mentioned Mean-Shift implementation was followed, namely, a rectangular ROI Ω (see Figs. 6.11, 6.12, and 6.9 for the dimensions) and a translation only motion φ (*i.e.*, $\lambda = \{1\}$) with a pixel resolution. The chosen radiometric space was YUV simply because the standard test sequences used in our experiments are available in this color space.

For the kNN-based methods, parameter k in measure $\mathfrak{D}_{\mathrm{KL}}(T_\varphi, R)$ (see (6.4)) was chosen equal to 3, which satisfies the conditions mentioned at Chapter 5. An experimental study of the stability of the methods with respect to this parameter is presented in Section 6.5.3. The distance $\rho_k(s)$ to the k-th nearest neighbor of s was defined in the classical Euclidean sense. For its computation, we used an implementation publicly available [Gt].

The components of the feature vectors were normalized as follows: Y, U, and V were rescaled into the interval $[0, 1]$ and the coordinates (x, y) were rescaled into $[-1, 1]$, both in the ROI and the candidate regions, the origin being located at the center of mass of the region. This latter normalization was made before applying the scaling factor α to the coordinates, meaning that (x_T, y_T) actually belongs to the interval $[-1/\alpha, 1/\alpha]$.

Finally, the minimization in $\varphi = (\alpha, u, v)$ was performed by a series of minimizations at α fixed (see Section 6.4.3) implemented using a suboptimal search procedure known as the diamond search [ZM00]. The size of the search window was empirically set to $[-12, 12]^2$. Tracking was performed with I_{ref} being fixed equal to, say, I_1 while I_{tgt} was successively equal to I_t, $t = 2, 3, 4 \ldots$ When searching for the ROI in frame I_t, the search window was centered around the position of the ROI computed in frame I_{t-1}.

[5] This Kullback-Leibler implementation is publicly available [Ihl].

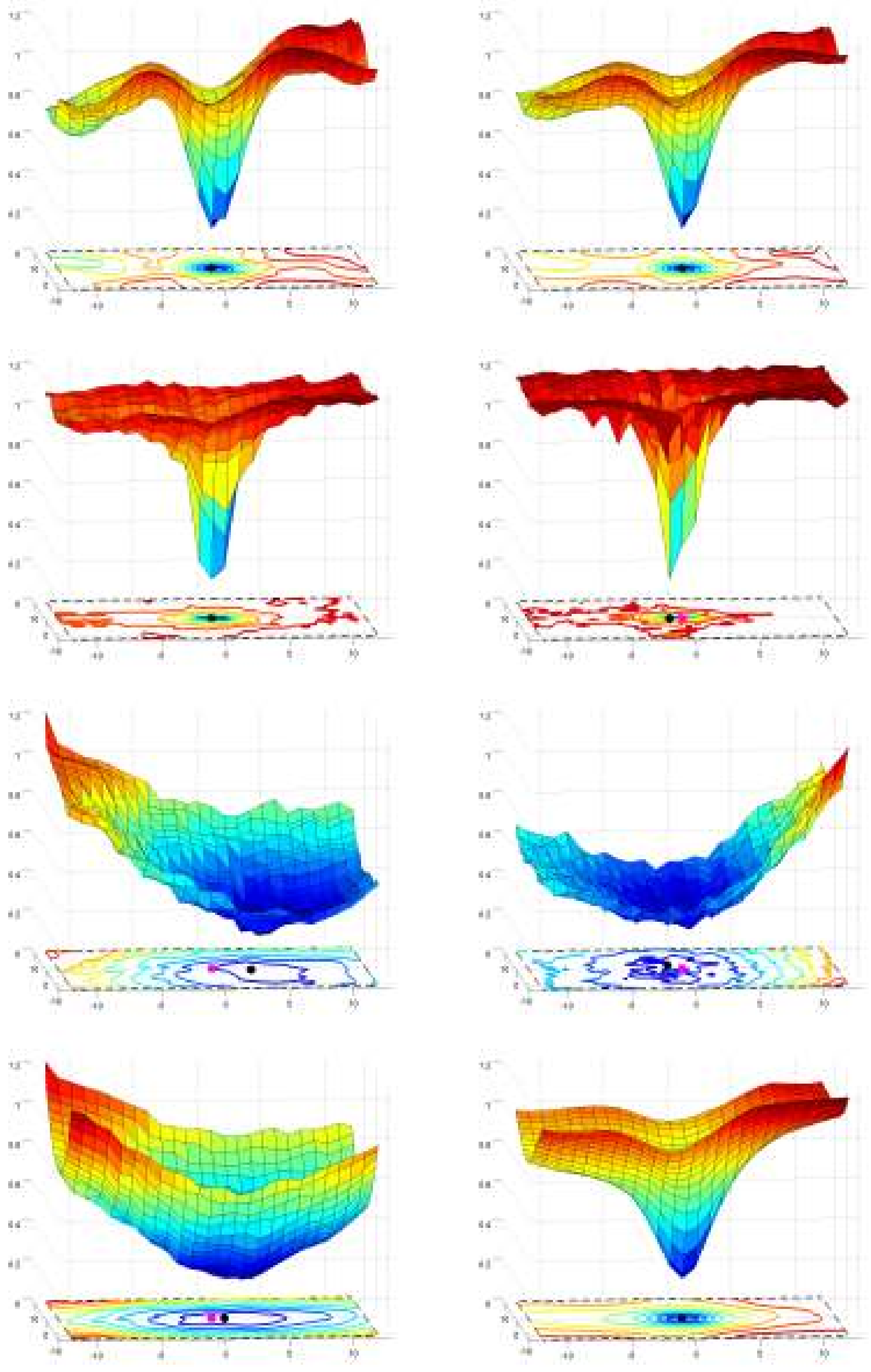


Figure 6.7: *Distance between the reference ROI of sequence "Football" and candidate regions in frame 2 as a function of horizontal and vertical translations: from left to right, top to bottom: (4 strict geometric methods) SSD, SAD, Pz-H, kNN-H (2 no geometric methods) Pz-KL, Pz-kNN (2 soft geometric methods) Pz-KL-Pz, kNN-KL-G (proposed method). The search is a* 12×12*-square (same size as the search window). The black spot at its center represents the correct translation. The purple spot represents the actual minimum of the criterions. The deformations between frame 1 and frame 2 are small, thus strict and soft geometric methods are able to find the correct minimum.*

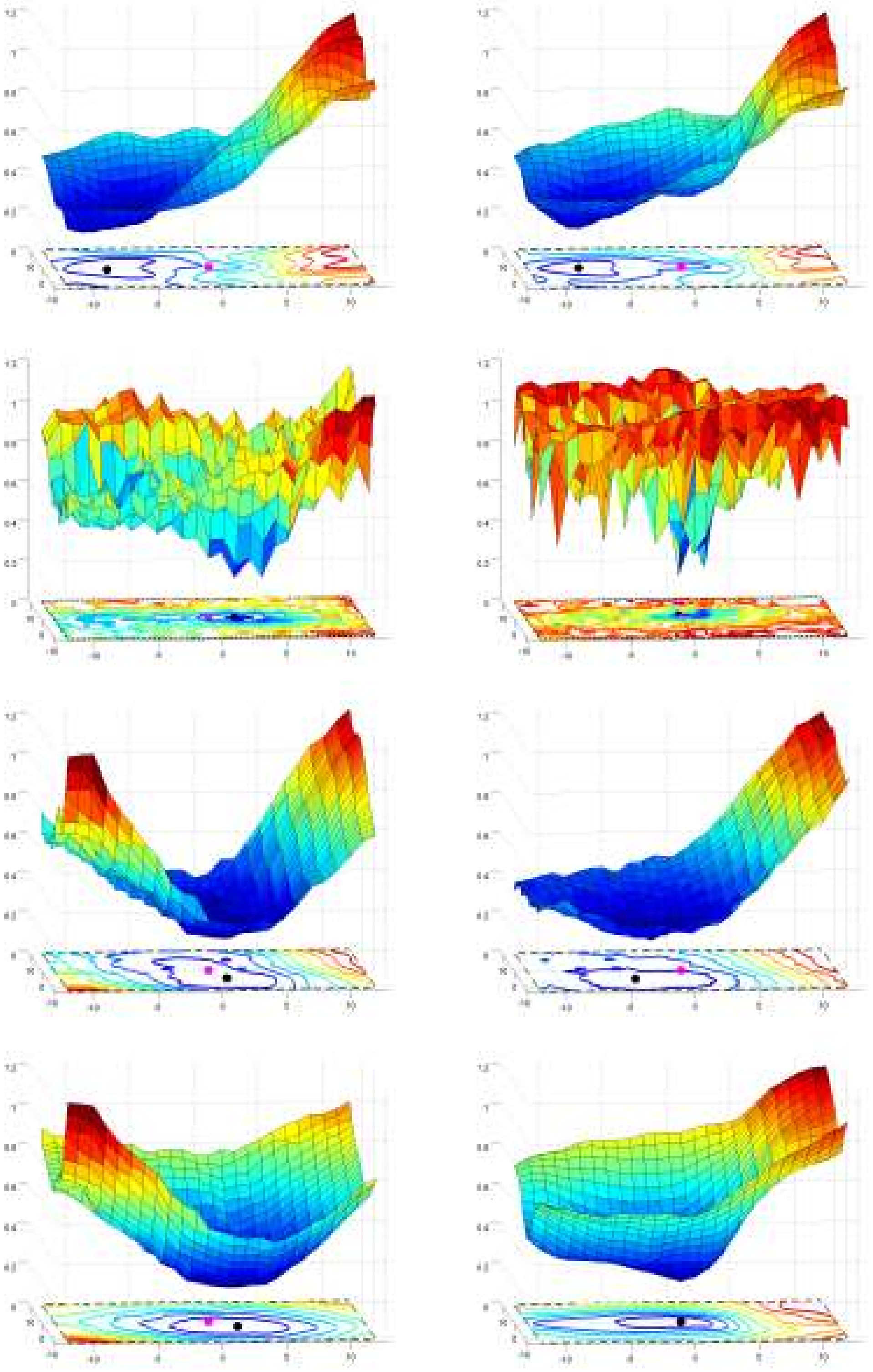
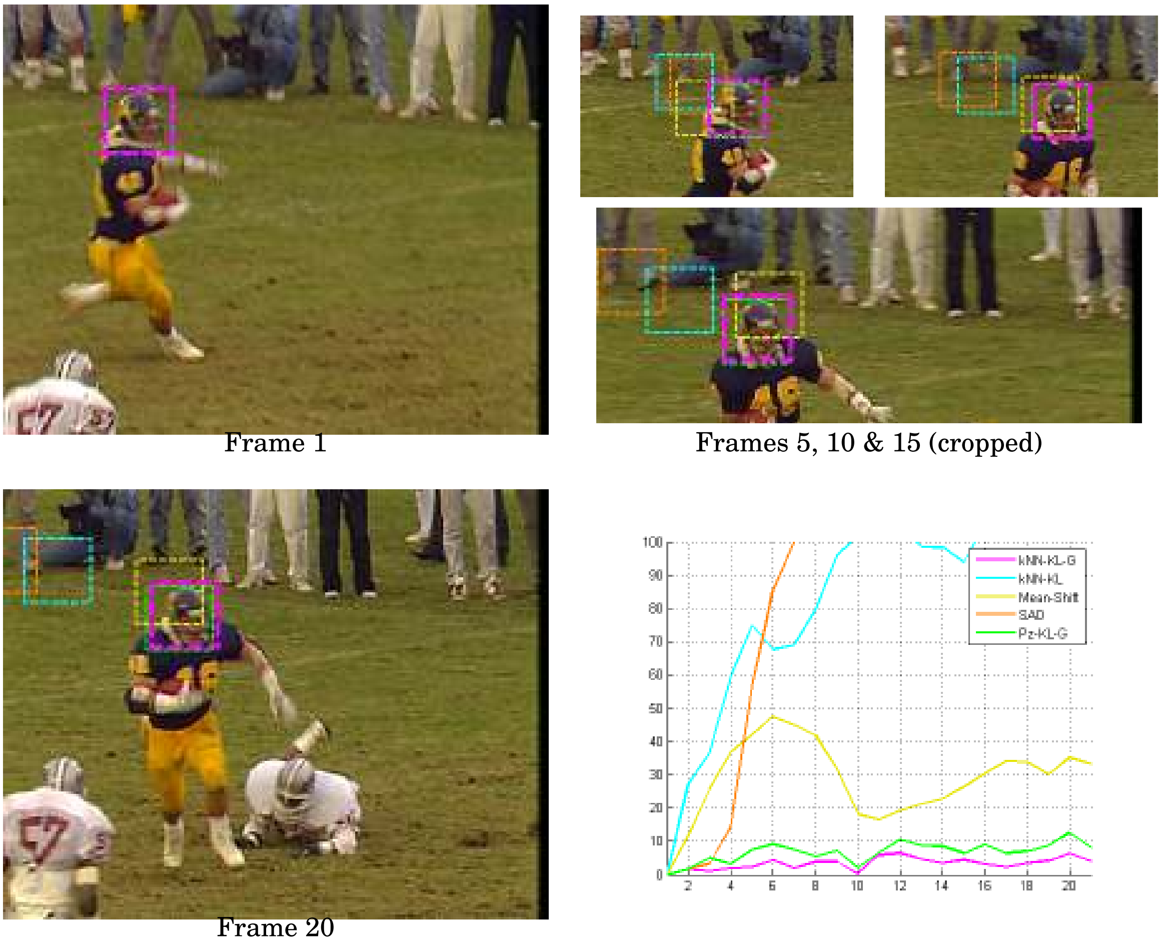


Figure 6.8: *Distance between the reference ROI of sequence "Football" and candidate regions in frame 20 as a function of horizontal and vertical translations: from left to right, top to bottom: (4 strict geometric methods) SSD, SAD, Pz-H, kNN-H (2 no geometric methods) Pz-KL, Pz-kNN (2 soft geometric methods) Pz-KL-Pz, kNN-KL-G (proposed method). The search is a* 12×12*-square (same size as the search window). The black spot at its center represents the correct translation. The purple spot represents the actual minimum of the criterions. The deformations between frame 1 and frame 20 are large, thus only soft geometric method with accurate kNN estimation of the divergence, kNN-KL-G, is able to find the correct minimum.*



Frame 1

Frames 5, 10 & 15 (cropped)

Frame 20

Figure 6.9: *Tracking on sequence "Football": frames 1, 5, 10, 15 and 20 (relative to the reference frame). kNN-KL-G (proposed method): pink; kNN-KL: cyan; Pz-KL-G: green; Mean-Shift: yellow; SAD: orange. This sequence is characterized by a fast motion generating motion blur. Moreover, the motion of the object of interest has a rotational component responsible for the disappearance of some areas and the exposure of others. The diagram represents the shift (in percent of the ROI diagonal) with respect to a manually defined tracking as a function of the frame index.* Ω*:* 43×43*-square.*

Complex motion

Sequence "Football" is composed of 352×288-frames. Tracking was performed on 20 consecutive frames (see Fig. 6.9). Note that part of the public has colors similar to colors that can be found in the ROI. In some frames, this area of the public is right above the ROI. This is probably the reason why kNN-KL stayed stuck in this region. Moreover, as the player runs, he turns and almost faces the camera toward the end of the sequence. Therefore, the translation model is not appropriate. This can explain why SAD, which relies on a strict translation model, lost the ROI in the first frames. Mean-Shift succeeded to track the ROI approximately. However, it could not avoid being attracted by the public. The geometric constraint of kNN-KL-G and Pz-KL-G allowed to avoid being attracted by the public area (where the color spatial arrangement is different from that of the reference ROI) while being soft enough to deal with the mismatch between the translation model and the actual motion. The resulting trackings

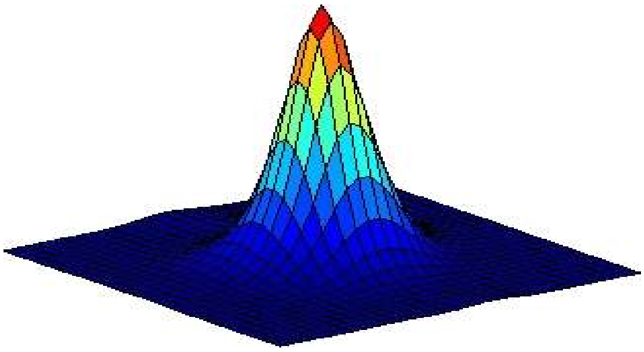


Figure 6.10: *The PDF of the pointwise motion between the reference ROI and the target ROI obtained with kNN-KL-G. For each pixel of the reference, this motion was computed as the space displacement to the nearest neighbor in the extended radiometric/geometric space among the samples of the target. The domain of definition is a* 35×35*-square (to be compared with* Ω*, a* 43×43*-square) centered around the null translation.*

Table 6.2: *Stability of kNN-KL-G with respect to* k*.*

Value of k	3	10	20	43= $\sqrt{\lvert\Omega\rvert}$
Average tracking shift	Reference	0.20 pixel	0.73 pixel	1.14 pixel

are accurate. (Nevertheless, kNN-KL-G performed better than Pz-KL-G, arguably because it relies on variable kernel bandwidth.)

Fig. 6.10 represents the PDF of the pointwise motion between the reference ROI and the target ROI obtained with kNN-KL-G. For each pixel of the reference, this motion was computed as the space displacement (*i.e.*, the distance after projection onto the geometric subspace) to the nearest neighbor in the extended radiometric/geometric space among the samples of the target. The PDF is not a Dirac delta function, illustrating the fact that the translation model was not correct.

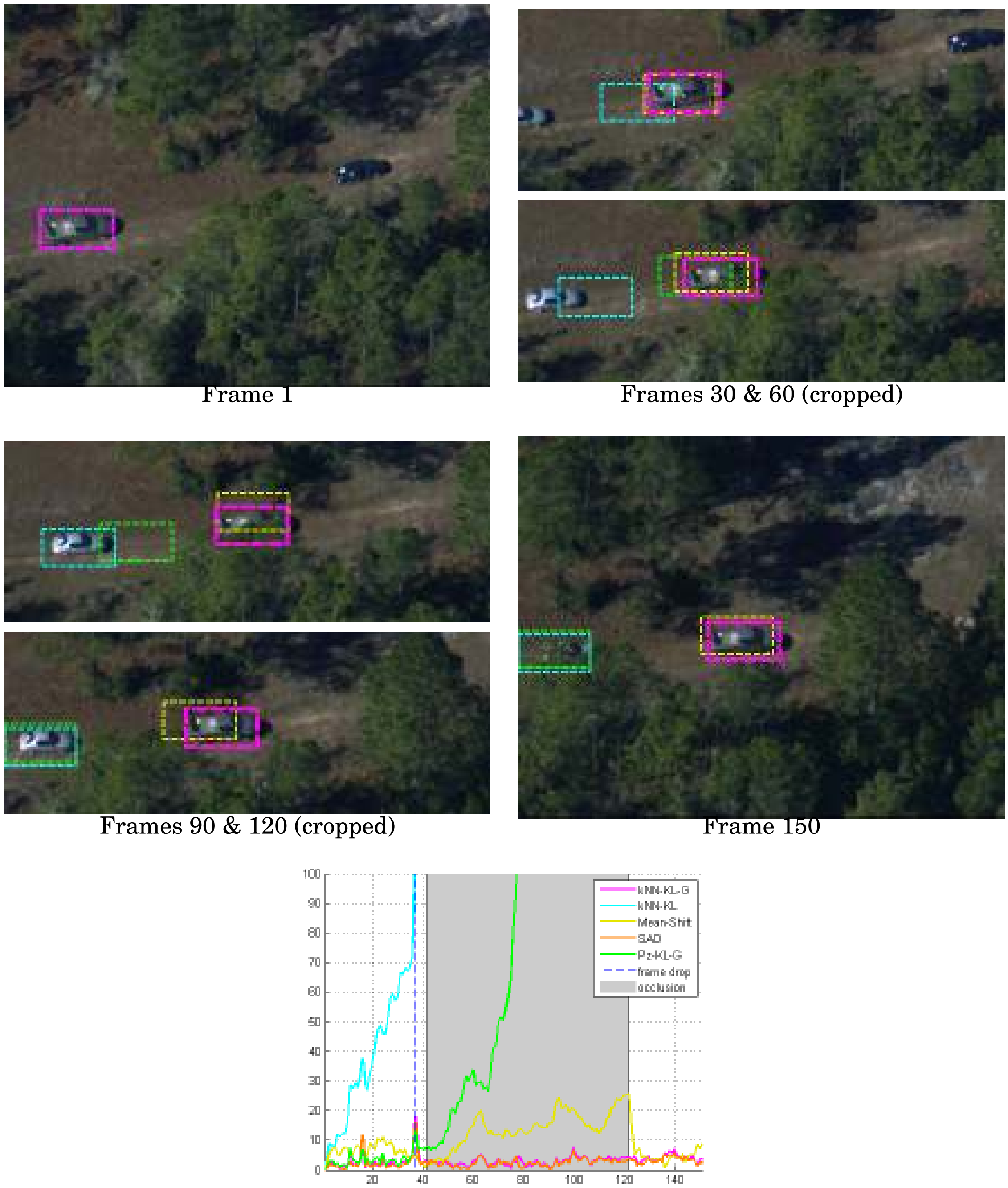
6.5.3 Stability with respect to k

To evaluate the stability of kNN-KL-G with respect to the choice of parameter k, tracking was performed on sequence “Football” with various values of k that comply with the conditions mentioned in Chapter 5. The tracking obtained for k equal to 3 was taken as a reference and the average shifts over the 20 frames resulting from using other values were measured (see Table 6.2). Therefore, the method appears quite stable with respect to k, which confirms the toy experiment in Appendix Table A.1.

6.5.4 Robustness to difficulties

Partial occlusions

Sequence “Car” is an aerial car chase which is part of the VIVID tracking testbed [CZT05]. It is composed of 640×480-frames. Tracking was performed on 150 consecutive frames (see Fig. 6.11). kNN-KL eventually lost the ROI and ended up tracking the second car which has colors similar to the ROI. This is probably due to the fact that it is based on radiometry only. Pz-KL-G also failed in tracking the first car. Mean-Shift performed quite well although


Frame 1

Frames 30 & 60 (cropped)

Frames 90 & 120 (cropped)

Frame 150

Figure 6.11: *Tracking on sequence "Car": frames 1, 30, 60, 90, 120 and 150 (relative to the reference frame). kNN-KL-G (proposed method): pink; kNN-KL: cyan; Pz-KL-G: green; Mean-Shift: yellow; SAD: orange. There is a frame drop of several frames at frame 38 (vertical dashed line in the diagram) and the tracked car is partially occulted by trees from frame 42 to frame 122 (gray area in the diagram). The diagram represents the shift (in percent of the ROI diagonal) with respect to a manually defined tracking as a function of the frame index.* Ω*:* 95×47*-rectangle.*

the tracking shifted upward when occlusion occurred in order to avoid including the green colors of the trees in the color PDF. Concerning SAD, the translation model being fairly well respected within the ROI, taking the pointwise residual makes sense while the use of the absolute value is robust to the outliers arising from the occlusion. As a consequence, the car was accurately tracked. Finally, kNN-KL-G also performed very well.

Variations of luminance

Sequence “Crew” is composed of 352×288-frames. Two faces were tracked on 80 consecutive frames (see Fig. 6.12). kNN-KL-G tracked the faces successfully. The other methods lost progressively the ROI, probably because of the variations of luminance.

Noise

Sequence “Schnee” is composed of 352×288-frames. Two cars were tracked on 160 consecutive frames (see Fig. 6.13).

6.5.5 Scale estimation

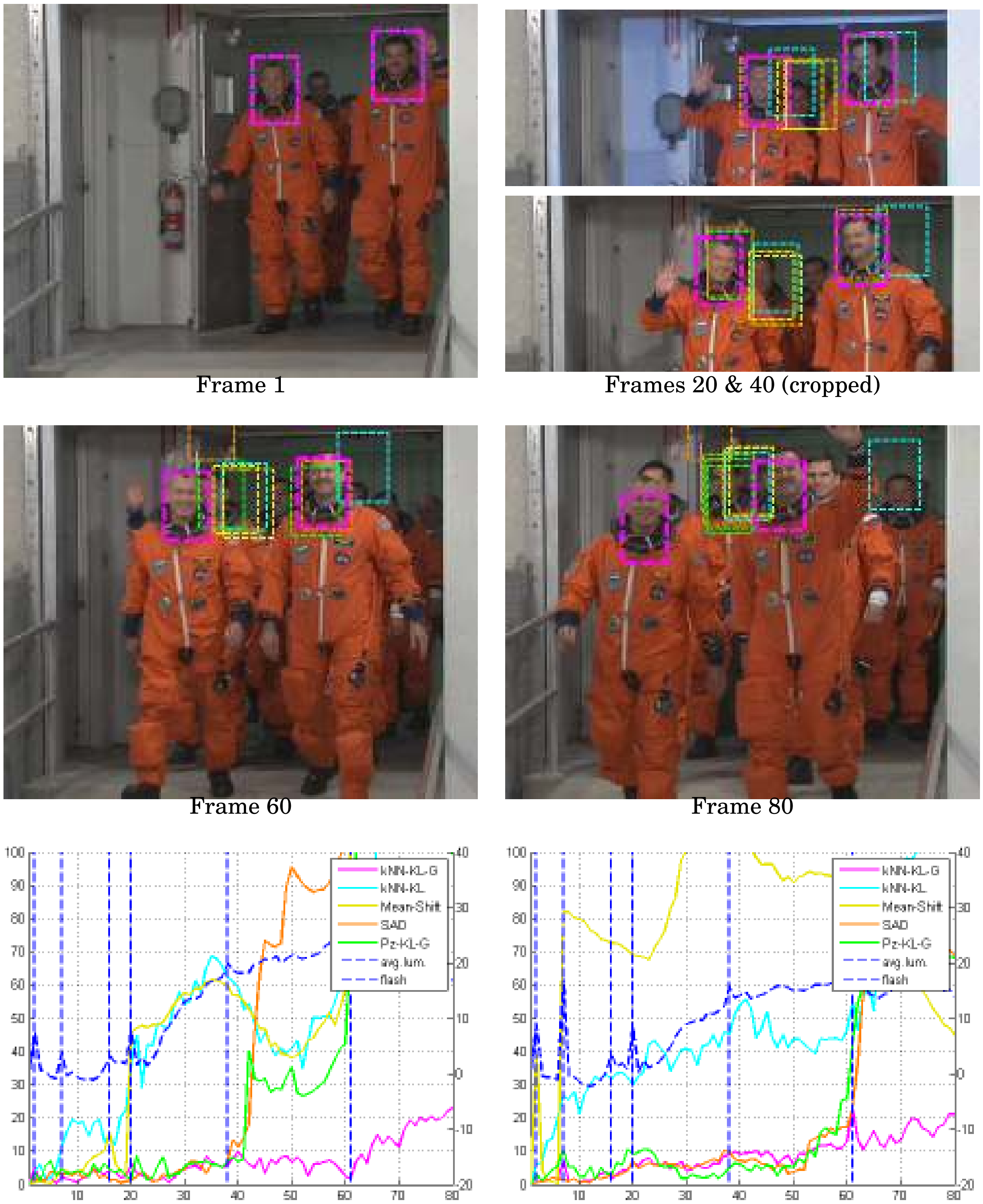
Sequence “WaterObject” is composed of 352×288-frames. Tracking was performed on 95 consecutive frames. For this sequence, the test set λ of scaling factors was chosen equal to $\{0.98, 0.99, 1, 1.01, 1.02\}$ (see Fig. 6.14).

6.6 Conclusion

This chapter presents a general framework for estimating high-dimensional statistical measures to perform ROI tracking. We focused on a measure derived from entropy with consistency and unbiasedness estimator [GLMI05, KL87].

In term of comparison with other approaches, the proposed method can be characterized by such keywords as statistical, non-parametric, variable kernel bandwidth (kNN), joint color and geometry processing, and soft geometric constraint. *(i)* SAD, or similar non-robust and robust similarity measures, is deterministic in essence although it corresponds to solving the tracking problem with a parametric assumption on the residual PDF. The strict geometrical constraint does not allow much tolerance regarding motion model mismatch and the parametric PDF assumption prevents data fitting. *(ii)* kNN-KL can adapt to the data thanks to its non-parametric nature and the use of a variable kernel bandwidth. Because of its statistical point of view, it can account for some color variability of the ROI. Unfortunately, as it is well known, the absence of geometric constraint is a serious penalty. *(iii)* Pz-KL-G does include a soft geometrical constraint. However, the approximation of a PDF-based measure using a fixed kernel bandwidth, *i.e.*, without adjustment to the local density of the samples, is a weakness, as is clear from the experimental results. *(iv)* The Mean-Shift-based tracker used in the comparisons [CRM00, CZT05] rely on another statistical measure: the Bhattacharya measure. Whether the differences observed between this tracker and the proposed method in the experimental results presented here depends on the measure itself or on the way geometry is involved[6] is unclear. Finally, *(v)* to a certain extent, the proposed method seems

[6] A Gaussian weighting of the features according to their distance to the center of the ROI (which can be seen as a radial layout constraint) for the Mean-Shift-based tracker versus a joint radiometric/geometric processing for kNN-KL-G.



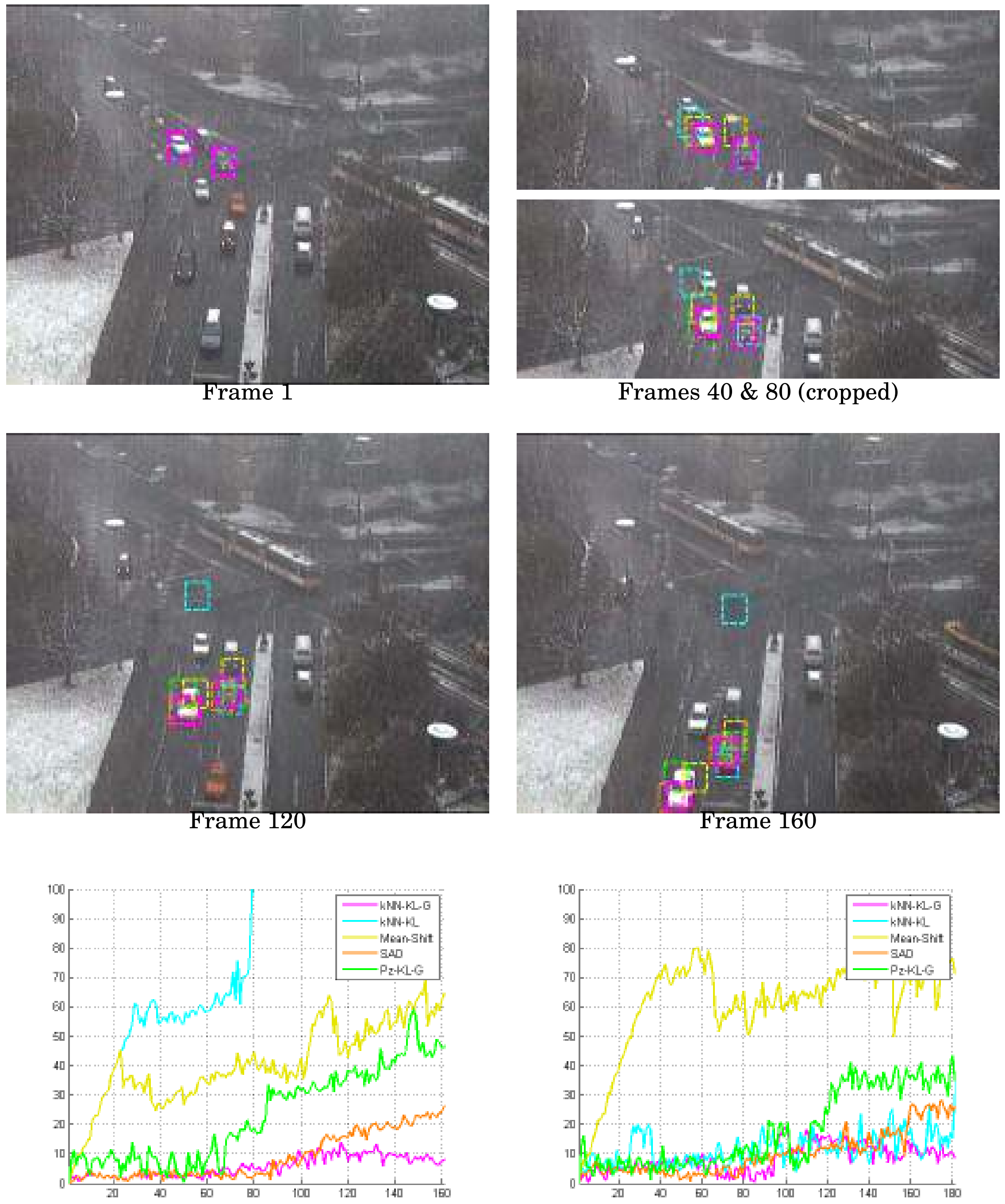
Frame 1

Frames 20 & 40 (cropped)

Frame 60

Frame 80

Figure 6.12: *Tracking on sequence "Crew": frames 1, 20, 40, 60 and 80 (relative to the reference frame). kNN-KL-G (proposed method): pink; kNN-KL: cyan; Pz-KL-G: green; Mean-Shift: yellow; SAD: orange. There are two kinds of intensity changes in the sequence: a slight, continuous intensity increase as the crew walks out of a dark area, and some strong and brief intensity peaks due to camera flashes (vertical dashed lines in the diagrams). The diagrams represent the shift (in percent of the ROI diagonal) with respect to manually defined trackings as a function of the frame index. The diagram on the left corresponds to the face on the left. The vertical axis on the right of each diagram corresponds to the blue dashed curves which represent the evolution of the average intensity (Y component) within the manually defined trackings. The average intensity in frame 1 is taken as the reference and the scale is in unit of intensity. Both the continuous intensity increase and the camera flashes are noticeable.* Ω*:* 33×52*-rectangle.*



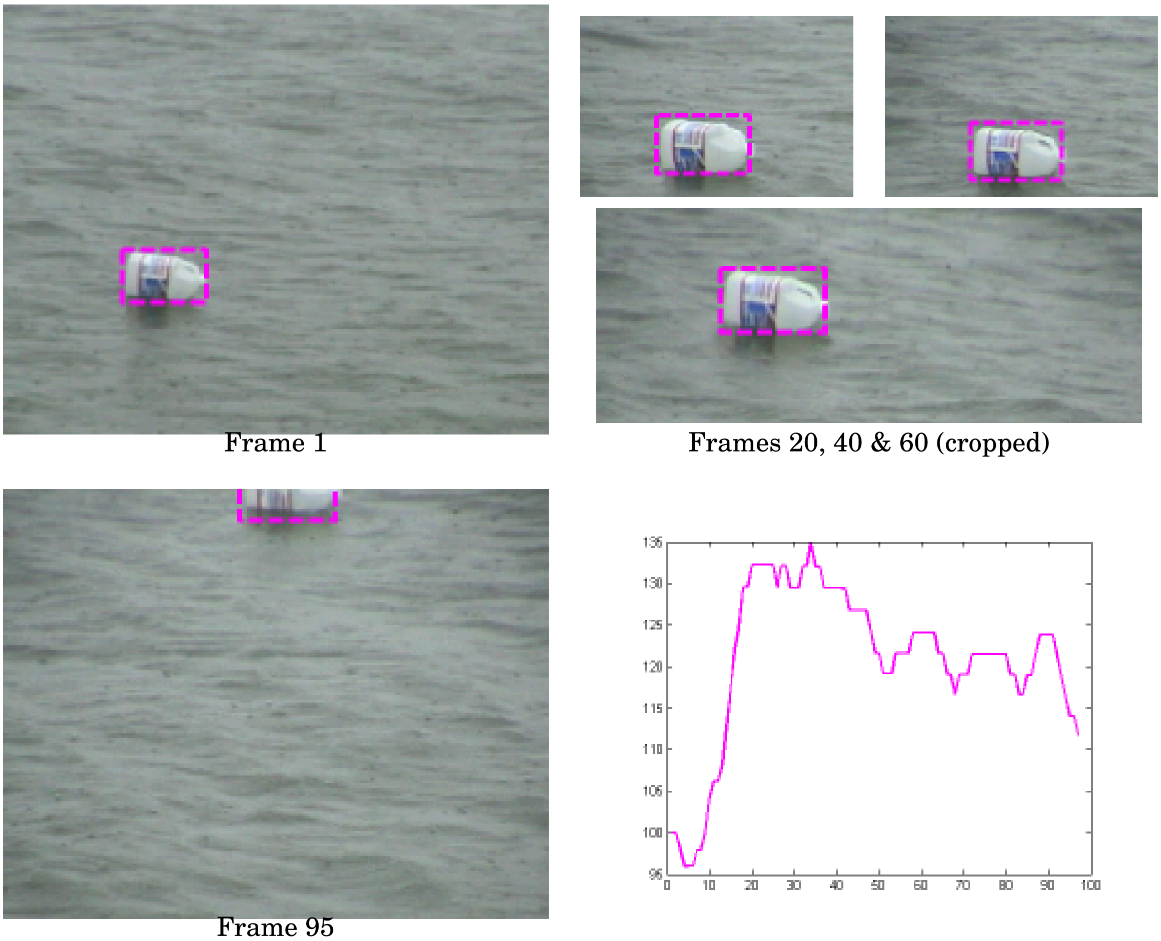
Frame 1

Frames 40 & 80 (cropped)

Frame 120

Frame 160

Figure 6.13: *Tracking on sequence “Schnee”: frames 1, 40, 80, 120 and 160 (relative to the reference frame). kNN-KL-G (proposed method): pink; kNN-KL: cyan; Pz-KL-G: green; Mean-Shift: yellow; SAD: orange. This sequence can be considered noisy due to the snow flakes. The diagrams represent the shift (in percent of the ROI diagonal) with respect to manually defined trackings as a function of the frame index. The diagram on the left corresponds to the car on the left. Ω: a 38×42-square for the car on the left and a 34×42-square for the car on the right.*


Frame 1

Frames 20, 40 & 60 (cropped)

Frame 95

Figure 6.14: *Tracking on sequence “WaterObject”: frames 1, 20, 40, 60 and 95 (relative to the reference frame). This sequence is characterized by zooms in and out. The diagram represents the scaling of the ROI (parameter α in percent of the initial size) as a function of the frame index. Ω: initially, a 48×28-square.*

to provide answers to the problems previously mentioned, either theoretically, in practice, or both.

CHAPTER 7

OPTICAL FLOW

This chapter will define a new general statistical framework for optical flow. First, a measure issued from information theory will be used as data and regularization terms. Second, data and regularization term will be high dimensional and nonlocal to accurately model the resulting optical flow. Third, these high dimensional measures will be estimated with the kNN framework. Finally, the diffusion process takes advantage of the simple derivative approximation of the kNN framework and will end to meaningful equations in connections with classical methods of state of the art. This chapter follows the general introduction written in Section 2.2.

7.1 Introduction

The contribution of this chapter is a general statistical framework for optical flow computations. This statistical framework is expressed in a balanced manner as a sum of two different entropies, one on the residual, named data term and the other one on the flow field, named regularization term. Entropy is a function of the distribution of the data as opposed to a (robust) function of the data itself. Entropy is estimated non-parametrically to handle outliers. These outliers can be for instance occlusions or variations of illuminance on the data term, or motion discontinuities at objects boundaries on the regularization term. In the case where parametric assumptions are too strong, nonparametric techniques (*e.g.*: entropy minimization) performs better as it adapts to underlying data statistics. Moreover, recent works in optical flow [BBPW04] and also in segmentation [RBD03] tend to show that the more information you have, the more accurate will be your result. While optical flow methods in the literature often consider grayscale images, the use of other features like color, gradient [BBPW04], or structure tensor [RBD03] improves the accuracy of the results. These features are often treated independently and added with weightings between each components. In our framework, vector features naturally extend to multivariate data into a joint entropy.

The minimization of the framework composed of two entropies will result in a nonlocal coupled diffusion process for the data and regularization term. Each term taken independently follows the same philosophy as nonlocal diffusions for image denoising [Yar85, AW06].

Finally, while using nonparametric techniques often enhance the quality of the results in various applications (*e.g.* segmentation [RBD03], tracking [CRM00, BDB07]) they often increase both mathematical and computational complexity. Here, we show that using mean-shift simplifications [FH75], we obtain simple equations where the nonlinearities can be easily isolated and estimated with fixed point iterations. The solution is then obtained with an efficient linear iterated solver.

This chapter is organized as follows. The classical formulation of optical flow and its recent improvements are presented in Section 7.2. Section 7.3 introduces our unified entropy-based nonparametric framework for optical flow. Section 7.4 gives a numerical efficient solution to minimize the cost functional. Section 7.5 validates the method on an optical flow benchmark. Section 7.6 concludes.

7.2 Optical flow formulation

7.2.1 Classical methods

Classically, optical flow v can be retrieved by minimizing the following functional

$$E(\mathrm{v}) = E_{\mathrm{Data}}(\mathrm{v}) + \alpha E_{\mathrm{Smooth}}(\mathrm{v}). \tag{7.1}$$

where E_{Data} is a data term on the image domain D,

$$E_{\mathrm{Data}}(\mathrm{v}) = \int_D \Psi(\epsilon^2(\mathrm{m}, \mathrm{v}(\mathrm{m})))\,\mathrm{dm} \tag{7.2}$$

where Ψ is a non quadratic robust penalizer, ϵ is a residual function and will be the first order Taylor expansion of the image residual,

$$\epsilon_{\mathrm{m}} = \epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m})) = u_x(\mathrm{m})\mathrm{v}_1(\mathrm{m}) + u_y(\mathrm{m})\mathrm{v}_2(\mathrm{m}) + u_t(\mathrm{m}) \tag{7.3}$$

where u_x u_y and u_t represent the spatial and temporal partial derivatives of the features extracted from the image u (image luminance, color values, ...), the notation ϵ_{m} will be preferred to $\epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m}))$ in order to simplify notations.

E_{Smooth} adds a regularization penalty in the cost functional.

$$E_{\mathrm{Smooth}}(\mathrm{v}) = \int_D \Psi(|\nabla \mathrm{v}_1(\mathrm{m})|^2 + |\nabla \mathrm{v}_2(\mathrm{m})|^2)\,\mathrm{dm} \tag{7.4}$$

where Ψ is again a non quadratic robust penaliser, v_1, resp. v_2 are the horizontal and vertical components of the optical flow.

7.2.2 Improvements

On the data term, two kinds of methods have recently enhanced the quality of optical flow estimation.

The first one [BWS05] is based on constraining motion to be constant within a small neighborhood as in [LK81] but extended into a global functional. They minimize the quadratic form $\mathrm{w}^T(\mathrm{m}).J_\sigma(\nabla_3 I(\mathrm{m})).\mathrm{w}(\mathrm{m})$ where $\mathrm{w}(\mathrm{m}) = [\mathrm{v}(\mathrm{m})\, 1]$, $J_\sigma(\nabla I) = G_\sigma * (\nabla_3 I\, \nabla_3 I^T)$ is the structure tensor [BWS05], with G_σ a Gaussian kernel.

The second one [BBPW04] is to combine other features than just brightness consistency. The authors have added gradient consistency. The image features u are now multi dimensional $u = [I, \nabla I]$ and the different dimensions in the data energy (7.2) are added using different weightings.

Other works [RB05] modified the regularization term. They studied the statistics of optical flow fields and they deduced a prior that captures the rich statistics of optical flow patches (3×3 or 5×5).

7.2.3 Limitations

The data term is local: classical methods consider the residual seen at a pixel level or coherent within a small neighborhood. Actually, the residual is assumed to follow a Laplacian distribution as $\Psi(x^2)$ generally smoothly approximates the absolute value function [WS01, BWS05]. No global statistics are used on the residual to model specific residual distributions due to outliers such as occlusions or variations of illuminance.

The regularization term is also local: looking at classical anisotropic diffusion, the diffusion term is computed in a small neighborhood. The gradient of the flow is again assumed to follow a Laplacian distribution [WS01, BWS05]. A recent study on statistics of optical flow fields modeled more accurately the distribution of the flow [RB05]. Again, no global statistics on the flow are estimated to model specific flow distributions due to difficulties such as motion boundaries or non-translational flow fields.

In this chapter we propose a unified statistical framework to estimate and use global statistics (estimated with nonparametric techniques) of the data and regularization term. This framework allows to estimate nonlocal statistics of the flow and the residual, naturally adapts to multivariate features in a balanced functional (as opposed to techniques summing energies of different physical quantities). Finally, using entropy as a criterion naturally adapts to variations of illuminance and isolates outliers into different modes of the PDF. Next section will detail this framework.

7.3 A unified statistical framework

The global energy is defined as a sum of entropies, where global statistics are learnt on the whole image, as underlying PDFs are estimated nonparametrically.

7.3.1 General formulation

The new cost function is then a sum of two entropies:

$$E(\mathrm{v}) = E_{\mathrm{H-Data}}(\mathrm{v}) + E_{\mathrm{H-Smoooth}}(\mathrm{v}) \tag{7.5}$$

where $E_{\mathrm{H-Data}}$ is an entropy function of a data term, and $E_{\mathrm{H-Smooth}}$ is an entropy function of the vector field.

More precisely, the energy chosen is based an empirical estimate of entropy [AL76] over some observations on the image domain $t_D = \{t_{\mathrm{m}}, \mid \mathrm{m} \in D\}$

$$\begin{aligned} H(t_D) &= -E_t[\log f_t] & (7.6) \\ &= -\frac{1}{|D|} \sum_{\mathrm{m} \in D} \log f_t(t_{\mathrm{m}}). & (7.7) \end{aligned}$$

One can note that as the image size $|D|$ is constant, it will play no role in the energy or its minimization. Thus, the Ahmad-Lin entropy will be multiplied by $|D|$.

7.3.2 Entropy as a data term

We propose for $E_{\mathrm{H-Data}}$ an entropy of the residual ϵ. The data term is based an entropy on the residual ϵ (7.3) considered as a random observation.

$$E_{\mathrm{H-Data}}(\mathrm{v}) = |D|.H(\epsilon_D) = -\sum_{\mathrm{m}\in D} \log f_\epsilon(\epsilon_{\mathrm{m}}) \tag{7.8}$$

where $\epsilon_D = \{\epsilon_{\mathrm{m}} \mid \mathrm{m} \in D\}$, f_ϵ is a PDF estimated non parametrically using kernel techniques. Let us remind that ϵ_{m} is a short notation for $\epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m}))$ and is function of v.

A kernel estimate of f_ϵ can be written as

$$f_\epsilon(\epsilon_{\mathrm{m}}) = \frac{1}{|D|} \sum_{\mathrm{n}\in D} K_h(\epsilon_{\mathrm{m}} - \epsilon_{\mathrm{n}}) \tag{7.9}$$

where K_h is a kernel of bandwidth h. The choice of h is not an easy problem as discussed in this thesis.

ϵ is one dimensional if the constraint is computed only on brightness consistency but naturally extends to a vector when adding other features such as color or gradient consistency. Indeed we choose as features u in ϵ (7.3) a combination of color components and spatial structure tensor features as proposed in a segmentation application [RBD03]. The entropy $E_{\mathrm{H-Data}}$ will then be a joint entropy of dimension $d_\epsilon = 6$ between residual of several features: 3 for colors, 3 for the different components of the symmetric 2×2 color tensor $J_\sigma(\nabla I) = G_\sigma * (\nabla_2 I \, \nabla_2 I^T)$. The features u in (7.3) are 6-dimensional $u = [I, J_\sigma(I)]$

Finally, the energy $E_{\mathrm{H-Data}}(\mathrm{v})$ accounts for the true distribution of the residual ϵ, it is proposed to make the energy depend on an estimation f_ϵ of the residual distribution rather than on the residual itself as in (7.2). Using entropy of the residual introduces uncertainty on the appearance of each pixels, it was deeper motivated theoretically in Section 6.3.2 and experimentally in Appendix B.

7.3.3 Entropy as a regularization term

The regularization term is based on an entropy of the vector field. We present entropy of the vector field as regularizer.

$$E_{\mathrm{H-Smooth-Global}}(\mathrm{v}) = |D|.H(\mathrm{v}(D)) - \sum_{\mathrm{m}\in D} \ln f_{\mathrm{v}}(\mathrm{v}(\mathrm{m})) \tag{7.10}$$

where $\mathrm{v}(D) = \{\mathrm{v}(\mathrm{m}) \mid \mathrm{m} \in D\}$. A kernel estimate of f_{v} can be written as

$$f_{\mathrm{v}}(\mathrm{v}(\mathrm{m})) = \frac{1}{|D|} \sum_{\mathrm{n}\in D} K_h(\mathrm{v}(\mathrm{m}) - \mathrm{v}(\mathrm{n})) \tag{7.11}$$

Let us explain it with an illustration on a discrete 1-D case on Fig. 7.1. A classical regularization is the total variation based on SAD:

$$E_{\mathrm{SAD}}(\mathrm{v}) = \sum_{\mathrm{m}\in D} |\nabla \mathrm{v}(\mathrm{m})| \tag{7.12}$$

First let us compare a piecewise constant versus a piecewise smooth function, SAD regularization favors piecewise constant functions, often called on images “cartoon effect” of total variation. Entropy regularization follows the same behavior: entropy favors piecewise constant images. An idea of an entropic regularizer which does not favor piecewise constant images will be given in perspectives Chapter 10. Regarding regularization on an edge, SAD does not favor smooth or sharp contours as the total variation is the same, while entropy favors sharp contours.

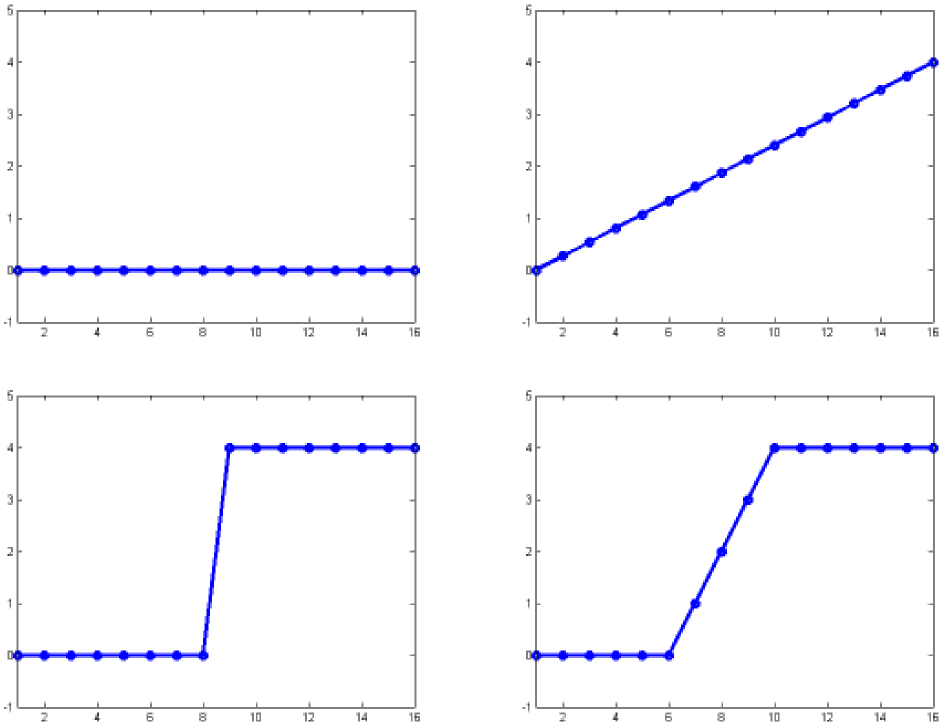


Figure 7.1: *Illustration of regularization: a discrete 1-D example, from left to right, top to bottom: constant, linear, sharp, smooth. Score of the function* $\sum|\nabla \mathrm{v}|$*: constant* 0*, linear* 16*: cartoon effect of SAD, score of* $H(\mathrm{v})$*: constant* 0*, linear* 2.8*: cartoon effect of entropy. Score of the function* $\sum|\nabla \mathrm{v}|$*: sharp* 4*, smooth* 4*, score of* $H(\mathrm{v})$ *sharp* 0.23*, smooth* 0.56*, entropy favors sharp contours*

However, while the distribution of the residual ϵ is considered as noise with no spatial coherence, the distribution of the vector field should have local structures and form patterns. As presented in section on new possibilities with high dimensions (Section 5.2), regularization chosen nonlocal as a diffusion in joint space of vectors and pixel positions. The different regularizations will be illustrated and compared in section on equations understanding (Section 7.4.2).

$E_{\mathrm{H-Smooth}}$ is then chosen to be a joint entropy of vector field and pixel positions.

$$E_{\mathrm{H-Smooth}}(\mathrm{v}) = |D|.H([D, \mathrm{v}(D)] - \sum_{\mathrm{m}\in D} \ln f_{\mathrm{v}}(\mathrm{m}, \mathrm{v}(\mathrm{m})) \qquad \textbf{(7.13)}$$

where $D = \{\mathrm{m}|\mathrm{m} \in D\}$ and $\mathrm{v}(D) = \{\mathrm{v}(\mathrm{m})|\mathrm{m} \in D\}$.

Finally, kernel estimate of f_{v} can be written as

$$f_{\mathrm{v}}(\mathrm{m}, \mathrm{v}(\mathrm{m})) = \frac{1}{|D|} \sum_{\mathrm{n}\in D} K_h([\mathrm{m}, \mathrm{v}(\mathrm{m})] - [\mathrm{n}, \mathrm{v}(\mathrm{n})]) \qquad \textbf{(7.14)}$$

7.3.4 Differentiation using mean-shift

In order to minimize E, energies $E_{\mathrm{H-Data}}$ (7.8) and $E_{\mathrm{H-Smooth}}$ (7.13) are differentiated with respect to the optical flow v.

The different features implied here are high-dimensional (ϵ is 6-dimensional and the joint position-vector space is 4-dimensional), thus energies are estimated within the kNN framework defined in Chapter 5. These energies are differentiable and their derivatives can be approximated with the kNN mean-shift as detailed in Section 5.5.

Details of the derivative can be found in Appendix D, simple expressions are obtained functions of k-th nearest neighbors. For the data term, the derivative writes

$$\nabla_{\mathrm{v_m}} E_{\mathrm{H-Data}}(\mathrm{v}) = \nabla_{\mathrm{v}} \epsilon_{\mathrm{m}} . \frac{d_\epsilon + 2}{d_k(\epsilon_{\mathrm{m}})^2} [\mu_B(\epsilon_{\mathrm{m}}) - \epsilon_{\mathrm{m}}]. \tag{7.15}$$

where $\mu_B(\epsilon_{\mathrm{m}})$ is the mean over a balloon neighborhood: the k-th nearest samples of ϵ_{m} in the 6-dimensional space of the whole residual of the image features u computed over the image domain $\epsilon(D)$, d_k is the distance to this k-th nearest sample, and d_ϵ is the dimension of ϵ equal to 6. $\nabla_{\mathrm{v}} \epsilon_{\mathrm{m}}$ is a $2 \times d_\epsilon$ matrix of partial derivatives: partial optical flow vector derivatives of the features in the rows and partial spatial derivatives of the features in the columns, $[\mu_B(\epsilon_{\mathrm{m}}) - \epsilon_{\mathrm{m}}]$ is a vector $d_\epsilon \times 1$, $\nabla_{\mathrm{v}} E_{\mathrm{H-Data}}(\mathrm{m})$ is then a vector 2×1. For the regularization term, the derivative writes

$$\nabla_{\mathrm{v_m}} E_{\mathrm{H-Smooth}}(\mathrm{v}) = \mathbb{P}_2 . \frac{d_{\mathrm{v}} + 2}{{d_k}^2(\mathrm{m}, \mathrm{v}(\mathrm{m}))} [\mu_B(\mathrm{m}, \mathrm{v}(\mathrm{m})) - [\mathrm{m}, \mathrm{v}(\mathrm{m})]] \tag{7.16}$$

where $\mathbb{P}_2$ is the projection operator $\mathbb{P}_2.[\mathrm{m}, \mathrm{v}(\mathrm{m})] = \mathrm{v}(\mathrm{m})$, where $\mu_B(\mathrm{m}, \mathrm{v}(\mathrm{m}))$ is the mean over a balloon neighborhood: the k-th nearest samples of $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$ in the 4-dimensional space of the optical flow combined with its position, d_{v} is the dimension of $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$ equal to 4.

7.4 Numerical Aspect

Optical flows are generally solved as solutions of linear solvers as they converge faster and with more accuracy to a minimum than classical gradient descend. This matrix is defined from $\nabla_{\mathrm{v}} E(\mathrm{v}) = \nabla_{\mathrm{v}} E_{\mathrm{H-Data}}(\mathrm{v}) + \nabla_{\mathrm{v}} E_{\mathrm{H-Smooth}}(\mathrm{v}) = 0$. Therefore, the derivative equations (7.15) and (7.16) need to be linearized. Plugging the mean-shift approximations in the derivative expressions, we have equations (7.15) and (7.16) that are linear except for the PDF estimation step (represented here by the estimation of the means μ_B). The PDF is then re-estimated with fixed point iterations. This technique is well known in optical flow community as robust functions Ψ (7.2) and (7.4) (and their derivatives) are generally nonlinear and are re-estimated using fixed point iterations [BBPW04].

$$\nabla_{\mathrm{v}} E_{\mathrm{H-Data}} + \nabla_{\mathrm{v}} E_{\mathrm{H-Smooth}} = 0 \tag{7.17}$$

$$\nabla_{\mathrm{v}} \epsilon(\mathrm{m}) . \frac{d_\epsilon + 2}{d_k(\epsilon_{\mathrm{m}})^2} [\mu_B(\epsilon_{\mathrm{m}}) - \epsilon_{\mathrm{m}}] + \mathbb{P}_2 \frac{d_{\mathrm{v}} + 2}{{d_k}^2(\mathrm{m}, \mathrm{v}(\mathrm{m}))} [\mu_B(\mathrm{m}, \mathrm{v}(\mathrm{m})) - [\mathrm{m}, \mathrm{v}(\mathrm{m})]] = 0 \tag{7.18}$$

7.4.1 Linear system solution and stability

An iterative scheme, SOR (Successive Over Relaxation), is then applied to solve the linear system (7.18). SOR has convergence proofs for diagonally dominant matrices. There is no insurance that the linear system from kNN mean-Shift expression (5.41) is diagonally dominant. However, one can note that in a kNN mean-shift expression the diagonal component has a weighting of 1 whereas the corresponding line is composed of k elements with a weighting of $1/k$ and 0 otherwise. Assuming all the features have the same values, the matrix would then be weakly diagonally dominant. In general, we will assume that the matrix is diagonally dominant for convergence purpose. Plugging the linearized equations (7.15), (7.16) of the derivatives in a SOR scheme, while keeping the neighborhoods estimations non linear, we have the following diffusion process to estimate optical flow.

$$\mathrm{v}_{1i}^{l+1} = (1-w)\mathrm{v}_{1i}^{l} + w\frac{\mathcal{R}_i\mu_B({\mathrm{v}_1}_i{}^{l,l+1}) + \mathcal{D}_i[u_x(i).\mu_B({\epsilon_i}^{l,l+1}) - u_xu_y(i).\mathrm{v}_{2i}^{l} - u_xu_t(i)]}{\mathcal{R}_i + u_x^2(i)\mathcal{D}_i} \quad (7.19)$$

$$\mathrm{v}_{2i}^{l+1} = (1-w)\mathrm{v}_{2i}^{l} + w\frac{\mathcal{R}_i\mu_B({\mathrm{v}_2}_i{}^{l,l+1}) + \mathcal{D}_i[u_y(i).\mu_B({\epsilon_i}^{l,l+1}) - u_xu_y(i).\mathrm{v}_{1i}^{l+1} - u_yu_t(i)]}{\mathcal{R}_i + u_y^2(i)\mathcal{D}_i} \quad (7.20)$$

where $\mathcal{D}_i = \frac{d_\epsilon+2}{d_k(\epsilon_i)^2}$ and $\mathcal{R}_i = \frac{d_\mathrm{v}+2}{d_k(\mathrm{v}(i))^2}$, ϵ is the residual function (7.3), where $\mu_B(\mathrm{v})$ and $\mu_B(\epsilon)$ are non local means detailed below.

$\mu_B({\epsilon_i}^{l,l+1}) = \frac{1}{k}(\sum_{\epsilon_j \in \mathcal{N}_B^-(\epsilon_i)} {\epsilon_j}^{l+1} + \sum_{\epsilon_j \in \mathcal{N}_B^+(\epsilon_i)} {\epsilon_j}^{l})$ are nonlocal means over the k-th nearest neighbors of ϵ_i ($\mathcal{N}_B$ is the neighborhood containing the k-th nearest neighbor (5.35) and is divided into two neighborhoods $\mathcal{N}_B = \mathcal{N}_B^+ \cup \mathcal{N}_B^-$).

$\mu_B({\mathrm{v}_i}^{l,l+1}) = \frac{1}{k}(\sum_{\mathrm{v}_j \in \mathcal{N}_B^-(\mathrm{v}_i)} {\mathrm{v}_j}^{l+1} + \sum_{\mathrm{v}_j \in \mathcal{N}_B^+(\mathrm{v}_i)} {\mathrm{v}_j}^{l})$ are nonlocal means over the k-th nearest neighbors of v_i ($\mathcal{N}_B$ is the neighborhood containing the k-th nearest neighbor (5.35) and is divided into two neighborhoods $\mathcal{N}_B = \mathcal{N}_B^+ \cup \mathcal{N}_B^-$).

The nonlinearities of these equations are isolated in the neighborhoods $\mathcal{N}_B$ (composed of the k-th nearest neighbors in the 6-dimensional space for residual, 4-dimensional space for regularization) used in the nonlocal means and in $\mathcal{D}_i$,$\mathcal{R}_i$. Indeed the distances $d_k(\epsilon)$ and $d_k(\mathrm{v})$ to the k-th nearest neighbors in $\mathcal{D}_i$ and $\mathcal{R}_i$ along with the two neighborhoods $\mathcal{N}_B$ computed for the data and regularization terms are recomputed as fixed point iterations at convergence of the SOR scheme. The numeric stability of the SOR scheme being controlled by the previously assumed diagonally dominance of the mean-shift.

7.4.2 Algorithm and equations interpretation

First, to connect with classic regularization terms, one can note that the approximation of the Laplacian in the diffusion process of [HS81] is expressed as a mean-shift, where the mean is computed in a local neighborhood. Indeed, in a more general view, heat equation results in a Laplacian-based diffusion where a standard discretization of the Laplacian can be written as a mean-shift. Isotropic regularization is then a mean-shift in the space of spatial coordinates m. Anisotropic regularizations can also be expressed as a mean-shift in the space of spatial coordinates but with weightings based on the gradient features. Indeed, the regularization part of our diffusion equations (7.19) and (7.20) are coherent with the ones in the literature

[BBPW04]: (7.21) and (7.22)

$$\mathrm{v}_{1i}^{l+1} =(1-w)\mathrm{v}_{1i}^{l} + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)_{i\sim j}^{l}\mathrm{v}_{1j}^{l+1} + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)_{i\sim j}^{l}\mathrm{v}_{1j}^{l} - \frac{(\Psi'_D)_i^l}{\alpha}(u_x u_y(i)\mathrm{v}_{2i}^{l} + u_x u_t(i))}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+} (\Psi'_S)_{i\sim j}^{l} + \frac{(\Psi'_D)_i^l}{\alpha}u_x^2(i)} \tag{7.21}$$

$$\mathrm{v}_{2i}^{l+1} =(1-w)\mathrm{v}_{2i}^{l} + w\frac{\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi'_S)_{i\sim j}^{l}\mathrm{v}_{2j}^{l+1} \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi'_S)_{i\sim j}^{l}\mathrm{v}_{2j}^{l} - \frac{(\Psi'_D)_i^l}{\alpha}(u_x u_y(i)\mathrm{v}_{2i}^{l} + u_y u_t(i))}{\sum\limits_{j\in\mathcal{N}_i^-\cup\mathcal{N}_i^+} (\Psi'_S)_{i\sim j}^{l} + \frac{(\Psi'_D)_i^l}{\alpha}u_y^2(i)} \tag{7.22}$$

where $\mathcal{N}_i^-$ denotes the neighbors j of i with $j < i$ and $\mathcal{N}_i^+$ the neighbors j of i with $j > i$, ϵ is ϵ_{LK}.

In the regularization term, local means $\sum\limits_{j\in\mathcal{N}^-(i)} (\Psi')_{i\sim j}^{l}\mathrm{v}_{1j}^{l+1} + \sum\limits_{j\in\mathcal{N}^+(i)} (\Psi')_{i\sim j}^{l}\mathrm{v}_{1j}^{l}$, (weighted by some gradient-based term $(\Psi')_{i\sim j}^{l}$) in (7.21) and (7.22) are replaced in our framework by nonlocal means $\mu_B(\mathrm{v})$. Indeed, the local neighborhoods $\mathcal{N}(i)$ are replaced by nonlocal neighborhoods $\mathcal{N}_B$. In addition, we have a mean-shift with (different) nonlocal neighborhoods $\mathcal{N}_B$ for the data term. Let us compare these different neighborhoods on a regularization example: no data term is attached in the diffusion process, (7.19) and (7.20). A simple way to visualize what are the remaining terms is to set $\mathcal{D}_i$ to zero.

We made two toy examples on Fig. 7.2 and Fig. 7.3. We plot a noisy synthetic flow with a square moving on a static background. The second flow has a grid in the foreground with a different motion. The color flow code is extracted from [BRS+07]. Noise is chosen of enough standard deviation (0.7) to overlap the distribution of the square motion $[-2, 0.5]$. We compared four different diffusions process with different neighborhoods. One in the spatial domain m (the diffusion is computed in a local neighborhood: classical isotropic diffusion), one in a weighted spatial domain (classical anisotropic diffusion), one in the feature domain $\mathrm{v}(\mathrm{m})$ (the diffusion is computed in a local neighborhood of the feature distribution) and one in the spatial and feature domain $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$ (the diffusion is computed in a local neighborhood in a joint distribution of features and spatial locations). This is a statistical and nonlocal alternative to anisotropic diffusion [CSV03] where the diffusion process is computed in a small neighborhood, weighted by the feature gradients. Results on anisotropic diffusion are, as expected, performing a lot better than isotropic diffusion on this example, however it is viewed as a weighted, but still local process. The diffusions are called local when their expression only implies neighboring pixels, even if, the effects propagate over large distances after several iterations.

Finally, the balancedness of the equations appears as it is a sum between two different mean-shifts, one for the data term, the other for the regularization term. As the mean-shifts are expressed in their kNN forms, each term involves a neighborhood made of k terms. In this statistical framework, we are adding values of similar physical quantities in a coupled diffusion process, and the tuning between the data and regularization part, if needed, would have more physical sense. Each resulting diffusion process have also the same behavior than nonlocal methods [Yar85, AW06] which have gained much interest in the recent image restoration literature as they can adapt to the inner image statistics. Finally one can still use fixed-size kernel neighborhoods instead of knn neighborhoods, the terms $\mathcal{D}$ and $\mathcal{R}$ would be

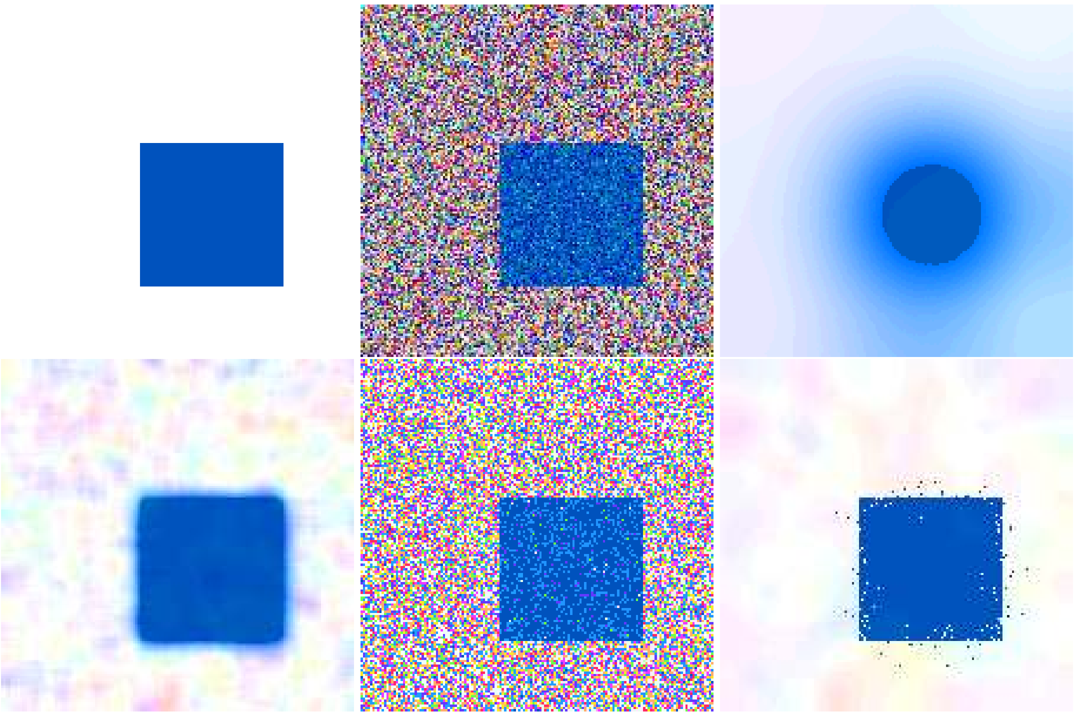


Figure 7.2: *From left to right, top to bottom: synthetic flow, noisy flow, spatial diffusion* m*, weigthed spatial diffusion (anisotropic diffusion), feature diffusion* $\mathrm{v}(\mathrm{m})$*, joint spatial and feature diffusion* $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$*. The diffusion in the spatial domain blur the contours, the diffusion in the feature domain is sensible to noise as there is no spatial diffusion, a good compromise is the spatial and feature diffusion in the 4-D space* $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$.

constant, replacing d_k by a fixed bandwidth h. The two mean-shifts would have a different behavior, one would preserve one region if its features are far enough (larger than h) from the other features in the images. The other would preserve a region if it is big enough in this image (larger than k). However, using the kNN framework is important here as we deal with high dimensional feature spaces (6-dimensional for data, 4-dimensional for regularization).

7.4.3 Multi resolution

The classical optical flow formulation contains a first order Taylor expansion to linearize ϵ. The flow must be small relatively to the image derivatives. A multi resolution scheme is then chosen: the flow is computed on down-sampled versions of the image, respecting the Shannon theorem. Instead of choosing the optical flow at lower resolution to initialize the next resolution $k+1$, we warp the image with the flow at resolution k and we initialize the flow at 0: $\mathrm{v}^{k+1} = \mathrm{v}^k + d\mathrm{v}^k$, where v^k is the optical flow obtained form the lower resolution and used to warp the image, then we are looking for a new flow $d\mathrm{v}^k$. A theoretical justification can be found in [BBPW04]. Let us now show some experiments to see the applicability of the method.

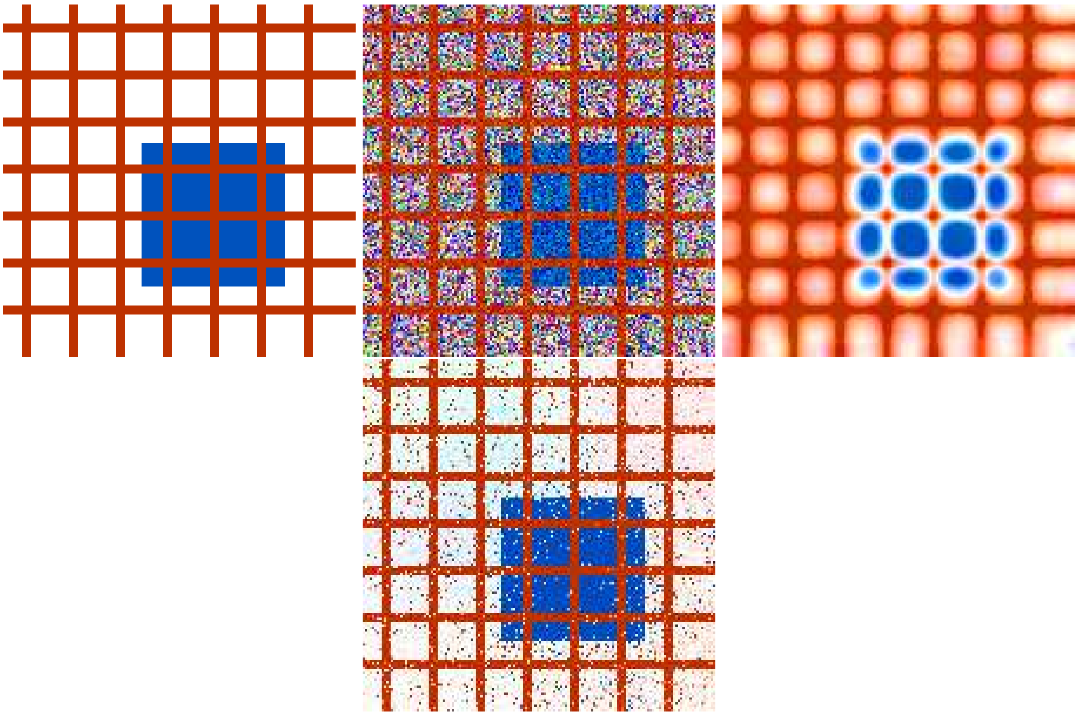


Figure 7.3: *From left to right, top to bottom: synthetic flow, noisy flow, anisotropic diffusion, joint spatial and feature diffusion* $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$*. The diffusion in the spatial domain blur the contours, the diffusion in the feature domain is sensible to noise as there is no spatial diffusion, a good compromise is the spatial and feature diffusion in the 4-D space* $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$.

7.5 Experimental results

The feature space considered for the data term in the experiments is u, 6-dimensional, composed of the three color channels of the image in the RGB color space and J_σ being the color spatial structure tensor with $\sigma = 1$. The feature space considered for the regularization term is $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$, m being the spatial coordinates of the image defined in the interval $[-1,1] \times [-1,1]$, $[0,0]$ being the center of the image. v is the optical flow field. The regularization term is then 4-dimensional. The SOR is applied in a multi resolution scheme: at the first resolution level, we have 10 fixed point iterations to get a first, rough estimation, of the PDF and 10 SOR iterations inside each fixed point iteration, at all the higher resolution levels, only 3 fixed point iterations are computed to refresh the PDF as well as 10 SOR iterations inside each fixed point iteration. The relaxation parameter w for the SOR resolution scheme is chosen to be 1.8. The k for k-th nearest neighbor which decides the size of the neighborhood to apply the mean-shift is chosen to be $\sqrt{n}$ where n is the size of the image. We validated the results on the optical flow benchmark [BRS$^+$07]. We show on Tab. 7.1 the Average Angular Error (AAE) results of our method on the publicly available data and groundtruth (sequence: ”RubberWhale”, ”Hydrangea”, ”Dimetrodon”). We also applied on the benchmark with hidden groundtruth (sequence: ”Army”, ”Mequon”, ”Schefflera”, ”Wooden”). We also show on Fig. 7.5 some visual results of the flow with the color code provided by the benchmark Fig 7.4.

Our method compares favorably to standard and more recent optical flow methods presented on the benchmark *http://vision.middlebury.edu/flow/eval/* and shows the applicability of this new framework.

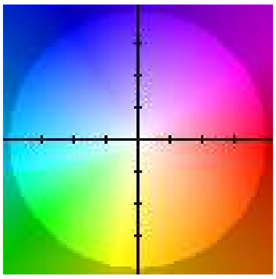
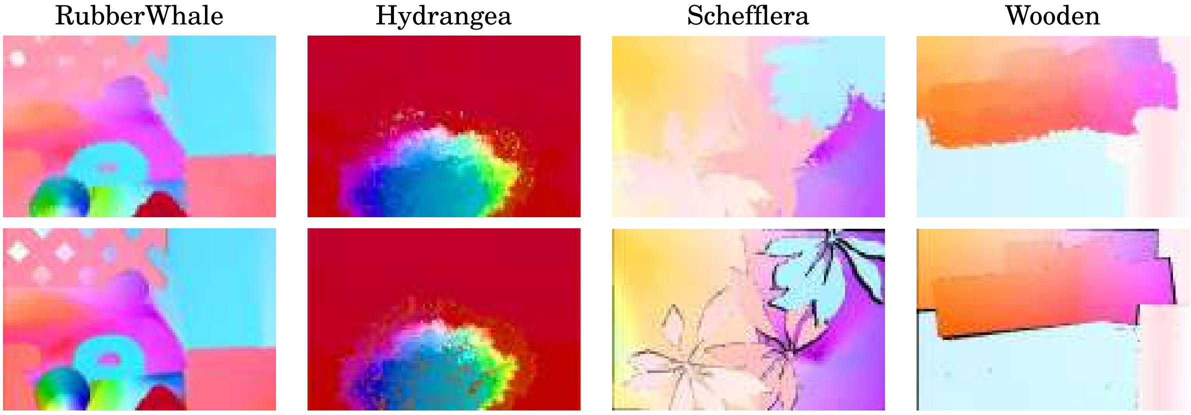


Figure 7.4: *Optical flow color code: each vector direction corresponds to a color*

Sequence	RubberWhale	Hydrangea	Dimetrodon	Army	Mequon	Schefflera	Wooden
AAE	7.12	9.82	10.83	7.40	9.04	11.40	7.62

Table 7.1: *Average angular error for our nonlocal optical flow method on several sequences*



RubberWhale Hydrangea Schefflera Wooden

Figure 7.5: *Visual results: first row, our method; second row, groundtruth*

7.6 Conclusion

In this chapter, we have proposed a general nonparametric statistical framework to compute optical flow. This framework is expressed as a sum of two entropies, one for the data term, the other for the regularization term in a balanced and meaningful functional. It naturally extends to some recent improvements of optical flow (integration of other features than image brightness) and follows the same philosophy as recent improvements in regularization (nonlocal filters). In this framework we can write some of the classical parametric optical flow formulations, as well as our new nonparametric formulation. Minimizing these energies, we end up with local diffusion for the parametric formulation, nonlocal diffusion for the nonparametric formulation, taking full advantage of estimation of statistics over the whole image. Finally, the numerical difficulties are avoided using kNN mean-shift for high dimensional features which isolates the nonlinearities in the neighborhood computations, the remaining terms of the equations being linear.

CHAPTER **8**

SEGMENTATION

This chapter revisits segmentation using region-based active contours. The main contribution of this chapter is a general high dimensional framework for active contours. Energy is expressed through information-theory inspired measures: entropy, Kullback-Leibler divergence. In particular, we derived a general framework for multivariate segmentation based on joint entropy estimated with kNN. Multivariate as any high dimensional space can be used including, RGB color spaces, nonlocal radiometric/geometric color spaces and shape priors, motion cues. On several examples, kNN estimation is performing better than classical methods even on low dimensional spaces (RGB colors). This chapter follows a general introduction written in Section 2.3.

8.1 Introduction

Image segmentation aims at partitioning an image into semantic objects. First, we consider the general case where there is poor knowledge about these semantic objects. One solution to partition the image is to follow some basic assumptions such as colors uniformity or coherence of objects. This uniformity criterion can be for instance the partition such as the variance in each region is minimum.

However color coherence or uniformity assumptions are often insufficient to characterize real-world objects, other cues must be integrated making the data vector valued. Vector values statistics in image processing applications are usually treated in two ways: either the features are assumed Gaussian distributed and a mean vector and covariance matrix are estimated to fit the data, or the features are estimated non parametrically and as Parzen techniques are impractical in high dimensions, independence between features are assumed. Both solutions are non satisfactory as various features extracted from images (RGB channels, tensors [RBD03], wavelets sub bands) are in general non independent and non-Gaussian distributed.

Information theory helps to define measures of uniformity of the (multivariate) random variables hidden in each region. The chosen measure of uniformity is the joint entropy between the features. In most cases, the features are correlated (*e.g.* the three RGB channels), in some other cases (*e.g.* color and motion), it is reasonable to consider the features as independent in order to limit the number of dimensions.

Let us see why this choice of energy is interesting. First, entropy is a measure of dispersion. If the object is assumed to be piecewise homogeneous, the color distribution has a small dispersion. Moreover, entropy coincides locally asymptotically with likelihood at the optimum [1]. Thus, a minimum entropy criterion should have near optimal performances in case of a parametric distribution while being able to adapt to nonparametric cases. In particular, entropy appears to be less sensitive to outliers in practice.

We propose a general framework for multivariate segmentation: the joint entropy allows to combine an arbitrary number of features/modalities and is estimated with kNN. Using classical entropy estimation, the number of modalities that can be combined together is limited by the number of samples available, *i.e.*, the number of pixels of the image or sequence frame. Indeed, if the samples fill the distribution space too sparsely, then classical entropy estimation (or any other statistical measure) cannot be approximated accurately. To a certain extent, the high dimensional kNN framework, defined in this manuscript, circumvents this problem.

Information theory also provides divergences connected with Shannon entropy. Kullback-Leibler divergence is used integrate priors in our energies. As automatic segmentation remains a challenge, a more realistic solution is to introduce priors. Priors can be defined for instance on the appearance, shape or motion of the object. An example is a video tracking application, if we assume that a segmentation of the object is known on a previous frame, one can use this segmentation to define a prior on the appearance or shape of the object. Again the multivariate kNN segmentation framework allows to define these priors jointly. Combination of these priors with joint entropy segmentation is coherent as Kullback-Leibler divergence is linked with Shannon entropy, however this link is not presented in this manuscript and is a perspective.

Another cue information is motion. Searching for a moving object, one can add motion as a cue information in the segmentation algorithm. Motion information can be integrated with color in a joint entropy. Joint entropy was estimated with Parzen as it is reasonable, to consider these features independent. However, kNN extension of this work is a perspective.

Finally, although some equations below have some similarities with existing, likelihood-based or Bayesian methods, the philosophy here is different and somewhat more general. Bayesian methods are directly tied to the definition of the probability of the (observed) image or sequence given a segmentation. Assuming independence between the pixels, an energy is derived, which usually writes as a sum or integral of log probabilities. In the proposed approach, each region of the segmentation is regarded as a set of samples or realizations. The energy is defined as a function of a multivariate distribution in order to best fit the needs of the specific application. The link between the energy and the samples is then made through a nonparametric, Parzen-like or kNN estimation. This allows for example to keep the same energy definition while using different object features or different assumptions on the features. In particular, one could think of discarding the assumption of independence between the pixels and use a patch-based (or neighborhood-based) approach [AW06] to change the spatial information from color to texture.

This chapter is organized as follows. Section 8.2 presents a multivariate image segmentation with the kNN framework, Section 8.3 introduces how to define priors in this framework and gives a example in video tracking using a shape and appearance prior from a previous

[1] This is interesting since the maximum likelihood estimator is optimal when the distribution of data is parametric.

frame. Section 8.4 shows how motion segmentation can be used as a cue integration in a joint entropy segmentation. Finally, Section G presents a simplification of active contours algorithms implemented for an object-driven video coding application.

8.2 Image segmentation

8.2.1 Multivariate segmentation

Parametric and nonparametric segmentation

The goal is to define a general functional for image segmentation. Let us remind the Bayesian model for image segmentation (2.35):

$$E_B(\Gamma) \quad = \quad -\int_{\Omega} \log f_{\Omega}(u(\mathrm{m}))\ \mathrm{dm} \tag{8.1}$$

where $u(\mathrm{m})$ is the pixel feature value at position m, Ω is the inner region of Γ, f_Ω is the PDF estimated over Ω. Estimation of f_Ω can be parametric or nonparametric: using Gaussian assumption of f_Ω and region competition this model is equivalent to [CV01], using nonparametric estimation of f_Ω and region competition this model is [ZY96].

A new multivariate framework based on information theory

As mentioned in the introduction, a more general nonparametric model for image segmentation is based on information theory, for instance [KFY$^+$05]. We propose to extend this model to vector-valued data. Let us consider a contour Γ defining region Ω and features inside this region $U = \{u(\mathrm{m}) \mid \mathrm{m} \in \Omega\}$. Features U are vector-valued random observations of a multivariate random variable. This variable has the following joint differential entropy:

$$H(\Gamma) = -\int_{\mathbb{R}^d} f_\Omega(\alpha)\ \log f_\Omega(\alpha)\ \mathrm{d}\alpha \tag{8.2}$$

where f_Ω is the PDF over the region Ω, d is the dimension of the features u. Since the actual PDF f_Ω is unknown, it must be estimated. As u is vector-valued, multivariate Parzen windowing method estimates the PDF:

$$\hat{f}_\Omega(\alpha) = \frac{1}{|\Omega|}\int_\Omega K_h(\alpha - u(\mathrm{m}))\ \mathrm{dm} \tag{8.3}$$

where $|\Omega|$ is the area of Ω, where K_h is a multivariate, Gaussian kernel with standard variation, or bandwidth, h.

As segmentation energies are usually defined by an integral on the region, see for examples [CS05, BRDW03, KFY$^+$05, RP02, ABFJB03], the Ahmad-Lin estimate (8.4) is preferred to the classical integral estimate (8.2)

$$\hat{H}_{\mathrm{AL}}(\Gamma) = -\frac{1}{|\Omega|}\int_\Omega \log f_\Omega(u(\mathrm{m}))\ \mathrm{dm}. \tag{8.4}$$

where again f_Ω is estimated with (8.3).

Summarizing, we defined an information-theoretic energy for multivariate features u and try to find a shape Γ, whose inner features have the minimum dispersion:

$$E_{\mathrm{IT}}(\Gamma) \quad = \quad -\frac{1}{|\Omega|}\int_\Omega \log f_\Omega(u(\mathrm{m}))\ \mathrm{dm} \tag{8.5}$$

where f_Ω is the PDF of the features u estimated inside region Ω.

$$f_\Omega(u(\mathrm{m})) \quad = \quad \frac{1}{|\Omega|}\int_\Omega K_h(u(\mathrm{m}) - u(\mathrm{n}))\ \mathrm{dn} \tag{8.6}$$

where K_h is a multivariate kernel of bandwidth h.

Region competition

In practice, due to approximations and roundoff errors, energy (8.5) might have the empty set as a unique global minimizer. A common solution is known as region competition: the energy of the background is added to the energy (8.5) of the object. It is not mandatory to use the same energy for the object and the background. However, it can be appropriate to do so. As a result, the segmentation will represent a trade off between the minimization of the object energy and the minimization of the background energy. It can also be interpreted as the maximal separation between object and background descriptors [YTW02], here, the respective joint distributions.

To account for the relative areas of the object and the background, or, in other words, to account for the probability of a pixel to belong to either of them, the following weighted sum will be used

$$E_{\mathrm{rc}}(\Gamma) = \frac{|\Omega|}{|D|}\ E_{\mathrm{IT}}(\Gamma) + \frac{|\overline{\Omega}|}{|D|}\ E_{\mathrm{IT}}(\Gamma^c) \tag{8.7}$$

where $\overline{\Omega}$ is the complement of Ω in D, the image domain, and Γ^c is its boundary $\partial\overline{\Omega}$.

Energy (8.7) can be rewritten as

$$E_{\mathrm{rc}}(\Gamma) = p(C=1)\ E_{\mathrm{IT}}(\Gamma) + p(C=0)\ E_{\mathrm{IT}}(\Gamma^c) \tag{8.8}$$

where C is the characteristic function of the object and $p(C = i)$ denotes the probability of the event $C = i$. Finally, as the division by $|D|$ can be omitted since it has no influence on the minimization, energy writes:

$$E(\Gamma) = |D|.E_{\mathrm{rc}}(\Gamma) = \int_\Omega \log f_\Omega(u(\mathrm{m}))\ \mathrm{dm} + \int_{\overline{\Omega}} \log f_{\overline{\Omega}}(u(\mathrm{m}))\ \mathrm{dm} \tag{8.9}$$

This energy ends up with the same equations as maximum likelihood in a Bayesian segmentation framework (2.35), where PDFs are estimated nonparametrically.

The prior term on the smoothness of the curve is removed since we will use a spline parametrization of the active contour, the smoothing parameter is hidden in the spline construction.

Shape derivative and evolution equation

Detailed in Appendix E, the shape derivative of (8.9) is equal to

$$\begin{aligned}
\mathrm{d}E(\Gamma, F) \quad = \quad & \int_\Gamma \left(-1 + \log f_\Omega(u(s)) \quad + \frac{1}{|\Omega|}\int_\Omega \frac{K_h(u(s) - u(\mathrm{m}))}{f_\Omega(u(\mathrm{m}))}\ \mathrm{dm}\right)\ N(s)\cdot F(s)\ \mathrm{d}s \\
& - \int_\Gamma \left(-1 + \log f_{\overline{\Omega}}(u(s)) \quad + \frac{1}{|\overline{\Omega}|}\int_{\overline{\Omega}} \frac{K_h(u(s) - u(\mathrm{m}))}{f_{\overline{\Omega}}(u(\mathrm{m}))}\ \mathrm{dm}\right)\ N(s)\cdot F(s)\ \mathrm{d}s \\
= \quad & \int_\Gamma \Big(\log f_\Omega(u(s)) - \log f_{\overline{\Omega}}(u(s)) \\
& + \frac{1}{|\Omega|}\int_\Omega \frac{K_h(u(s) - u(\mathrm{m}))}{f_\Omega(u(\mathrm{m}))}\ \mathrm{dm} - \frac{1}{|\overline{\Omega}|}\int_\Omega \frac{K_h(u(s) - u(\mathrm{m}))}{f_{\overline{\Omega}}(u(\mathrm{m}))}\ \mathrm{dm}\Big)\ N(s)\cdot F(s)\,\mathrm{d}s
\end{aligned} \tag{8.10}$$

The shape derivative (8.10) has the following form

$$\mathrm{d}E(\Gamma, F) = \int_{\Gamma} ((\alpha(s) - \alpha^c(s))\, N(s)) \cdot F(s)\, \mathrm{d}s = \langle (\alpha - \alpha^c)\, N, F \rangle \tag{8.11}$$

where $\langle , \rangle$ is the L^2-inner product on Γ. Therefore, $(\alpha - \alpha^c)\, N$ is, by definition, the gradient of (8.8) at Γ associated with this inner product.

Based on the notion of gradient defined in (2.42), energy (8.9) can be minimized using a steepest descent procedure in the space of contours. The following contour evolution process is known as the active contour technique [CKS97, HR04]: an initial contour[2] is iteratively deformed in the opposite direction of the gradient until a convergence condition is met. The evolution equation of the active contour is written as follows

$$\begin{cases} \Gamma(\tau = 0) = \Gamma_0 \\ \dfrac{\partial \Gamma}{\partial \tau} = (\alpha^c - \alpha)\, N \end{cases} \tag{8.12}$$

where τ is the evolution parameter and α^c has the same expression as α but is evaluated on $\overline{\Omega}$. The convergence condition is $\alpha^c - \alpha = 0$. This evolution equation is implemented using explicit parametrization of active contours, *i.e.* a smoothing spline. This parametrization will be preferred to the implicit representation along this thesis for its computational speed. The active contour toolbox used for implementation is publicly available at `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=11643&obje` and can manage topological changes.

8.2.2 kNN multivariate segmentation

Energy

In order to better evaluate statistics on vector-valued data, we propose a kNN estimate of entropy in a region competition framework:

$$E(\Gamma) = |U|(\log(v_d(|U|{-}1)){-}\log(k){+}d\,\mu_U(\log \rho_k(U))){+}|\overline{U}|(\log(v_d(|\overline{U}|{-}1)){-}\log(k){+}d\,\mu_{\overline{U}}(\log \rho_k(\overline{U}))) \tag{8.13}$$

where $U = \{u(\mathrm{m}) \mid \mathrm{m} \in \Omega\}$ and $\overline{U} = \{u(\mathrm{m}) \mid \mathrm{m} \in \overline{\Omega}\}$. In order to compute the shape derivative, let us remind that this energy has a PDF interpretation and is equal to:

$$E(\Gamma) = \int_{\Omega} \log f_{\Omega}(u(\mathrm{m}))\, \mathrm{dm} + \int_{\overline{\Omega}} \log f_{\overline{\Omega}}(u(\mathrm{m}))\, \mathrm{dm} \tag{8.14}$$

where f is estimated through the balloon estimate

$$\hat{f}_U(u(\mathrm{m})) = \frac{1}{|U|} \int_{\Omega} K_{h(u(\mathrm{m}))}(u(\mathrm{m}) - u(\mathrm{n}))\, \mathrm{dn} \tag{8.15}$$

where K is a uniform kernel of balloon variable bandwidth $h(u(\mathrm{m}))$. The same shape derivative scheme as detailed in Appendix E can thus be applied.

[2] For example, a user-defined contour.

Shape derivative

Shape derivative (8.10) becomes in the kNN framework:

$$\mathrm{d}E(\Gamma,F) = \int_\Gamma \Bigg(d\log\rho_k(\overline{U},u(s)) - d\log\rho_k(U,u(s)) \\ +\frac{1}{k}\left[\sum_{u\in\mathcal{N}_B(U,u(s))} \frac{\rho_k(U,u)}{\rho_k(U,u(s))} - \sum_{u\in\mathcal{N}_B(\overline{U},u(s))} \frac{\rho_k(\overline{U},u)}{\rho_k(\overline{U},u(s))}\right] \Bigg) N(s)\cdot F(s)\ \mathrm{d}s \tag{8.16}$$

where $\mathcal{N}_B(U,u(s))$ is the balloon neighborhood of $u(s)$ in U (5.35)[3].

This shape derivative splits in two parts, divided in two rows. Indeed energy E has a double dependency in Ω. The first row comes from the first dependence of Ω in the entropy construction (8.5). The second row is for the second dependence of Ω in the PDF construction (8.6). The same remarks holds for $\overline{\Omega}$.

The first term on the equation $\log\rho_k(\overline{U},u(s)) - \log\rho_k(U,u(s))$ is intuitive and connects with knn clustering. The contour will inflate toward the region containing the nearest kNN. Using clustering terminology, a point on the contour will be labeled with the class containing the nearest kNN to the point. kNN clustering [FH51] is similar, each point has the class label of the most common class among its k-nearest neighbors.

The second term of the equation is the second influence of a sample in the PDF estimate. In practice, this term vanishes for large enough regions as the influence of one sample point in the PDF of the region is minimum.

Finally, the shape derivative can be written as (8.11) and is minimized using a evolution equation (8.12) in an active contour framework.

Local PDF definition

In order to define more accurate statistics inside a region, we propose to add a spatial constraint in the PDF construction.

In order to define local statistics, again, the d-dimensional color features U are enriched with geometry $U_g = \{[\mathrm{m}, u(\mathrm{m})] \mid \mathrm{m}\in\Omega\}$. $(d+2)$-dimensional sample points of these features enriched with geometry are noted $u_g(\mathrm{m}) = [\mathrm{m}, u(\mathrm{m})]$.

This choice corresponds to the illustration Fig. 5.2(d) to model natural image statistics in a nonlocal neighborhood. The kNN energy inspired of classical energies (8.13) corresponds to illustration Fig. 5.2(c). Statistics are computed on all the image without spatial constraints.

$$E(\Gamma) = |U_g|(\log(v_d(|U_g|-1)) - \log(k) + d\,\mu_{U_g}(\log\rho_k(U_g))) + |\overline{U_g}|(\log(v_d(|\overline{U_g}|-1)) - \log(k) + d\,\mu_{\overline{U_g}}(\log\rho_k(\overline{U_g}))) \tag{8.17}$$

Again the shape derivative writes

$$\mathrm{d}E(\Gamma,F) = \int_\Gamma \Bigg(d\log\rho_k(\overline{U_g},u_g(s)) - d\log\rho_k(U_g,u_g(s)) \\ +\frac{1}{k}\left[\sum_{u_g\in\mathcal{N}_B(U_g,u_g(s))} \frac{\rho_k(U_g,u_g)}{\rho_k(U_g,u_g(s))} - \sum_{u_g\in\mathcal{N}_B(\overline{U_g},u_g(s))} \frac{\rho_k(\overline{U_g},u_g)}{\rho_k(\overline{U_g},u_g(s))}\right] \Bigg) N(s)\cdot F(s)\ \mathrm{d}s \tag{8.18}$$

[3] The k-th nearest neighbors of u(s) in U

where $\mathcal{N}_B(U_g, u_g(s))$ is the balloon neighborhood of $u_g(s)$ in U_g (5.35)[4],

Again, the shape derivative can be written as (8.11) and is minimized using a evolution equation (8.12) in an active contour framework. Geometry is present explicitly as a new feature. It enforces statistics to be computed in a spatial local window around point s. As it is known that defining local statistics increases the number of local minima, this method will be used after the minimization of a kNN energy with global statistics (8.13).

This method connects with recent methods which performed local segmentation by locally weighting the PDF construction [MAT07].

8.2.3 Experimental results

Setup

The images are extracted from a segmentation benchmark [MFTM01], Fig. 8.1. This benchmark also provides ground truth. Unfortunately, this ground truth provides only contours and not regions. As the above algorithms are region-based, we filled these contours to form regions of interest (Fig. 8.2). In this section, the performance of seven algorithms are compared on this image basis.

The segmentation algorithms are all region-based active contours, with an joint entropy energy (8.9) (and to a certain extent, Bayesian energies) on RGB colors : u is 3-d dimensional, u_g is 5-d dimensional, only the methods used to estimate PDFs differ.

The seven different PDF estimations are:

- SSD: Gaussian distribution on independent features, equivalent to [CV01] using the square function, each color channels are summed independently,
- SAD: Laplacian distribution on independent features, equivalent to SSD but using absolute value,
- Pz-Hind: Nonparametric distribution with Parzen estimates (8.9) and where the dimensions are treated independently (product of univariate distributions), bandwidth is tuned by rule-of-thumb,
- Pz-H: Joint Parzen estimation (3-dimensional multivariate distributions), bandwidth is tuned by rule-of-thumb,
- kNN-H: kNN estimate of PDF and entropy (3-dimensional multivariate distributions) (8.13),
- kNN-H-G: kNN estimate of entropy and PDF with spatial constraints, *i.e.* local statistics (5-dimensional multivariate distributions) (8.17),
- kNN-H-G-I: kNN estimate of entropy and PDF with spatial constraints (5-dimensional multivariate distributions) *i.e.* local statistics and initialized with kNN-H *i.e.* global statistics .

As groundtruth is avaliable, the error criterion is a percentage of miss classified pixels normalized by the size of the region.

$$d = 100 \frac{\sum\limits_{i,j} |M_g(i,j) - M_c(i,j)|}{\sum\limits_{i,j} M_g(i,j)} \tag{8.19}$$

[4] The k-th nearest neighbors of $u_g(s)$ in U_g

where M_g is the mask of the ground truth segmentation and M_c is the mask obtained with the segmentation algorithm. Initialization by a user is simulated by defining a rectangular region of interest, $2/3$ smaller than the real bounding box Fig. 8.3.

The results are presented in the table Tab. 8.1. For each image, if the result is worse than initialization in terms of error (8.19), it is noted X. Green color means it is in the best performing methods $(+/-5)$ in terms of error, and orange color means it is in the good performing methods $(+/-10)$ in terms of error. Last row is the count of how many times the method was labeled green (first number) and orange (second number). The last row maximum possible score is 27, as on 5 images, all segmentation algorithms fail.

kNN-H segmentation on RGB components is giving the best results compared to other segmentation algorithms. It outperforms similar algorithms Pz-Hind and Pz-H based on entropy but with Parzen estimates of entropy. Finally, local PDF segmentation kNN-H-G and kNN-H-G-I are performing better on some images. However, they are too sensitive to local minima and are not performing better than kNN-H on RGB components on the whole image base.

Visual segmentation results of the top performing method are showed on Fig. 8.4, kNN-H. Only the 27 images where segmentation was possible with one at least of the methods are showed.

8.2.4 Conclusion

Results on the segmentation image base tend to show that even when performing on relatively low dimensional spaces (3-D spaces: RGB components), Parzen windowing, either estimated on joint feature space Pz-H or with independence assumptions Pz-Hind, reaches its limitation. Whereas H-kNN provides accurate segmentation due to the accuracy of kNN estimate of entropy and the kNN PDF estimate (in the evolution equation).

In order to improve accuracy of the method, one can introduce priors in the same framework, for instance by using a divergence linked with Shannon entropy: the Kullback-Leibler divergence, based on features spaces defining priors. In the next section, a joint appearance and shape prior will be defined as a Kullback-Leibler divergence on an high dimensional appearance-shape space, following the same presentation of the tracking algorithms detailed in Chapter 6.

8.3 Segmentation with a prior: a tracking example

In the last section, segmentation priors were limited to smooth contour assumptions. If a more accurate prior is available, for instance, a training set of segmentations of similar objects of interests. One way to integrate this prior is through computing of a similarity measure between the prior and the image being segmented.

Information theory provides a measure in connection with entropy, the Kullback-leibler divergence. The example chosen in this section is video tracking, the goal is to incorporate priors from the previous segmented frames. Two priors are then available, an appearance prior of the object of interest, as well as a shape prior of this object. While these two priors are often treated independently, we propose a joint prior definition in a high dimensional space. This high dimensional space will be efficiently handled by the kNN framework. Finally, this problem is presented as in Chapter 6, a tracking problem. The differences come from the geometric features, which must handle deformable shapes, and the active contour framework.

Energy Img	Init	SSD	SAD	Pz-Hind	Pz-H	kNN-H	kNN-H-G	kNN-H-G-I
1	55.1	13.7	13.8	13.7	13.9	14	18.6	14.7
2	69	26	28.7	38.2	13.8	12.9	11.3	11.2
3	78.2	X	X	X	X	67.3	X	74.5
4	65.4	13.6	13.5	23.6	12.1	9.7	15.6	15.4
5	45.9	X	X	39.8	32.2	35.2	X	X
6	65.7	17.7	18	25.4	5.5	5.3	20.5	5.6
7	51.6	X	X	X	X	X	X	X
8	81.4	28.3	29.2	27.9	7.3	8.2	41.4	7.9
9	62.7	60.9	61.4	39.8	X	38.9	X	X
10	100.4	X	X	X	68.5	X	X	X
11	97.9	59.8	62.8	79.1	54.5	70.8	X	X
12	49.7	X	X	X	37.5	10.6	35.4	14.3
13	87.7	69.9	70.8	72.1	60.4	19.8	X	14.3
14	67.4	27.3	27.9	26.6	29.5	18.4	21.3	17.3
15	39.7	32.1	31.9	21.8	25.6	21.3	17.3	18.2
16	35.3	17.9	X	14.0	X	8.5	8.1	7.8
17	61.1	X	X	X	X	X	X	X
18	51.5	X	X	18.1	8.9	9.1	16.9	10.3
19	80.1	X	X	73.5	77.9	X	X	X
20	43.1	X	X	X	X	14.1	X	X
21	67.3	X	X	45.9	X	34.3	36.3	36
22	51.9	X	X	X	X	X	X	X
23	54.7	X	X	X	X	X	X	X
24	45.7	X	X	X	35.2	26.8	26.8	25.5
25	107.5	103.1	X	37.0	60.6	29	X	28.3
26	42.3	X	X	X	9.5	8.3	7.8	7.6
27	55.1	X	X	X	X	X	X	X
28	48.7	X	X	X	22.9	15.1	19	20.2
29	44.5	39.7	40.4	37.8	15.7	13.4	18.9	19
30	49.6	X	X	49.1	21.3	21.6	21.2	21.1
31	43.9	X	X	22.0	27.9	9.6	14.3	14.4
32	38.7	X	X	21.6	6.2	3.8	6.1	6.6
Score(on 27)		1 +1	1 +1	3 +4	14 +3	23 +1	10 +5	17 +4

Table 8.1: *Results compares 7 segmentation energies on 32 images of the segmentation benchmark. Error (8.19) based on normalized symmetrical mask difference with ground truth for 32 images (rows) comparing 7 segmentation methods (columns). X means segmentation is worse than initialization. Green color means it is in the best performing methods* $(+/-5)$ *and orange color means it is in the good performing methods* $(+/-10)$. *Last row is the count of how many times the method was labeled green (first number) and orange (second number).*

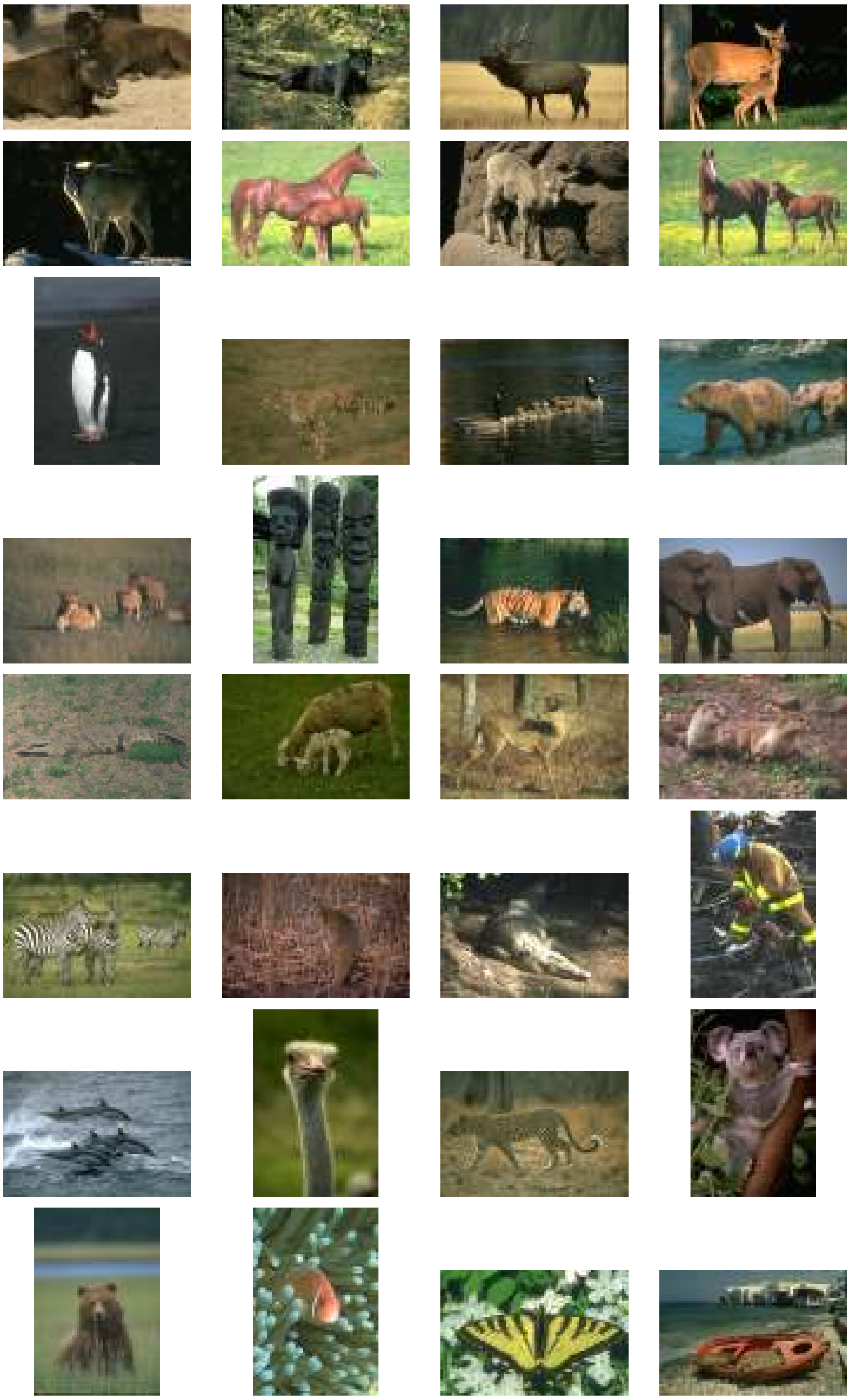


Figure 8.1: *Image database extracted from Berkeley segmentation benchmark*

8.3.1 Introduction

Two aspects of similarity measures between the reference region and a target region can be distinguished: radiometry, which indicates if the regions have similar color distributions (appearance prior), and geometry, which correlates where these colors are present in each region (shape prior). Similarity measures based solely on radiometry include distances between

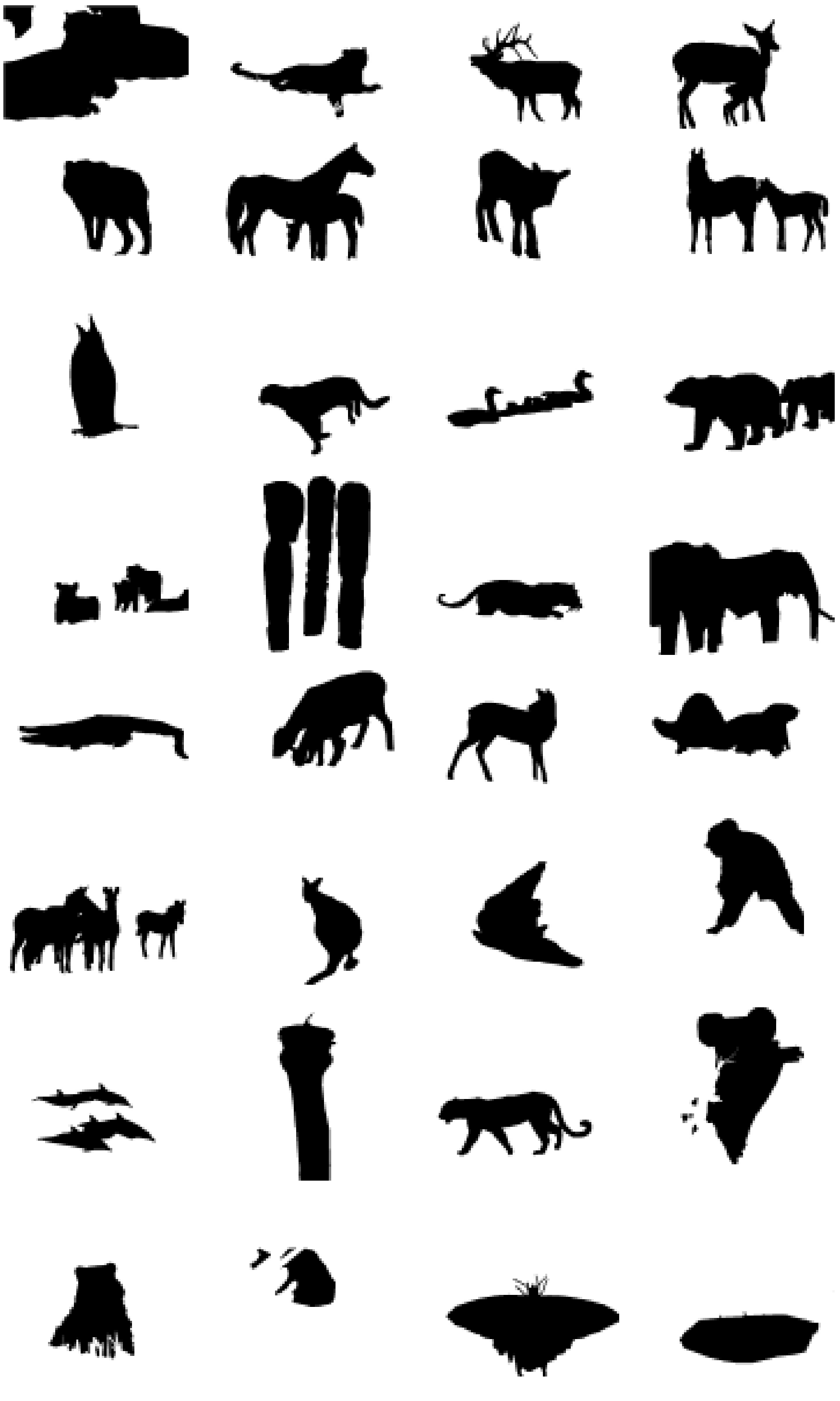


Figure 8.2: *Ground truth segmentation masks, extracted from Berkeley segmentation benchmark and filled to define objects of interest*

color histograms or probability density functions (PDF), for instance, mutual information [KFY$^+$05], Kullback distance [FZ04]. Not accounting for the information of where a given color was present in the region allows to be more flexible regarding the geometric transformation between the reference region and the target region. However, it increases the number

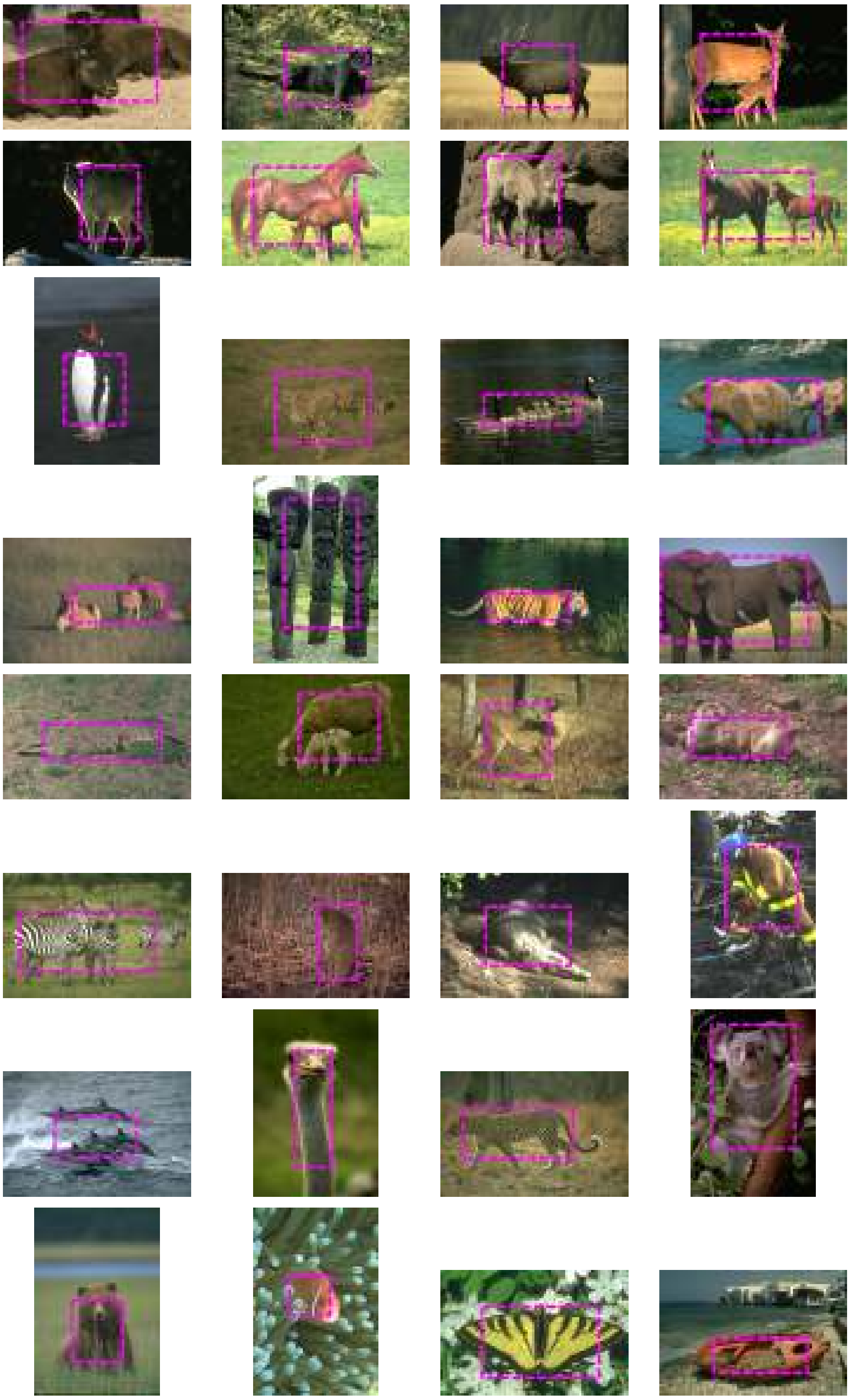


Figure 8.3: *Simulated initializations by an operator: an active contour* 2/3 *smaller than the actual bounding box of the mask is chosen.*

of potential matches and then the risk for the tracking to fail after a few frames. This can be avoided by using a geometry-aware similarity measure. The absence of geometric information implies that several candidate regions can appear as good matches.

As an alternative, geometry can be added by means of a motion model used to compute

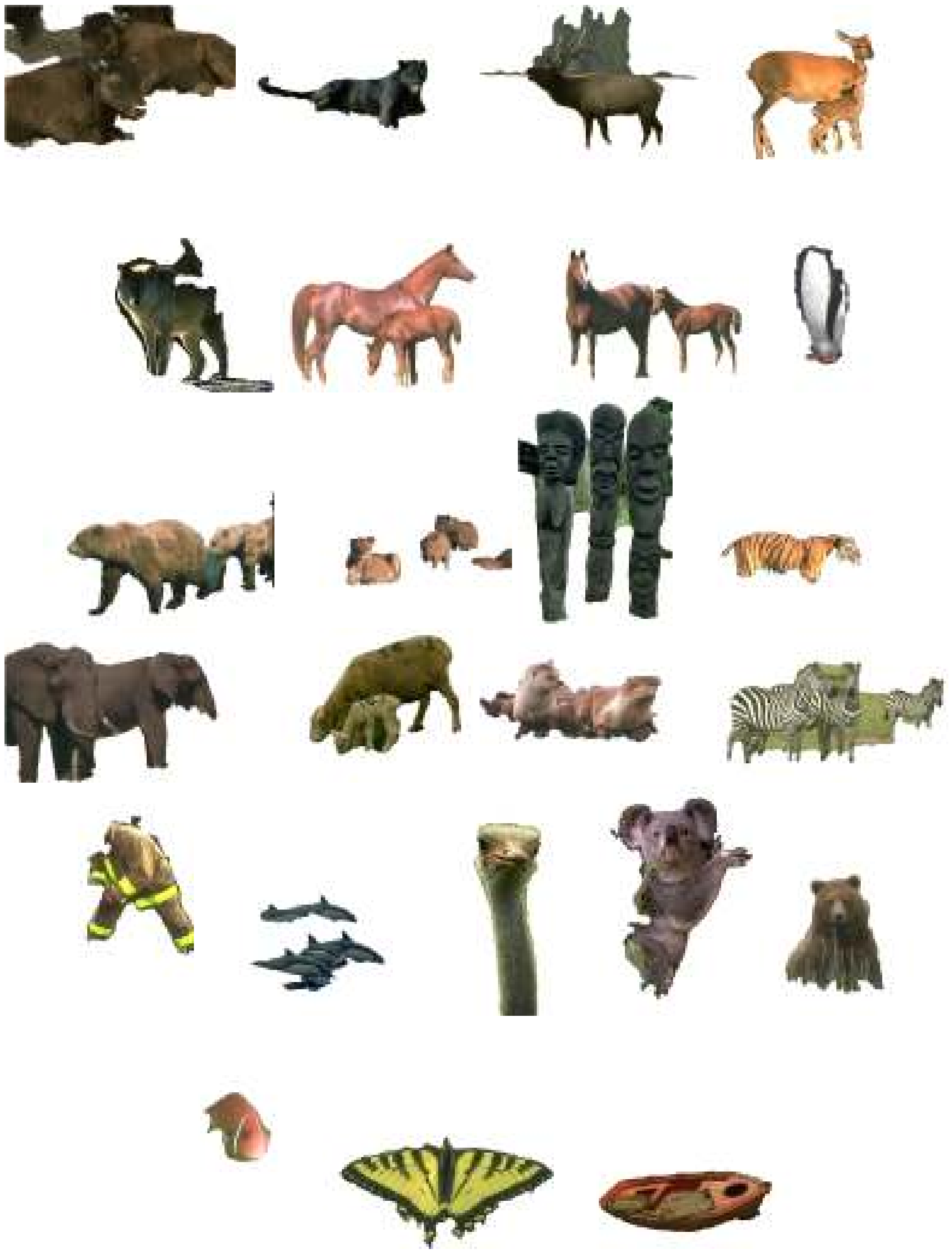


Figure 8.4: *The 24 segmentations labeled green or orange for top performing segmentation algorithm kNN-H, in table 8.1*

the point-wise residual between reference and candidate regions. A function of the residual can serve as a similarity measure, classically, the sum of squared differences (SSD), functions used in robust estimation [BA96] such as the sum of absolute differences (SAD), or statistical measures. An example where the energy is defined on a first order approximation of the point-wise residual, the optical flow constraint, in segmentation is [CS05]. However, in presence of complex motions or homogeneous zones, the residual term looses its efficiency. The geometric constraint can be softened, *e.g.*, by combining an energy based on a color distribution and an other based on optical flow [BRDW03]. An alternative is to add a shape prior to the energy [RP02, COS06].

Another approach defines a joint geometric/radiometric PDF, for example in bounding box tracking [EDD03]. Geometric data add a spatial location information to the radiometric PDF. In bounding box tracking [EDD03], the choice of good spatial coordinates is straightforward, for instance Cartesian coordinates of the box. However, defining spatial coordinates for deformable regions is an issue. One can think on defining a coordinate system embedded in the region, shape coordinates. Shape coordinates can be canonical Cartesian coordinates or polar coordinates. The latter would better handle rotations of some parts of the shape. The polar coordinates mapping of regions is a simplified version of the recent results for shape recognition in [SM06]. However, the last two shapes coordinate change when the region goes under deformations. A third shape coordinate based on the distance map, already applied in medical imaging [LFGWI00], can be considered. These shape coordinates suit very well medical imaging as the intensity is constant over the level sets of the regions. To handle non medical videos, one can extend it adding a nearest contour coordinate, the arc length of the nearest point (NAL). This approach is motivated by the fact most of the shape coordinates and shape correspondences algorithms in the literature [BMP02] are based on correspondence of the contour. By combining both distance map and contour correspondence with NAL, we extend this model to region correspondence. These region coordinates are constant under many object deformations.

Shape coordinates are added to radiometric data in a single joint high-dimensional PDFs. Although there are efficient methods [Sco92] to estimate multivariate PDFs using Parzen windowing, limitations appear as the dimension of the domain of definition of the PDFs increases.

The first contribution of this section is to apply in segmentation a joint radiometric/geometric, color-spatial criterion [EDD03]. We propose as geometric data, shape coordinates, adapted for deformable regions. Recent segmentation methods also tried to combine multiple features (spatial data, gradient, wavelets coefficients, motion) to perform accurate segmentation [BRDW03]. The PDFs are then high-dimensional and some assumptions have to be made (*e.g.*: independence between components, Gaussian assumptions). The second contribution of this section is to plug kNN methods in an active contour framework using the shape derivative tool [ABFJB03]. This high dimensional Kullback distance is not differentiable, however using the shape derivative, no direct differentiation of the Kullback distance is needed and we can bypass this difficulty.

In this section we use the high dimensional statistical measure estimation based on the kNN framework proposed in [BDB07]. This study in tracking did not require PDF estimation. Here we build the kNN framework using both kNN PDF estimation and statistical measure estimation. Indeed, in this section we compute the shape derivative of the criterion proposed in [BDB07] and this derivative requires the underlying PDF estimation of the statistical measure. Moreover the work [BDB07] was presented on rigid shapes (rectangles) with Cartesian coordinates. This section takes into account deformable shapes and a new system of coordinates had then to be defined.

This section is organized as follows. Section 8.3.2 defines Kullback distance on geometric/radiometric data. Section 8.3.3 defines geometric data for deformable shapes. In Section 8.3.4 we plug this distance in a segmentation method through active contours. Section 8.3.5 provides some results of segmentation performed on two standard sequences. Finally Section 8.3.6 concludes.

8.3.2 Similarity measures with a soft geometric constraint

Let I_{ref} and I_{target} be, respectively, the reference frame in which the ROI Ω_R is (user-) defined and the candidate, or target, frame in which the region Ω_T best matches the ROI, in terms of a given similarity measure, is to be searched for. This search amounts to finding the region Ω_T which minimizes

$$E(\Omega_T) = D(I_{\text{ref}}(\Omega_R), I_{\text{target}}(\Omega_T)) \tag{8.20}$$

where D is a similarity measure, or distance, between the two sets of data. Ω_T and Ω_R are subsets of $\mathbb{R}^2$ (or subsets of $\mathbb{N}^2$ in the discrete framework).

For clarity, the reference data set $I_{\text{ref}}(\Omega_R)$ will be denoted by R and the target data set $I_{\text{target}}(\Omega_T)$ will be denoted by T. Thus, $r(\mathrm{m})$ resp. $t(\mathrm{m})$, $\mathrm{m} \in \Omega_R$ resp. $\mathrm{m} \in \Omega_T$, represent corresponding samples from their regions R resp. T. Traditionally, $r(\mathrm{m})$ and $t(\mathrm{m})$ are a triplet of color components in a given color space, *e.g.*, RGB or YUV.

Two aspects of similarity measures can be distinguished: radiometry which indicates if the regions have similar colors and geometry which correlates where these colors are present in the regions. Measures based solely on geometry, do the point-wise difference between the reference region and the target region. An example in segmentation is based on the Taylor expansion of this point-wise difference, the optical flow constraint [CS05].

Measures based solely on radiometry include distances between the probability density functions (PDF) of the color information in the regions, for example mutual information [KFY$^+$05], Hellinger distance [ABFJB03].

A widely used distance, in segmentation in [FZ04], is the Kullback divergence[5]

$$\begin{aligned} D_{\text{KL}}(T, R) &= \int_{\mathbb{R}^d} f_T(\alpha) \, \log \frac{f_T(\alpha)}{f_R(\alpha)} \, \mathrm{d}\alpha \\ &= -H(f_T) + H_\times(f_T, f_R) \end{aligned} \tag{8.21}$$

where f_T is the PDF of data set T, f_R is the PDF of data set R, H is the Shannon entropy and $H_\times$ is the cross entropy, also called relative entropy or likelihood.

The geometric constraint can be softened by expressing it in the PDF-based approach, *i.e.*, by adding geometry to the original radiometric data [EDD03]. Formally, the PDF $f_R(\alpha)$ resp. $f_T(\alpha)$ is built on $\alpha = t(\mathrm{m}) = \{I_{\text{target}}(\mathrm{m}), \mathrm{m}\}$ for $\mathrm{m} \in \Omega_T$ resp. on $\alpha = r(\mathrm{m}) = \{I_{\text{ref}}(\mathrm{m}), \mathrm{m}\}$ for $\mathrm{m} \in \Omega_{\text{ref}}$. m are spatial features, based on a coordinate system. In the next section, we will discuss which coordinate system could be used and which best fits our method.

8.3.3 Spatial features

First spatial features choice are the canonical Cartesian coordinates of the image. It needs though to be compensated by the motion of the region. A complex motion model is hard to define (for example for articulated objects) and computationally expensive to estimate. Instead we propose to define coordinates embedded in the region, named shape coordinates. In this way, estimation motion is skipped as shape coordinates change when the region deforms.

[5] Kullback-Leibler divergence is a not a distance, as it is not symmetric. The symmetrised Kullback divergence $D_{\text{KL}}(T, R)$+ $D_{\text{KL}}(R, T)$ is a distance. For clarity, we presented all the calculus with the classic Kullback-Leibler divergence.

The segmentation and motion deformation (hidden in the shape coordinates) are then jointly solved.

Most shape coordinates for shape correspondence are based on contour [BMP02]. Here we aim at defining interior region coordinates. In practice, shape coordinates should have three properties. First they should map efficiently the region, *i.e.* each point should have a unique representation in the shape coordinates. Second, the shape coordinates should remain constant when the region goes under deformation. Third property, computational speed must remain reasonable, as the spatial features are computed at each iteration of the active contour evolution.

Region Cartesian coordinates

One can define Cartesian coordinates local in the region, for example, we choose the bounding box of the region. Inside the bounding box, we define Canonical Cartesian region coordinates $\{x_{\text{region}}, y_{\text{region}}\}$.

This method, Kullback with Cartesian geometric data (KL-CG) has 5-dimensional features: $\{Y, U, V, x_{\text{region}}, y_{\text{region}}\}$. The last two features are plotted on a example on Fig. 8.5.

This model should perform well for rigid objects, and is the one used for bounding box tracking in [EDD03].

Region polar coordinates

We now consider polar coordinates. We define as the origin of the polar coordinates the barycenter of the region. When the object is articulated, the radius coordinate should remain constant, while the angle coordinate should measure the deformation. On the opposite, with the previous region Cartesian coordinates, both coordinates change, in particular articulated members under rotation far from the barycenter (for example feet of a human body).

This approach could be extended to conformal mapping of a shape to a circle [SM06]. But for computational considerations, as the spatial features will have to be computed at each iteration of the active contour framework, we preferred basic polar coordinates.

This method, Kullback with polar geometric data (KL-PG) has 5-dimensional features: $\{Y, U, V, r_{\text{region}}, \theta_{\text{region}}\}$. The last two features are plotted on a example on Fig. 8.5. The discontinuity visible on this figure is due to the transition between the angles $-\pi$ to $+\pi$.

Distance map and contour correspondence

As mentioned in the previous section, both Cartesian and polar coordinates on the region change when the region goes under deformations. In this section, we define shape coordinates constant under shape deformations. First, we propose to use the distance map d

$$d(\mathrm{m}) = \min_{t\in[0,1]} ||\mathrm{m} - C(t)||_2 \tag{8.22}$$

where $C : [0, 1] \to \mathbb{R}^2$ is a parametric curve representation of the contour.

This model has been proposed in medical images [LFGWI00] where regions of interest have uniform intensity on the level sets of the distance map. This model is in general not true on non medical videos. We propose to complete the distance map with another spatial feature: a contour correspondence. We chose the simplest contour correspondence coordinate, arc length of the nearest point on the contour l (NAL).

$$t_{\mathrm{m}} = \arg\min_{t\in[0,1]} ||\mathrm{m} - C(t)||_2 \tag{8.23}$$

$$l(\mathrm{m}) = \int_0^{t_{\mathrm{m}}} ||C'(t)||\ \mathrm{d}t \tag{8.24}$$

Combining distance map coordinate and a nearest contour coordinate $\{d, l\}$, we have a unique representation of each point in the shape. We must define an origin $C(0)$ on the parametric contour to define the NAL. We used as reference point the highest point of the curve in the image. This reference point can move with rotating or articulated objects but it is not the case in the videos used in our experiments. There are many works in the literature to define more efficient contour correspondences between two deformable shapes [BMP02]. However, we did not yet implement these techniques, and even with this basic contour correspondence, our shape coordinates can handle many types of deformations.

This method Kullback with Distance Map and NAL Geometric data (KL-DG) has 5-dimensional features: $\{Y, U, V, d, l\}$. The last two features are plotted on a example on Fig. 8.5. The two discontinuities visible on this figure are on the skeleton of the region as the NAL changes. The other discontinuity is on the top of the head as it is the origin of the arc length.

Finally, this method can be seen as a joint distribution of radiometric data with a shape prior. While shape priors energy are often internal as they do not depend on the image (they contains no radiometric information). Shape priors geometrically match two regions [RP02] and the idea is to add radiometric information in this geometric match.

Weighting between components

One could claim our method is parameter-free as the use of joint probability allows no difficult weightings considerations (in comparison with quadratic error of different physical quantities). However, one may still want to define weightings between color and spatial components. On one hand, if there is knowledge on the rigidity on the object, one can increase the weightings on the spatial components, allowing more variability in color changes. On the other hand, if the object is articulated, one can lower the weighting of spatial components, relying more on color distributions.

To tune this parameter, we rescale spatial features, shape coordinates, into the interval $[0, 1]$, and we rescale the color features on the interval $[0, \alpha]$, α being the weighting parameter. Moreover, for the DG coordinates system, if we assume the region is a circle of radius r, the maximum of the distance map would be r while the maximum of the NAL would be $2\pi r$. To circumvent this, we divide the NAL coordinates by 2π.

8.3.4 Segmentation using active contours

Estimation of entropy and probability distributions

The energy used is the KL divergence. As developped in (8.21), it is a sum of two entropies. In this section we present a method to estimate entropy.

Ω_U defining dataset $U = \{u(\mathrm{m}), \mathrm{m}\}$ for m in Ω_U (U being either T or R, Ω_U being either Ω_T or Ω_R) has the following Shannon entropy:

$$H(X_U) = -\int_{\mathbb{R}^d} f_U(\alpha)\ \log f_U(\alpha)\ \mathrm{d}\alpha \tag{8.25}$$

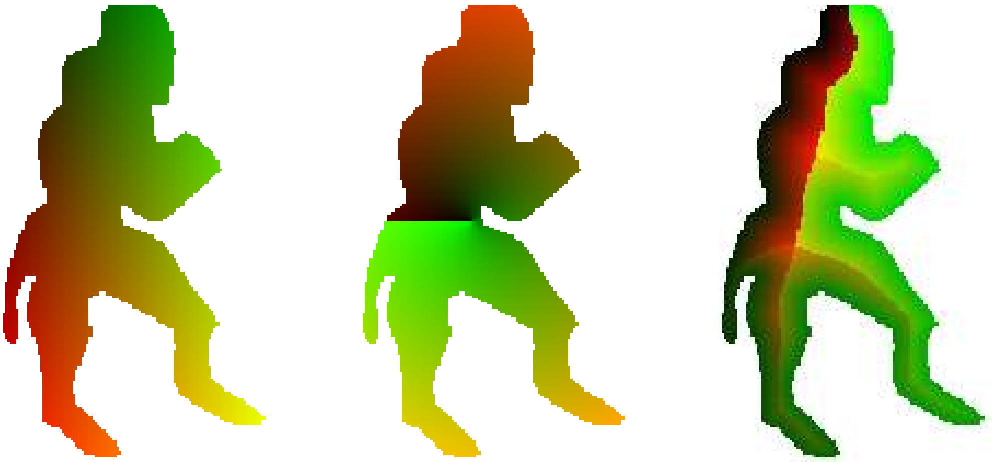


Figure 8.5: *Spatial features components $\{x, y\}$, first component red color, second component green color: $\{0, 0\}$ black, $\{1, 0\}$ red, $\{0, 1\}$ green, $\{1, 1\}$ yellow. From left to right: region Cartesian coordinates (CG), region polar coordinates (PG), distance Map and NAL (DG)*

Again, we used the Ahmad-Lin estimate [AL76], named resubstitution estimate of entropy

$$\hat{H}_{\mathrm{AL}}(f_U) = -\frac{1}{|U|}\int_{\Omega_U} \log f_U(u)\ \mathrm{d}u \tag{8.26}$$

where $|U|$ is the area of Ω_U. Since the actual PDF f_U is unknown, it must be estimated. A common practice is to use the non-parametric, Parzen windowing method.

$$\hat{f}_U(u) = \frac{1}{|U|}\int_{\Omega_U} K_h(u - \mathrm{m})\ \mathrm{dm} \tag{8.27}$$

where K_h is a multivariate, Gaussian kernel with standard variation, or bandwidth, h.

Shape derivative

The energy to be minimized through active contours is the Kullback divergence (8.21). In addition, as the distribution of the object can be characterized by a subregion inside the object, we propose to add a maximum area constraint with a weighting λ.

$$E(T) = D_{\mathrm{KL}}(T, R) - \lambda.|T| \tag{8.28}$$

To differentiate this energy we write its expression (8.26).

$$\begin{aligned} E(T) &= -\frac{1}{|T|}\int_{\Omega_T} \log f_T(t(\mathrm{m}))\ \mathrm{dm} \\ &- \frac{1}{|T|}\int_{\Omega_T} \log f_R(t(\mathrm{m}))\ \mathrm{d}m - \lambda.|T| \end{aligned} \tag{8.29}$$

We have an energy which has many dependencies with the region Ω_T. We propose to compute the shape derivative [ABFJB03] on Ω_T in the direction of a vector field V. Computing the derivative, we obtain:

$$\begin{aligned} \mathrm{d}E(\Gamma,F)=\int_\Gamma\Bigg[&\quad \frac{1}{|T|}(D_{\mathrm{KL}}(T,R)-\log f_T(t(s))+\log f_R(t(s)))+\lambda \\ &+\quad \frac{1}{|T|^2}\int_{\Omega_T}1-\frac{K_h(t(\mathrm{m})-t(s))}{f_T(t(\mathrm{m}))}\,\mathrm{dm}\Bigg]N(s).F(s)\,\mathrm{d}s \end{aligned} \tag{8.30}$$

where $\Gamma=\partial\Omega_T$ and where N is the inward unit normal of Γ. It can again be written as a scalar product.

$$\mathrm{d}E(\Gamma,F)=\int_\Gamma((\alpha(s))\;N(s))\cdot F(s)\,\mathrm{d}s=\langle\alpha\;N,F\rangle \tag{8.31}$$

where $\langle,\rangle$ is the L^2-inner product on Γ.

Based on the notion of gradient defined in (2.42), energy (8.28) can be minimized using a steepest descent procedure in the space of contours. The following contour evolution process is known as the active contour technique [CKS97, HR04]: an initial contour[6] is iteratively deformed in the opposite direction of the gradient until a convergence condition is met. The evolution equation of the active contour is written as follows

$$\begin{cases}\Gamma(\tau=0)=\Gamma_0\\ \dfrac{\partial\Gamma}{\partial\tau}=-\alpha\;N\end{cases} \tag{8.32}$$

where τ is the evolution parameter. The convergence condition is $\alpha=0$. This evolution equation is implemented using explicit parametrization of active contours, *i.e.* a smoothing spline.

One can note that Eq. (8.30) requires the PDF estimation of a 5-dimensional joint geometric/radiometric data set (three color components plus two spatial components) of the reference region and the target region: R and T. The sparsity of this high-dimension data space makes the PDF estimation, and therefore the similarity measure estimation, even more problematic. The kNN framework can be applied as it is the same shape derivative with balloon variable kernels, it provides again an advantageous alternative to circumvent high dimensions.

Simplification in active contours using kNN

Using for f_R and f_T the kNN expression given in (5.5) and for D_{KL} the expression given in (5.12), expression (8.30) reduces to :

$$\begin{aligned} \mathrm{d}E(\Gamma,F)=\int_\Gamma\Bigg[&\quad -\frac{d}{|T|}[\mu_T(\log\rho_k(R))-\mu_T(\log\rho_k(T)) \\ &-\quad \log\rho_k(R,t(s))+\log\rho_k(T,t(s))]+\lambda \\ &+\quad \frac{1}{|T|}(1-\frac{1}{k}\sum_{t_i\in\mathcal{N}_B(T,t(s))}\left(\frac{\rho_k(T,t_i)}{\rho_k(T,s)}\right)^d)\Bigg]N(s).F(s)\,\mathrm{d}s. \end{aligned} \tag{8.33}$$

where $\mathcal{N}_B(T,t(s))$ is the balloon neighborhood of sample $t(s)$ in data set T, *i.e.* the support of $K_{h(s)}$, which in the kNN framework, is a uniform kernel centered in s of size $\rho_k(T,s)$ (5.5). Moreover one can note that choosing the sample point estimate expression (5.6) instead of the balloon estimate expression (5.5), the last row in (8.33) would be equal to zero. However both simplified expression of Kullback distance (5.12) and simplified expression of PDF estimate (5.5) are only valid for balloon estimate.

[6] For example, a user-defined contour.

$$
\begin{aligned}
\mathrm{d}E(\Gamma, F) = \int_{\Gamma} \Bigg[& -\frac{d}{|T|}[\mu_T(\log \rho_k(R)) - \mu_T(\log \rho_k(T)) \\
- & \log \rho_k(R, t(s)) + \log \rho_k(T, t(s))] + \lambda \\
+ & \frac{1}{|T|}(1 - \frac{1}{k} \sum_{t_i \in \mathcal{N}_{SP}(T, t(s))} 1) \Bigg] N(s).F(s) \; \mathrm{d}s.
\end{aligned}
\tag{8.34}
$$

Finally, kNN version of energy derivative (8.33) is plugged in the evolution equation (8.32). Let us remind active contour energy (8.28) and active contour evolution equation (8.32) required a high dimensional joint PDF over the data. The kNN framework estimates both active contour energy (8.28) and active contour evolution equation (8.32) without explicit estimation of the PDF but with a reduced expression using distances to nearest neighbors.

8.3.5 Experimental results

In this section we will compare two methods, the Kullback distance computed through kNN but with no geometry kNN-KL (no spatial features, R and T are 3-dimensional) and the Kullback distance computed through kNN with geometry in a general sense kNN-KL-G (spatial features, R and T are 5-dimensional), geometry being either Cartesian kNN-KL-CG, polar kNN-KL-PG or distance map with NAL kNN-KL-DG. k of the kNN framework is set to $\sqrt{|T|}$.

The reference histograms for kNN-KL and kNN-KL-G are built over a region Ω_R on frame 1 for “Erik” Fig. 8.6, frame 74 for “Football” Fig. 8.7 using a manual segmentation. The goal is to find the corresponding region Ω_T in frame 6 for “Erik”, frame 75 for “Football”. We initialize the segmentation with a circle far from the solution to show the stability of the method.

First we present results on sequence “Erik” Fig. 8.6, size 288×352. This sequence shows a translating man over a static background. This sequence was chosen because its motion is very simple, while it is composed of many colors which will lead to complex color histograms. This sequence is considered as rigid, we tuned the weighting α presented in Section 8.3.3 to 1. This means, an error of 1 unit in geometry is similar to an error of 1 color intensity. Some parts of the background have similar colors than Erik. Therefore kNN-KL includes it as object while kNN-KL-G detects their spatial features are not correct so it does not include it as object. These results did not use maximum area constraint, $\lambda = 0$. As expected for rigid objects, all spatial features kNN-KL-CG, kNN-KL-PG and kNN-KL-DG led to exactly the same results, which is the correct segmentation.

Results are presented on sequence “Football” Fig. 8.7, size 288×352. This sequence shows fast and articulate motions. This sequence is considered as nonrigid, we tuned the weighting α presented in Section 8.3.3 to 10. This means, an error of 10 units in geometry is similar to an error of 1 in color intensity, a good motion variability is allowed in this sequence. Some parts of the public on the upper part of the video have the same colors as the player. kNN-KL-G excludes again them as their spatial features are not correct while kNN-KL includes them in the segmentation. The Kullback distance kNN-KL-G slightly increases when taking the legs of the player as their are articulated (error of registration in the spatial features). However, as the geometric constraint is soft, it increases less than with segmenting the public, the player is then correctly segmented with the help of maximum area constraint. Here the results with different spatial features are not the same. kNN-KL-CG has difficulties to properly segment the legs of the player as it is nonrigid (spatial features have changed). kNN-KL-PG is a little better as the spatial features change only on one spatial feature components (the angle).

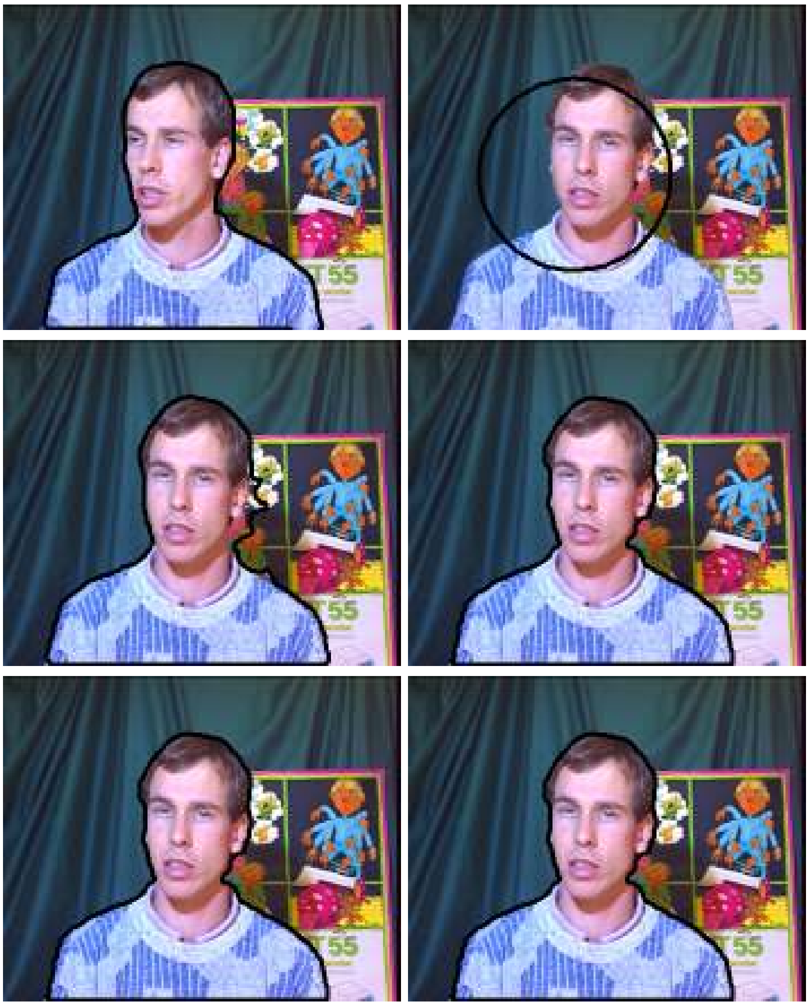


Figure 8.6: *Segmentation on sequence "Erik" on frame* 6*: (from left to right and top to bottom) region of interest* Ω_R *manually segmented on frame* 1*, initialization of the segmentation, results* Ω_T *of segmentation with method kNN-KL, results* Ω_T *of segmentation with method kNN-KL-CG,results* Ω_T *of segmentation with method kNN-KL-PG,results* Ω_T *of segmentation with method kNN-KL-DG*

Finally kNN-KL-DG gives the best results as the spatial features on the reference frame and on the target frame represent the same pixels.

The maximum area constraint was tuned to segment the whole object in all cases. kNN-KL and kNN-KL-CG required a parameter $\lambda = 2.10^{-4}$ to ignore color variability for kNN-KL, and to ignore spatial features variability for kNN-KL-CG. It segmented all the football player but it led to an over-segmentation in some parts of the image. kNN-KL-PG only needed a parameter $\lambda = 1.10^{-4}$ to segment all the football player, leading to less over-segmentation. Finally kNN-KL-DG needed a parameter $\lambda = 5.10^{-5}$ to segment all the football player, leading to accurate segmentation.

Finally we discuss about the robustness to the bandwidth parameter k of kNN. $k = \sqrt{|T|}$, setting k to $2^n\sqrt{|T|}$, from $n = -2$ to $n = 2$, the absolute difference between the segmentation masks and the one generated is less than $3.7\% \times |T|$. The segmentation algorithm is robust to the bandwidth k, changing from 1 time to 16 times its initial value, while segmentation methods based on Parzen techniques observed an high sensitivity to the bandwidth parameter.

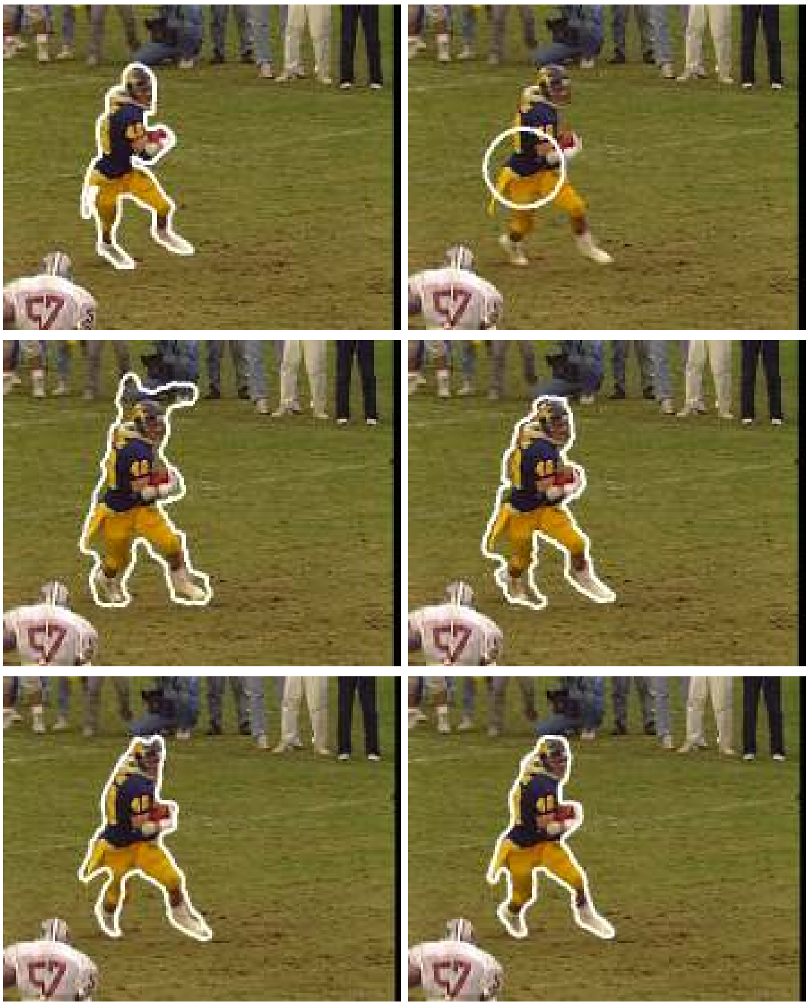


Figure 8.7: *Segmentation on sequence "Football" on frame 75: (from left to right and top to bottom) region of interest Ω_R manually segmented on frame 74, initialization of the segmentation, results Ω_T of segmentation with method kNN-KL, results Ω_T of segmentation with method kNN-KL-CG,results Ω_T of segmentation with method kNN-KL-PG,results Ω_T of segmentation with method kNN-KL-DG*

8.3.6 Conclusion

The results presented for this method show the applicability of kNN framework in active contour segmentation. It shows that it can be particularly efficient in high dimensional cases such as joint feature-spatial segmentation. We proposed new shape coordinates for deformable regions. The results tend to show that for rigid object, shape coordinates are not an issue. On the opposite, for deformable regions, our region shape coordinates compares favourably to other region coordinates systems.

We only compared soft geometric to no geometric, and different shape coordinates. One could expect comparisons with classical Parzen techniques for segmentation. First, a comparison would not be fair as the model presented in [LFGWI00] is only valid for medical images as discussed in Section 8.3.3. Second, this model uses only grayscale and distance map feature, namely 2-dimensional features. This model is acceptable for Parzen, but when using more high dimensional features (5-dimensional features in our model) Parzen techniques suffer from the curse of dimensionality while kNN can handle it. We refer to [TS92] for details on this subject.

Finally our method compares favorably to state of the art, on sequence “Football”. In this section we compared to a radiometric method. Geometric methods often require a motion estimation step. An affine motion model cannot model the high level of deformations of the articulated player. In addition the sequence has motion blur and there are fast motions. As a consequence, optical flow estimation is also very difficult on this sequence.

8.4 Motion segmentation

In this section we demonstrate how motion segmentation can be simply performed by adding a motion cue in the vectors. Color and motion residual are combined in a joint entropy. Motivations to treat color and residual independently are also given. Indeed, even if the kNN framework allows to efficiently deal with higher dimensional energies. It is sometimes of less interest when data are assumed enough decorrelated (in a meaning defined in this section). In this section, Parzen estimation will then be preferred as it is fast and efficient in low dimensions.

8.4.1 Introduction

Video segmentation aims at partitioning some video frames into objects and background. (For simplicity, it will be supposed that there is a single object.) This task can be performed without motion computation. If reference values of some descriptors are available (*e.g.*, mean color, color variance, color distribution... of the object of interest), an object can be segmented by minimizing a distance between the actual values of the descriptors computed on a candidate object domain and the reference values [FZ04]. However, the lack of sensitivity of some descriptors near the object boundary (*e.g.*, the color distribution might not vary significantly if the candidate domain is slightly deformed) and the degree of freedom of the object motion (*a priori* infinite) may increase the number of potential solutions. Therefore, the segmentation framework involving motion computation will be considered.

Let us first consider the motion estimation task. Dense flow field estimation (*i.e.*, one motion vector per pixel) is an underdetermined problem. Moreover, when using the first order approximation of the brightness/color constancy constraint, only the motion component in the direction of the image gradient can be estimated. This limitation is known as the aperture problem. Motion estimation is therefore an ill-posed problem. It needs to be regularized, *i.e.*, constrained. On the one hand, the so-called global methods estimate a dense flow field while imposing the solution to be smooth [WS01, BBPW04]. On the other hand, local methods constrain the motion to follow a parametric model (*e.g.*, translation, affine motion, homography) with constant parameters, either in the whole image or within blocks or regions [LK81, OB95, WK93]. Both approaches have also been combined [BWS05]. Given the link between motion estimation and object segmentation in a video, it can be noted that global methods require anisotropic smoothing to preserve object boundaries [WS01] whereas local methods are characterized by a chicken-and-egg dilemma: *(i)* estimating motion knowing the object boundary while *(ii)* the boundary is defined as an optimal partition knowing the motion of the object and its neighborhood. This suggests to perform motion estimation and segmentation jointly [CS05], which will be the approach followed here.

Focusing on motion estimation again, imposing the brightness/color constancy constraint is equivalent, in variational terms, to minimizing a function of the motion compensation error (MCE), or of its first order approximation, as already mentioned. There is a correspondence between the choice of one such function and an assumption on the distribution of the MCE, *e.g.*, the square function and the assumption of a Gaussian distribution or the absolute value [WK93] and the assumption of a Laplacian distribution. This point of view will be referred to as parametric since the underlying distribution is characterized by a small set of parameters. In contrast, it is proposed to get rid of the parametric assumption on the data by trying to estimate the actual distribution as proposed for various related problems [CRM00, ABFJB03, KFY^{+}05, MP04] or in the context of shape prior [COS06, LGF00]. This approach will be referred to as nonparametric.

In this nonparametric framework, we propose to use a unique statistical measure to both estimate the motion and segment the object. (A review of statistical methods in image segmentation was done recently [COS06].) Among the popular measures such as entropy [CH04], mutual information [VW97], or the Kullback-Leibler divergence [Kul59], the entropy [BHDB06, HBDB06] was chosen for its interesting properties (it is a measure of dispersion and it is robust to outliers - see Section 8.4.3) and because manipulating a single distribution (the distribution of the MCE) was preferred over taking the reference/target distribution comparison approach.

Motion-based segmentation can fail in areas insufficiently textured. In particular, the MCE is equal to zero in any homogeneous region. Therefore, adding such a region to, or subtracting it from, a given segmentation still produces potential solutions. This can be solved with the help of shape regularization [CS05], by adding spatial terms to the motion-based energy [BRDW03, PD02a], or by processing color and motion sequentially [DPM06]. This last alternative is interesting but asks the difficult question of ordering the features, say, by importance (especially if involving even more features). The first two ones often require a non-trivial adjustment of the weighting of the different terms. It will be shown that, using joint distributions, an objective choice of the weighting of the motion term and the spatial term can be made (namely, equal weighting or, equivalently, weight-free).

In brief, we propose to define a single spatio-temporal energy[7] to perform joint motion estimation and segmentation. To account for noise and model mismatch, the energy will be based on a statistical measure, namely entropy. In order to adapt to the data, no assumption will be made on the MCE or color distributions; they will be estimated using a nonparametric method. Finally, it will be shown that, with the proposed approach, the motion term need not to be weighted relative to the spatial term. In a way, this offers a solution to the implementation of the operator AND between several properties (related here to motion "and" color) jointly describing the object of interest.

The section is organized as follows: Section 8.4.2 details the problem statement. In Section 8.4.3, the classical parametric assumption on the MCE distribution is discarded and the proposed nonparametric framework for video segmentation, involving the actual residual and color distributions, is described. A single spatio-temporal energy is proposed to perform mo-

[7] Note that, here, the usage of the terms "temporal" and "spatio-temporal" should be understood as "motion-based" and "based on motion and color", respectively. "Spatio-temporal" more typically refers to a process performed in the xyt-space where x and y are video frame coordinates and t is the time coordinate. As far as active contours are concerned, such a process would manipulate a tube oriented along the time dimension as opposed to a planar curve here.

tion estimation and motion-based segmentation simultaneously. A piecewise motion model is introduced to allow enough flexibility for segmenting articulated objects. An active contour procedure is proposed in Section 8.4.4 to minimize the energy. Finally, Section 8.4.5 presents some results on synthetic and natural video sequences.

8.4.2 Problem Statement

The motion of an object domain Ω can be computed by choosing a motion model and finding the motion parameters that minimize a function of the MCE over Ω. At a pixel level, making the assumption of brightness/color constancy, the MCE is classically equal to the following residual

$$e_n(v(\mathrm{m}), \mathrm{m}) = I_n(\mathrm{m}) - I_{n+1}(\mathrm{m} + v(\mathrm{m})) \tag{8.35}$$

where m is a pixel of Ω, I_n is the n^{th} grayscale or color frame of the sequence, and $v(\mathrm{m})$ is the apparent motion between I_n and I_{n+1} at m (known as the optical flow). Ideally, $e_n(v(\mathrm{m}), \mathrm{m})$ is equal to zero up to some noise. In grayscale, this condition provides a single equation for two unknowns (the components of $v(\mathrm{m})$) and, both in grayscale and color, it is likely that several pixels y have the same value $I_{n+1}(y)$. As a consequence, the motion estimation problem cannot be solved without additional constraints. A possible way to constrain the problem is to assume that the motion is coherent with a chosen model inside Ω [WK93]. Then, the motion estimate v can be computed as

$$v = \arg\min_{w} \int_{\Omega} \varphi(e_n(w, \mathrm{m}))\, \mathrm{dm} \tag{8.36}$$

where φ can be, for example, the square function, the absolute value, or a function typical of the robust estimation framework [BA96, CBFAB97].

The motion-based segmentation of frame I_n can be formulated as the largest domain Ω inside which the motion is coherent with model (8.36), formally,

$$\begin{cases} \hat{\Omega} = \arg\min_{\Omega} \int_{\Omega} \varphi(e_n(v(\Gamma), \mathrm{m}))\, \mathrm{dm} \\ v(\Gamma) = \arg\min_{w} \int_{\Omega} \varphi(e_n(w, \mathrm{m}))\, \mathrm{dm} \end{cases} \tag{8.37}$$

where Γ is the boundary $\partial\Omega$ of Ω. Note that writing $v(\Gamma)$ or $v(\Omega)$ is only a matter of notation since Ω is completely determined by Γ and conversely. Let us denote by E_t the following domain energy[8]

$$E_t(\Gamma) = \int_{\Omega} \varphi(e_n(v(\Gamma), \mathrm{m}))\, \mathrm{dm} \ . \tag{8.38}$$

Choosing φ results in making an assumption on the distribution of the residual e_n in Ω. However, these assumptions may not be appropriate. In particular, the presence of outliers in the residual (*e.g.*, due to occlusions, mismatch between the chosen motion model and the actual motion, variation of luminance...) may result in a complex, multimode distribution. As a consequence, the motion estimator in (8.37) may be biased, leading to a loss of accuracy of the motion-based segmentation.

8.4.3 Proposed segmentation energy

Three steps will be taken to derive the proposed energy: the definition of an ideal energy, its simplification, and its “symmetrization”.

[8] Subscript t stands for *temporal*.

Nonparametric, entropy-based energy

To account for the true distribution of the residual e_n, and in general any feature that will be used for segmenting, it is proposed to make the energy depend on an estimation of the feature distributions rather than on the features themselves as it was the case in (8.38) concerning the residual. For the present segmentation task, the residual e_n will be combined with the spatial feature I_n (similar combinations of geometry and radiometry have been proposed [EDD03, LFGWI00]). The proposed energy has the following form

$$\begin{cases} E(\Gamma) = -\dfrac{1}{|\Omega|} \displaystyle\int_{\Omega} \log f(e_n(v(\Gamma), \mathrm{m}), I_n(\mathrm{m})) \, \mathrm{dm} \\ v(\Gamma) = \arg\min\limits_{w} E_{\Gamma}(w) \end{cases} \tag{8.39}$$

where f is the joint distribution of the residual $e_n(v(\Gamma))$ and the image color I_n inside the object domain Ω, and

$$E_{\Gamma}(w) = -\frac{1}{|\Omega|} \int_{\Omega} \log f(e_n(w, \mathrm{m}), I_n(\mathrm{m})) \, \mathrm{dm} \, . \tag{8.40}$$

Energy (8.39) is the continuous version of the Ahmad-Lin approximation of differential entropy [AL76]. In both (8.39) and (8.40), f is the joint distribution of the residual and the color. The residual being a function of the motion, f is itself a function of $v(\Gamma)$ in the former and w in the latter.

Let us see why this choice of energy is interesting. First, entropy is a measure of dispersion. If the segmentation is optimal, the residual should be distributed around zero with a minimal dispersion. Similarly, if the object is assumed to be piecewise homogeneous, the color distribution has a small dispersion. Thus, a minimum entropy criterion should have near optimal performances in case of a parametric distribution while being able to adapt to nonparametric cases. In particular, entropy appears to be less sensitive to outliers in practice.

Simplification using marginal distributions

A fixed-size kernel-based procedure will be employed to estimate the distributions (see Section 8.4.3). To avoid that the entropy estimation be biased as an effect of the curse of dimensionality, energy (8.39) will be “simplified”. Thus, the residual and the color will be assumed to be independent (See Appendix F.2). As a consequence, energy (8.39) can be rewritten as the following sum involving the marginal distributions

$$\begin{aligned} E(\Gamma) &= -\frac{1}{|\Omega|} \int_{\Omega} \log f_t(e_n(v(\Gamma), \mathrm{m})) \, \mathrm{dm} - \frac{1}{|\Omega|} \int_{\Omega} \log f_s(I_n(\mathrm{m})) \, \mathrm{dm} && (8.41) \\ &= E_t(\Gamma) + E_s(\Gamma) && (8.42) \end{aligned}$$

where subscript t, respectively s, in f_t and E_t, respectively f_s and E_s, stands for temporal, respectively spatial. Note that the second integral in (8.41) was proposed for image segmentation [KFY$^+$05].

The temporal energy in (8.41) is of the form

$$E_t(\Gamma) = \int_{\Omega} \psi(f_t(e_n(v(\Gamma), \mathrm{m}))) \, \mathrm{dm} \, . \tag{8.43}$$

One can say that the parametric approach (8.38) is extended to nonparametric distributions by substituting for a function of the residual $\varphi(e_n)$ a function of its distribution $\psi(f_t(e_n))$.

By making the assumption of independence, one obtains a sum of two energies, meeting the philosophy usually adopted when one wants to simultaneously minimize several energies. However, in general, weighting parameters are introduced to tune the influence of the respective energies whereas, here, there are no such weights (it seems indeed natural not to favor any of the two terms since they have the same *unit*).

Similarly to section 8.2 a region competition framework is defined justified by conditional entropy.

$$E_{\mathrm{rc}}(\Gamma) = \frac{|\Omega|}{|D|}\, E_{\mathrm{np}}(\Gamma) + \frac{|\overline{\Omega}|}{|D|}\, E_{\mathrm{np}}(\Gamma^c) \tag{8.44}$$

As defined in (8.39), energy $E(\Gamma)$ is (an approximation of) the joint entropy of the residual and the color conditional on $C = 1$. Let us denote it by $H(e_n, I_n|C = 1)$. Equivalently, $E(\Gamma^c)$ is equal to $H(e_n, I_n|C = 0)$. Then, Eq. (8.8) is equal to

$$\begin{aligned} E_{\mathrm{rc}}(\Gamma) &= \sum_{i \in \{0,1\}} p(C = i)\; H(e_n, I_n|C = i) && (8.45) \\ &= H(e_n, I_n|C)\ . && (8.46) \end{aligned}$$

where C is the characteristic function of the object and $p(C = i)$ denotes the probability of the event $C = i$. Therefore, energy (8.44) is equal[9] to the conditional, joint entropy of the residual and the color $H(e_n, I_n|C)$.

Motion estimation

As mentioned in Section 8.4.2, the motion v is assumed to follow a given model inside Ω. For example, it can be defined by a set of parameters p [OB95]. Then, estimating v in (8.43) is only a matter of estimating p. This task is certainly made easier if the relation between v and p is linear

$$v(\Gamma) = M\; p(\Gamma) \tag{8.47}$$

where M is a $2 \times l$ matrix if p is an l-vector. Even if the motion model is complex, it will hardly account for general motions such as motions of articulated objects, and if it does, solving for the model parameters is likely to be an ill-posed inverse problem. Instead, we propose to keep the model simple while solving for its parameters locally. Frame I_n is divided into k blocks B_i of identical size, where k depends on the frame size. Let Ω_i be the intersection of Ω with B_i and let Γ_i be the boundary $\partial\Omega_i$ of Ω_i (see Fig. 8.8). The temporal energy (8.43) is replaced with

$$E_t^{\mathrm{local}}(\Gamma) = -\frac{1}{|\Omega|} \int_{\Omega} \log f_t(e_n(v_1, \ldots, v_k, \mathrm{m}))\ \mathrm{dm} \tag{8.48}$$

where v_i is a short notation for $v(\Gamma_i)$, the motion of Ω_i. (The consequence of using this local approach is discussed in Appendix F.4.) In this context, the motion model can simply be translation. Therefore, Eq. (8.35) is replaced with

$$e_n(v_i, \mathrm{m}) = I_n(\mathrm{m}) - I_{n+1}(\mathrm{m} + v_i),\ \mathrm{m} \in \Omega_i\ . \tag{8.49}$$

This local approach will be used when the object of interest is articulated (see Section 8.4.5). In the other experiments, a global translation will be used. It corresponds to decomposing I_n into a single block B_1 covering the whole frame. Note that in the following,

[9] In fact, would be equal if the assumption of independence between e_n and I_n had not been made (see Section 8.4.3).

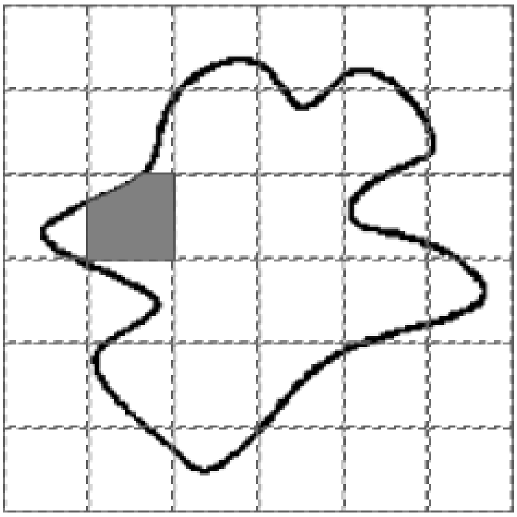


Figure 8.8: *Solid line: contour $\Gamma = \partial\Omega$; Dashed blocks: B_i; Gray-filled block: an example of a domain Ω_j with boundary Γ_i.*

for clarity, the notations E_t and $e_n(v(\Gamma), \mathrm{m})$ will be preferred over E_t^{local} and $e_n(v_1, \ldots, v_k, \mathrm{m})$, respectively.

Finally, to minimize the influence of occlusions, expression (8.49) is regarded as the forward residual and compared with the backward version as follows

$$e_n(v, \mathrm{m}) = \mathrm{min}_{\mathrm{abs}}\{I_n(\mathrm{m}) - I_{n+1}(\mathrm{m} + v),\ I_n(\mathrm{m}) - I_{n-1}(\mathrm{m} - v)\} \tag{8.50}$$

where $\mathrm{min}_{\mathrm{abs}}$ is equal to

$$\mathrm{min}_{\mathrm{abs}}\{a, b\} = \left\{ \begin{array}{lll} a & \text{if} & \min\{|a|, |b|\} = |a| \\ b & \text{if} & \min\{|a|, |b|\} = |b| \end{array} \right. . \tag{8.51}$$

Function 8.51 is not differentiable. However, in the present work, it does not need to be differentiated (see Appendix F.3).

Distribution estimation

Parzen windowing is a classical distribution estimation procedure [Par62]. The following continuous version was used

$$f(r) = \frac{1}{|\Omega|} \int_{\Omega} K_h(r - g(\mathrm{m}))\, \mathrm{dm} \tag{8.52}$$

where $|\Omega|$ is the measure of Ω, K_h is a Gaussian kernel with zero mean and a variance equal to h^2, and g is a random variable whose distribution is to be estimated (*i.e.*, $e_n(v(\Gamma))$ or I_n). It is usual to adapt h^2 to the data [Sil86, Sco92].

8.4.4 Segmentation using active contours

Shape gradient of the energy

Minimization of energy (8.41) requires the computation of its derivative with respect to Γ. There exists an infinite number of ways of deforming Γ. The shape derivative [DZ01, HR04, JBBA03, ABFJB03] of (8.41) can be interpreted as the derivative in a direction F, a vector field defined on Γ. It can be shown that the shape derivative of (8.43) is equal to (see Appendix F.3)

$$\begin{aligned} \mathrm{d}E_t(\Gamma, F) &= \frac{1}{|\Omega|} \int_\Gamma \Bigg[\log f_t(e_n(v(\Gamma), s)) - 1 + E_t(\Gamma) \\ &+ \frac{1}{|\Omega|} \int_\Omega \frac{K_h(e_n(v(\Gamma), s) - e_n(v(\Gamma), \mathrm{m}))}{f_t(e_n(v(\Gamma), \mathrm{m}))} \, \mathrm{dm} \Bigg] N(s) \cdot F(s) \, \mathrm{d}s \end{aligned} \tag{8.53}$$

where N is the inward unit normal of Γ.

Note that the distribution f_t appears explicitly in (8.53), hence the necessity to estimate it.

The expression of $\mathrm{d}E_s$ is similar to (8.53) (see Appendix F.3). Finally, the shape derivative of (8.41) is equal to

$$\mathrm{d}E(\Gamma, F) = \mathrm{d}E_t(\Gamma, F) + \mathrm{d}E_s(\Gamma, F) \; . \tag{8.54}$$

The shape derivative (8.54) has the following form

$$\mathrm{d}E(\Gamma, F) = \int_\Gamma ((\alpha_t(s) + \alpha_s(s)) \; N(s)) \cdot F(s) \, \mathrm{d}s = \langle \alpha \; N, F \rangle \tag{8.55}$$

where $\langle , \rangle$ is the L^2-inner product on Γ. Therefore, $\alpha \; N$ is, by definition, the gradient of (8.41) at Γ associated with this inner product.

Region competition

The shape derivative of (8.44) can be obtained by applying the traditional differentiation rule $(u \; v)' = u' \; v + u \; v'$ and determining the shape derivative of $|\Omega|$ (see Appendix F.3). The terms related to the object and the terms related to the background can be gathered together by noting that Γ and Γ^c are identical up to a change of orientation. In particular, the inward unit normal N^c of Γ^c is equal to $-N$.

Evolution equation

Based on the notion of gradient defined in Section 8.4.4, energy (8.41) can be minimized using a steepest descent procedure in the space of contours. The following contour evolution process is known as the active contour technique [CKS97, HR04]: an initial contour[10] is iteratively deformed in the opposite direction of the gradient until a convergence condition is met. The evolution equation of the active contour is written as follows

$$\begin{cases} \Gamma(\tau = 0) = \Gamma_0 \\ \dfrac{\partial \Gamma}{\partial \tau} = (\alpha^c - \alpha) \; N \end{cases} \tag{8.56}$$

where τ is the evolution parameter and α^c has the same expression as α but is evaluated on $\overline{\Omega}$. The convergence condition is $\alpha^c - \alpha = 0$.

8.4.5 Experimental results

Test settings

As a reminder, the proposed segmentation energy has the following form

$$E_{\mathrm{rc}}(\Gamma) = |\Omega| \; E(\Gamma) + |\overline{\Omega}| \; E(\Gamma^c) \tag{8.57}$$

[10] For example, a user-defined contour.

where

$$E(\Gamma) = E_t(\Gamma) + E_s(\Gamma) \ . \tag{8.58}$$

For comparison purposes, energy (8.57) will also be used in two incomplete forms: when E_s is removed from the definition of E in (8.58), the energy will be called temporal energy; when E_t is removed from the definition of E, the energy will be called spatial energy. In its complete form, it was already defined as the spatio-temporal energy.

The tests were performed on synthetic and natural sequences composed of 300×300-pixel frames and `cif`[11] frames, respectively, all defined in the YUV-color space. The V channel was discarded. Therefore, the distributions of I_n and e_n are functions from $\mathbb{R}^2$ to $\mathbb{R}$ with support $[0, 255]^2$ and $[-255, 255]^2$, respectively. In computing e_n, $I_{n+1}(\mathrm{m} + v)$ was bilinearly interpolated. Independence between spatial and temporal information was assumed in Section 8.4.3 in order to write E as the sum of E_t and E_s. The computation of these two components was also simplified by assuming independence between the channels Y and U. As a consequence, E_t and E_s were themselves estimated as the sum of a Y-based entropy and a U-based entropy.

The standard deviation h of the Parzen kernel (see Eq. (8.52)) was adapted to the data[12] by using the empirical standard deviation $\hat{\sigma}$ of the residual or the color in Ω

$$h = 0.9 \min(\hat{\sigma}, \hat{p}/1.34)\ |\Omega|^{-1/5} \tag{8.59}$$

where $\hat{p}$ is the interquartile range of the data in Ω. Therefore, h should be regarded as a function of Ω. This would add some terms to the shape derivative dE since expression (E.27) would not be valid anymore. However, these terms can be neglected because h does not change significantly between two iterations of the active contour process.

As mentioned in Section 8.4.3, translation was chosen as the motion model. The motion estimation in (8.43) was performed by fast, suboptimal (as opposed to exhaustive) search [ZM00] within a search window of -12/+12 pixels in both directions and a quarter of a pixel precision. This procedure was used whether the motion was estimated globally in Ω or locally in each Ω_i.

In the following, “segmentation” refers to object detection with an initialization far from the solution (typically a circle) while “tracking” refers to object detection with an initialization obtained by translating by v_{global} the object contour as detected in the previous frame, where v_{global} is the motion of Ω computed by the suboptimal procedure described above.

Comparing spatial, temporal, and spatio-temporal energies

In this section, motion is estimated globally on Ω (see Section 8.4.3).

Synthetic sequences

Several synthetic sequences were designed by combining different textures and homogeneous areas with a given *motion scenario*: an object is translating horizontally by -3 pixels over a background translating horizontally by 1 pixel. Segmentation was performed with the spatial energy, the temporal energy, and the spatio-temporal energy (see Fig. 8.9). These results suggest that the temporal energy is adapted whenever there is texture. On the contrary, the spatial energy seems more reliable in homogeneous areas. Finally, the combination of temporal and spatial information appears appropriate for segmenting sequences that contain homogeneous areas, textured areas, or both.

[11] The frame size `cif` corresponds to 352×288 pixels.

[12] Adapting the kernel bandwidth to the data is known as a plug-in procedure [Sil86].

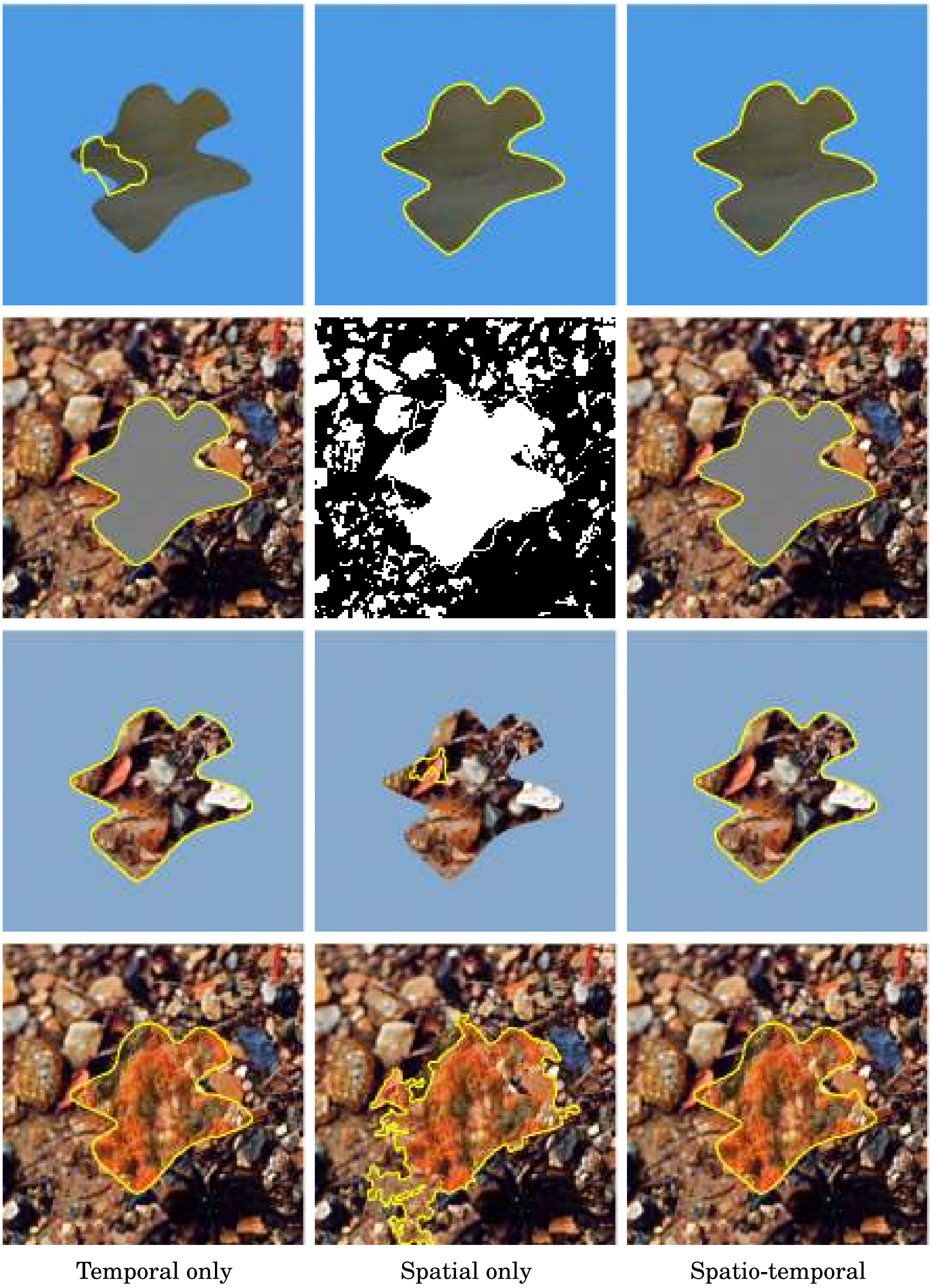


Figure 8.9: *Segmentation of synthetic sequences accounting for motion, color, or both. First row: homogeneous object over homogeneous background; second row: homogeneous object over textured background; third row: textured object over homogeneous background; last row: textured object over textured background.*

Standard test sequences

The same comparison as in Section 8.4.5 was performed with standard test sequences `'Flowers and garden'` and `'Soccer'`.

In sequence 'Flowers and garden', the sky bordering the tree is rather homogeneous (see Fig. 8.10). Therefore, oversegmentation occurs with the temporal energy, as noted in Appendix F.1. With the spatial energy, the segmentation process also fails because part of the houses in the background have colors similar to the tree. Finally, the spatio-temporal segmentation mostly excludes the sky since it has a different color (spatial information) and also excludes the houses since they have a different motion (temporal information).

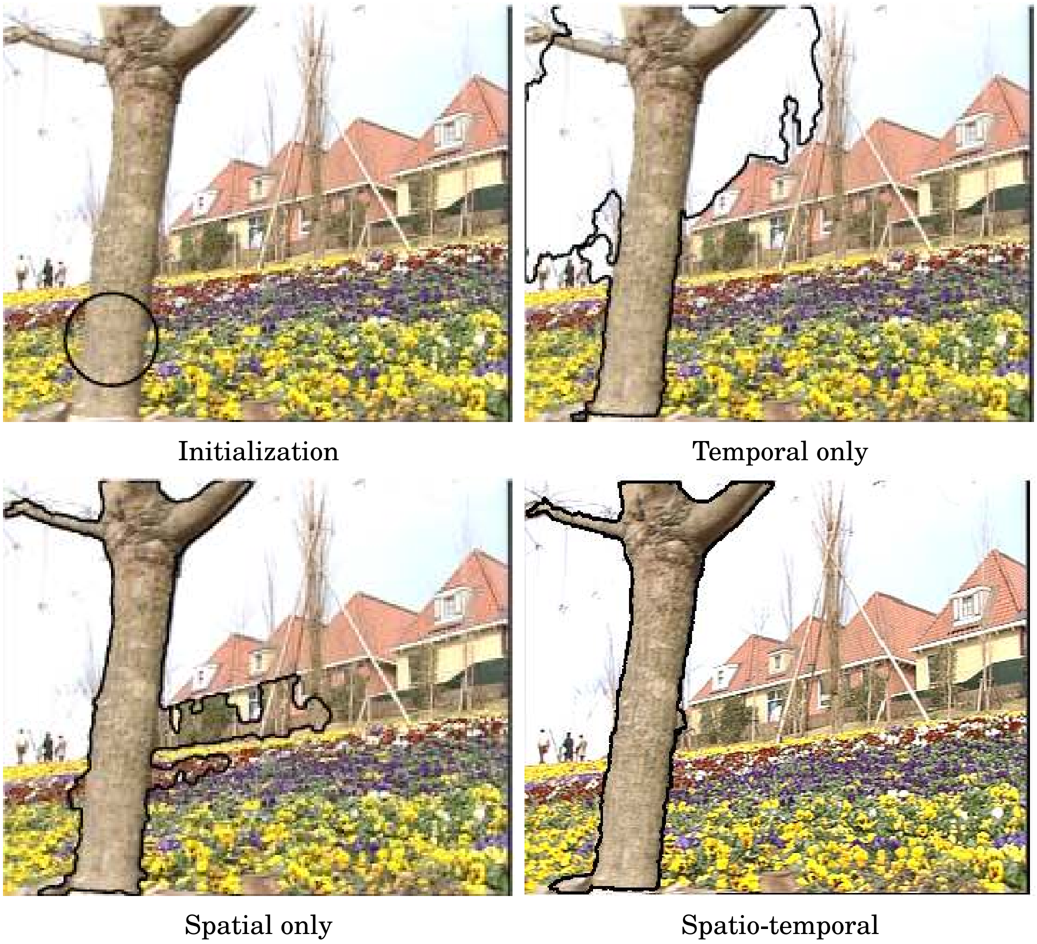
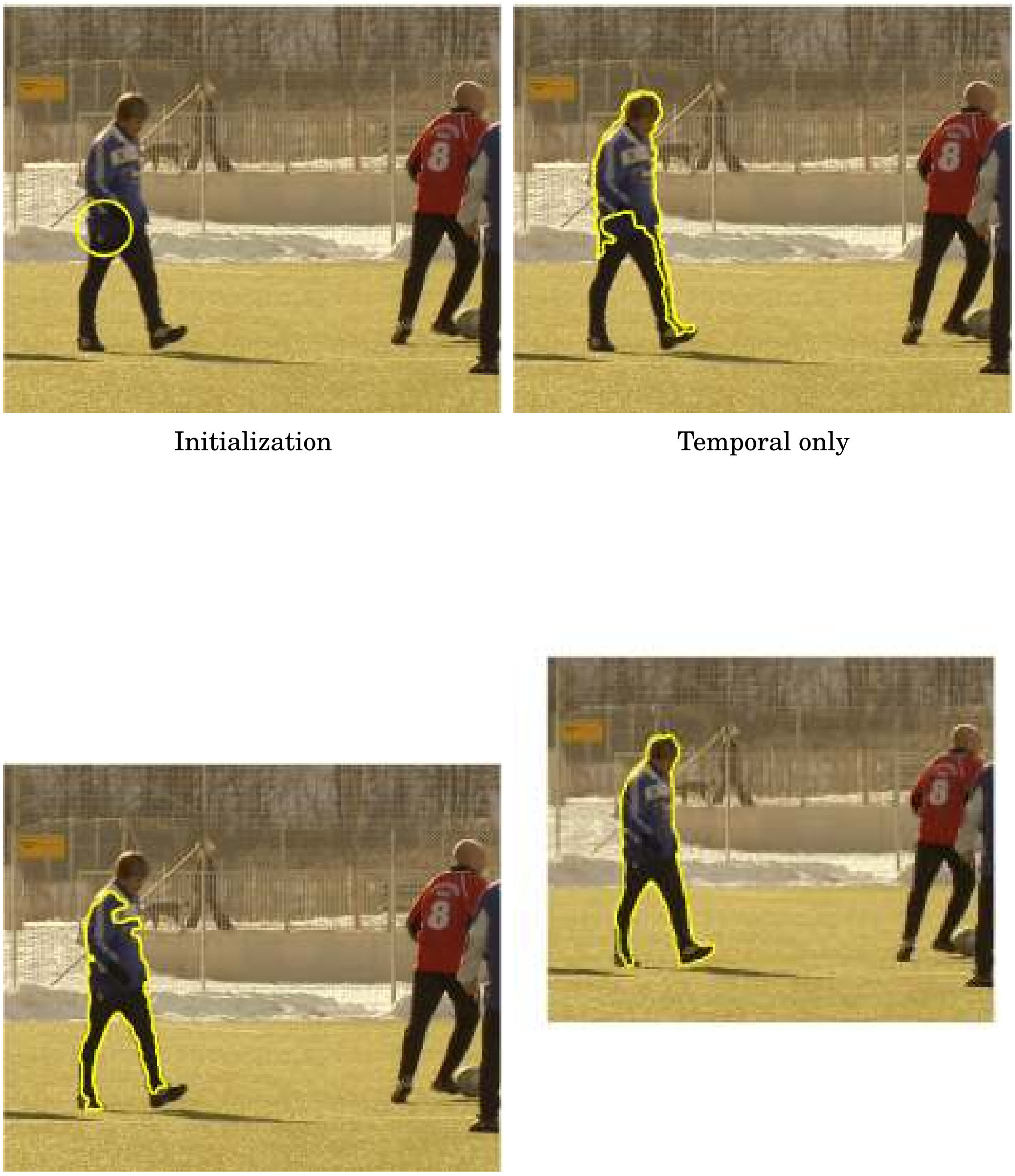


Figure 8.10: *Segmentation of frame 237 of sequence* 'Flowers and garden' *accounting for motion, color, or both.*

In sequence 'Soccer', the soccer player has a complex, articulated motion (see Fig. 8.11). The temporal energy only captures the rigid part of the body while the spatial energy does not capture the head as it has colors similar to the background. The spatio-temporal energy provides a good tradeoff, although it sometimes *misses* a foot of the player (see Fig. 8.16) for which both the temporal information and the spatial information (the color of the shoe is similar to background colors in the YU-color space) are unreliable. This satisfying result can be explained by the fact that the spatial energy helps the temporal term when the motion model mismatches the actual motion, and the temporal energy helps the spatial term when the color is not discriminating.



Initialization

Temporal only

Spatial only

Spatio-temporal

Figure 8.11: *Segmentation of frame 162 of sequence* `'Soccer'` *accounting for motion, color, or both.*

Parametric vs. nonparametric

In this section, motion is estimated globally on Ω (see Section 8.4.3).

One can wonder the practical benefits of relying on nonparametric estimations of the residual distribution and the color distribution as opposed to using classical error terms corresponding to parametric assumptions. For a fair comparison, the parametric assumptions for the residual and the color distributions have to be chosen appropriately. The residual is corrupted by outliers mainly due to noise, illumination variations, motion model mismatch, and occlusion. The Sum of Absolute Differences (SAD) [WK93] was chosen since it is robust to outliers. Note that it follows from a Laplacian assumption (see Appendix F.5.1).

It is clear that there is no ideal parametric assumption concerning the spatial term. Nevertheless, noting that the spatial entropy in (8.41) can be interpreted as a piecewise color homogeneity criterion, it seems reasonable to make the assumption of a Gaussian distribution[13] (see Appendix F.5.2).

The continuous form of criteria (F.32) and (F.34) can be linearly combined to define a parametric, space-time segmentation energy

$$E_p(\Gamma) = \int_\Omega (I_n(\mathrm{m}) - \mu_I(\Gamma))^2 \,\mathrm{dm} + \alpha \int_\Omega |I_n(\mathrm{m}) - I_{n+1}(x + v(\Gamma))| \,\mathrm{dm} \tag{8.60}$$

where

$$\begin{cases} \mu_I(\Gamma) = \int_\Omega I_n(\mathrm{m}) \,\mathrm{dm} / \int_\Omega \,\mathrm{dm} \\ v(\Gamma) = \arg\min_w \int_\Omega |I_n(\mathrm{m}) - I_{n+1}(x + w)| \,\mathrm{dm} \end{cases} . \tag{8.61}$$

and α is a positive constant. The nonparametric energy (8.41) does not weight the spatial term relatively to the temporal term. Therefore, to be coherent, α should be equal to one. However, the results on sequence `'Flowers and garden'` suggest to choose α larger than one (see Fig. 8.12). In each experiment, the optimal value was determined empirically. Moreover, to give an idea of the behavior of each term of the parametric energy (8.60), segmentation was also performed using each term separately (same procedure as in Section 8.4.5). The parametric approach was also tested on the other, more challenging sequence `'Football'`. Even when assigning a higher weight to the temporal term, the segmentation is not satisfying (see Figs. 8.13, 8.14, and 8.15).

In light of these results, three intuitive conclusions can be made. *(i)* As expected, when the parametric assumptions are roughly in accordance with the actual distributions (sequence `'Flowers and garden'`), the parametric approach can perform well. *(ii)* The Laplacian assumption for the residual distribution is more reliable than the Gaussian assumption for the color distribution. Indeed, with sequence `'Flowers and garden'`, the correct segmentation is obtained only when the temporal term is weighted significantly more than the spatial term. *(iii)* Again, as expected, when the parametric assumptions clearly mismatch the actual distributions for the motion being complex or the object and background being composed of several colors (sequence `'Football'`), the parametric approach fails, as opposed to the proposed nonparametric approach (see Figs. 8.10 and 8.11).

Tracking and piecewise motion estimation

In this section, an object of interest is tracked in two standard test sequences using the proposed method. In both sequences, the object of interest is composed of several colors and has a complex, articulated motion. Therefore, they are appropriate for comparing the global (on Ω) motion approach and the local (on Ω_i) motion approach (see Section 8.4.3). Sequence `'Soccer'` (already seen in Section 8.4.5) is less complex than sequence `'Football'` (already seen in Section 8.4.5) since the latter suffers from motion blur. For the piecewise motion estimation, each frame was divided into 16×16 blocks B_i of size 22×18 pixels. As a reminder, domain Ω_i is defined as $\Omega \cap B_i$. The comparison between the two approaches is presented in Figs. 8.16 and 8.17. Although the local approach clearly improves the segmentation, it is not perfect in sequence `'Football'`. This can be explained by the combined effects of motion

[13] To be coherent with the *piecewiseness* property of entropy, a mixture of Gaussians would be more appropriate. However, the purpose of this section is to compare the proposed approach with classical error terms such as the Sum of Squared Differences (SSD).

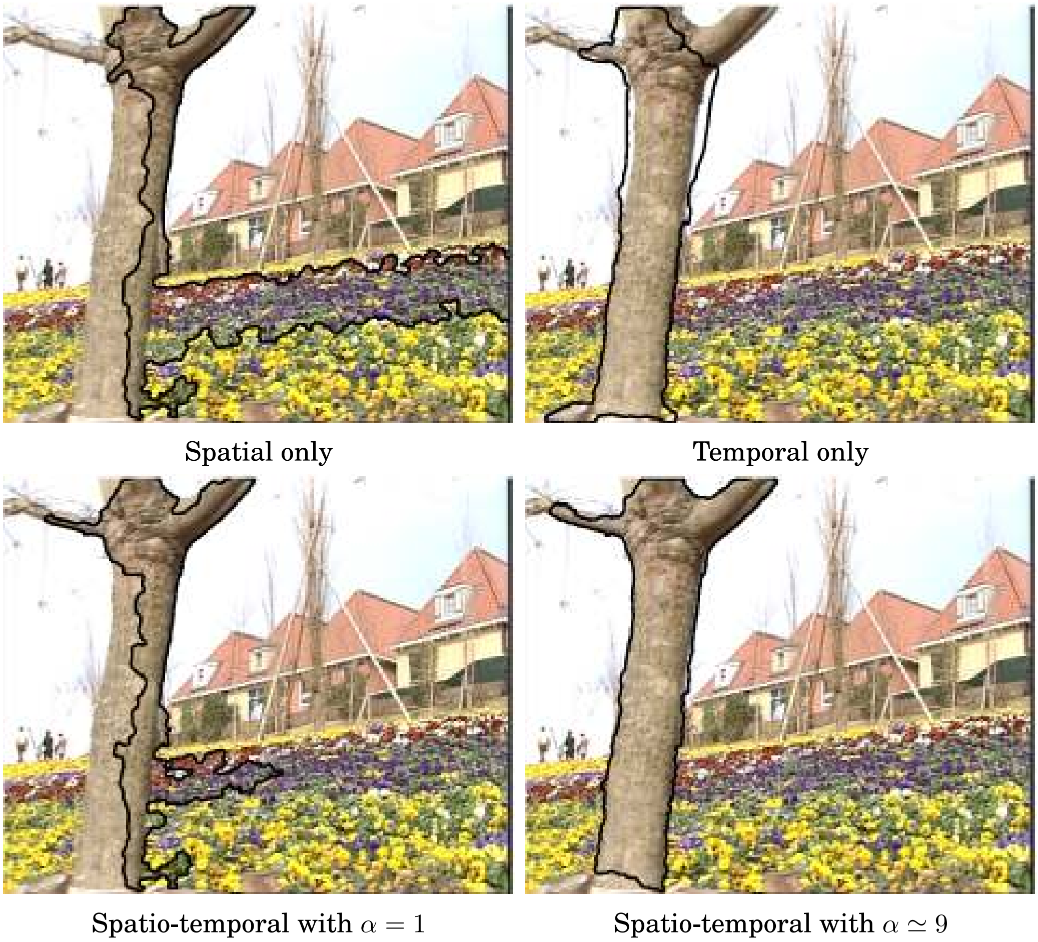

Spatial only
Temporal only
Spatio-temporal with $\alpha = 1$
Spatio-temporal with $\alpha \simeq 9$

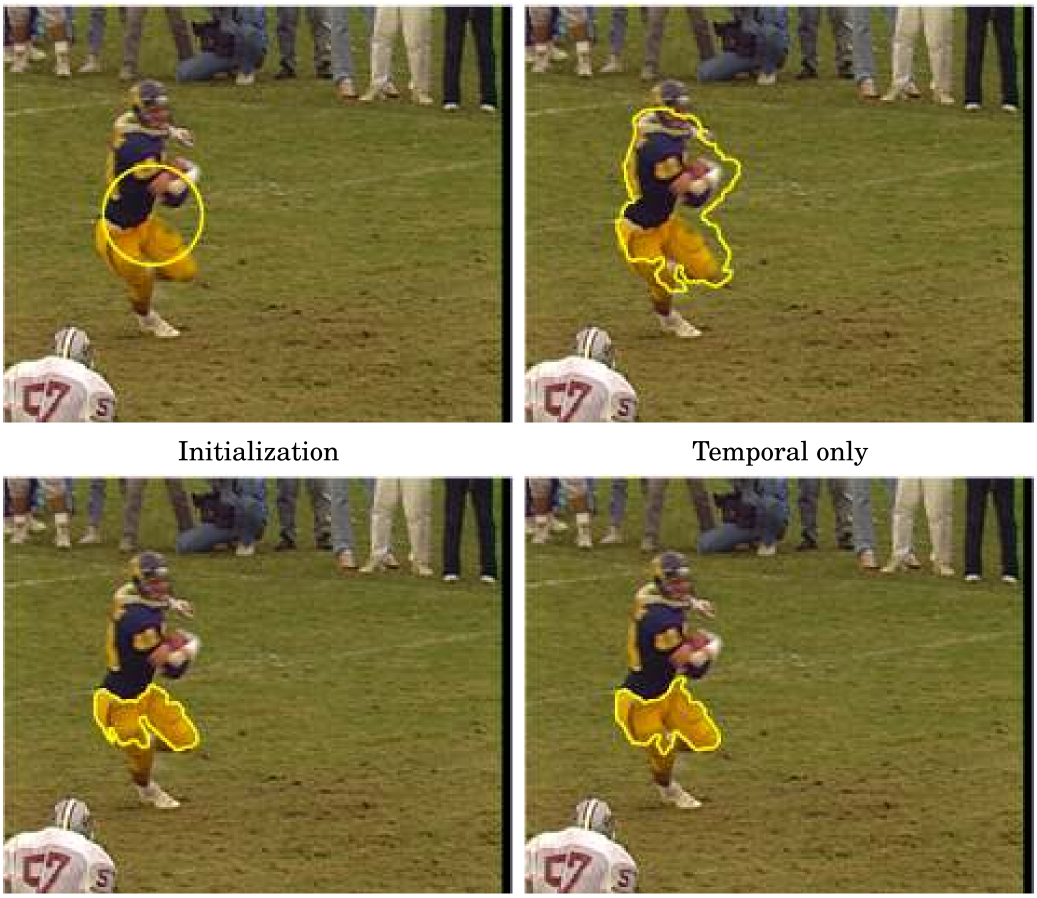
Figure 8.12: *Segmentation of frame 237 of sequence* `'Flowers and garden'` *assuming parametric distributions and using the same initialization as in Fig. 8.10. (Lower left) The spatio-temporal energy relies equally on space and time and (Lower right) the spatio-temporal energy favors the temporal term.*

blur and a domain Ω_i too small (which may happen for blocks B_i that intersect Γ), resulting in a less reliable local motion estimate.

8.4.6 Conclusion

The addressed problem was the segmentation of a video sequence. A spatio-temporal approach was chosen in order to make use of both spatial and temporal coherence. As opposed to the classical approach consisting in dealing with time by involving the MCE directly, the proposed method is based on the use of the distribution of the MCE. This allowed to combine temporal and spatial information coherently using joint distributions. The distributions were estimated nonparametrically to fit the data. Entropy was chosen as the energy to minimize, in particular because, in practice, it is robust to outliers. In order to make the motion model complex enough to describe articulated objects, it was proposed to keep it simple (namely, translation) while estimating its parameters locally.

The proposed method was qualitatively compared with a classical, parametric approach followed by some existing methods. Thorough comparison with specific methods is out of



Initialization

Temporal only

Spatial only

Spatio-temporal

Figure 8.13: *Segmentation of frame 72 of sequence* `'Football'` *assuming parametric distributions. The spatio-temporal energy relies equally on space and time ($\alpha = 1$).*

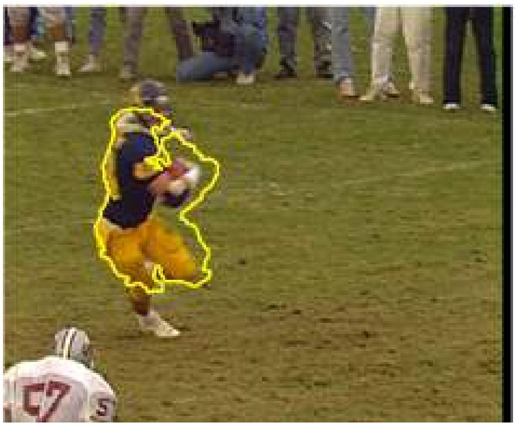
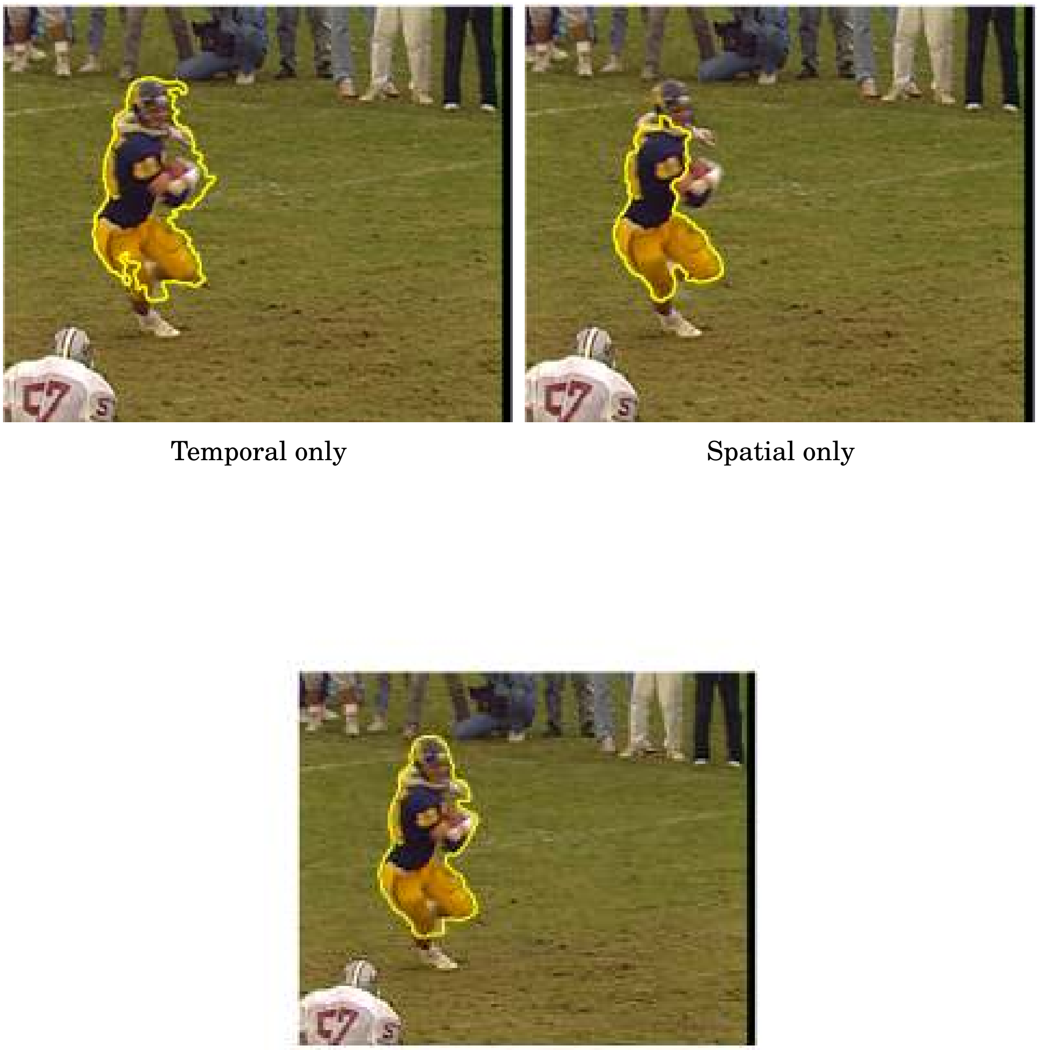


Figure 8.14: *Segmentation of frame 72 of sequence* `'Football'` *assuming parametric distributions. The spatio-temporal energy favors the temporal term ($\alpha \simeq 9$).*

scope of this section. Nevertheless, on sequence `'Flowers and garden'`, our results (see Fig. 8.10) are comparable to those of recent segmentation methods [CS03], Fig. 4 in both articles.

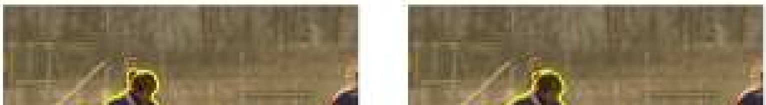


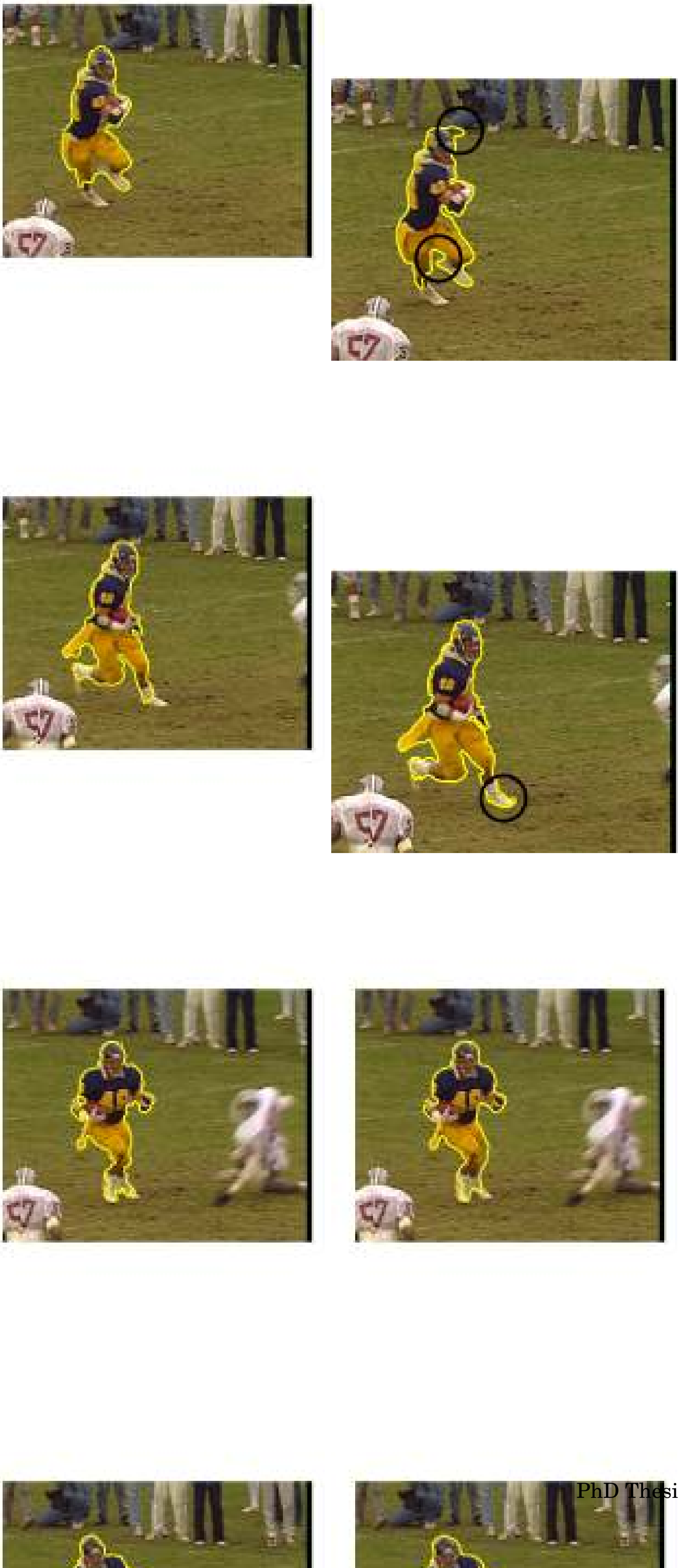
Temporal only

Spatial only

Spatio-temporal

Figure 8.15: *Segmentation of frame 72 of sequence* `'Football'` *with the nonparametric approach using the same initialization as in Fig. 8.13.*





Part IV

Conclusion and Perspectives

CHAPTER **9**

CONCLUSION

The main contribution of this thesis is a new general framework to deal with high dimensions in variational problems. Through three classical image processing problems, kernel tracking, optical flow and segmentation, various mechanisms involved in variational problems solutions were exhibited. In these problems, high dimensionality of the data is often pointed as a major direction of improvement but lacked an efficient nonparametric framework to use it in practice.

Indeed, a framework for variational problem able to: (1) efficiently estimate statistical energies (all variational problems basically require to compute an energy); (2) approximate derivatives of these energies in order to be used in derivative-based solution (in particular for optical flow estimation); (3) approximate locally a probability connected to these energies (in particular for image segmentation); on high dimensional data, was never, to the best of our knowledge, exhibited and is the major contribution of this thesis.

We have proposed a unified framework relying on kNN distances, which satisfies all the above conditions:

1. the class of kNN estimates of entropy-based measures is computed directly on the data shortcutting PDF estimation,
2. the kNN mean-shift is able to approximate derivatives of kNN estimates of entropy-based measures,
3. the kNN estimate of PDF is the underlying PDF construction behind these kNN estimates.

Finally, complexity is isolated in the k-nearest neighbor search, which can be solved very efficiently using k-d trees or GPU-based implementations [GDB08].

Novelty of this framework relies on simple kNN forms of various entropy-based measures derived from locally adaptive PDF estimates. The unification of kNN entropy-based measures, kNN mean-shift methods and kNN locally adaptive PDF estimates, is also novel, and aims at solving efficiently variational problems.

Mean-shift was already applied in image processing [CRM03, Com03] but used as a gradient ascent algorithm rather than a derivative of entropy-based statistical measures. Locally adaptive PDF estimates were also applied in image processing [Com03, TPJ05] but as a clustering algorithm rather than a PDF estimate connected to statistical measures.

The problem of dimensionality being, to a certain extent, circumvented, we explored high dimensional spaces and exhibited how they can advantageously capture natural image statistics.

We revisited the three initial variational problems using the kNN framework, tracking, optical flow and segmentation as well as contributions which take advantage of high dimensions. In particular, these new dimensions allow to introduce smooth constraints for image matching, smooth regularization for optical flow and define local probabilities or shape priors for segmentation. In general, the kNN framework allows to define information-theoretic energies on multivariate data in variational problems.

We obtained competitive results in the three variational problems in terms of accuracy and quality. Yet, this thesis had two industrial applications. The tracking algorithm presented in Chapter 6 with GPU implementation of the kNN search is being implemented by a cinema post-production company (Mikros Image). The simplified version of the motion segmentation algorithm is implemented in a H264 video coder by the French national telecom operator (Orange).

CHAPTER **10**

PERSPECTIVES

10.1 Exploration of high dimensional spaces

The problem of dimensionality is circumvented, to a certain extent, in the kNN framework. Thus, exploration of high dimensional spaces for variational problems can be pushed further.

10.1.1 Space of Neighborhood patches

One possible study is the integration of neighborhood patches in this framework, following the idea of recent denoising algorithms [BCM05, AW06] and our illustrations on Fig. 5.2. In particular, in the regularization term for optical flow (7.13) we can simply write $u_{\mathrm{m}} = V(\mathrm{m}) = \{\mathrm{v}(\mathrm{n}) \mid \mathrm{n} \in \mathcal{N}(\mathrm{m})\}$, where $\mathcal{N}(\mathrm{m})$ is a spatial neighborhood of m (typically a 5×5 patch, in this case V is a vector field in a 25 dimensional space). This perspective has already led to some applications in image retrieval, [ADPB08, PADB08].

10.1.2 Medical imaging

Medical imaging offers by nature multiple features derived directly from the acquisition process (DTI,QBI), from simultaneous or multiple acquisitions (*e.g.*, PET/CT scans, intra patient/inter patient studies), or from priors (acquisition vs. atlas). Variational formulations of medical imaging through entropy-based measures, for instance image registration through mutual information, could take advantage of these high dimensional spaces.

10.1.3 Regularization

Entropy of an image luminance, as a regularization term, still suffers from cartoon effect as classical total variations. One could try entropy of the gradient instead of entropy on the data. Indeed, the gradient of a piecewise smooth function has a low entropy. This type of regularizer would not favor piecewise constant but piecewise smooth functions.

10.1.4 Shape priors

Shape priors can be define with geometric features based on shape coordinates as defined in the section on shape priors in segmentation 8.3. We will explore other priors by using a more efficient shape coordinates system based on contour correspondence such as shape contexts [BMP02], or other shape coordinates such as free form deformations [HPM06] or conformal mapping [SM06].

10.1.5 Manifolds

High dimensional spaces on several data in image processing (neighborhood patches [HM99], optical flow [RB05], tensors [AFPA06], shapes [ESK07]) have exhibited specific manifolds. We will also explore the manifold structure of such high-dimensional data and explore PDF estimation that accounts explicitly for underlying manifolds. Indeed, the Euclidean metric used in the kNN framework may reach here its limitations in these cases and adapted metrics should be derived to introduce anisotropy.

10.2 Dimension reduction techniques

As the number of dimensions increases, a strategy is to employ dimension reduction techniques. These algorithms project high dimensional spaces to lower dimensional space. The basic idea is that correlated dimensions are grouped together. This idea is closely related to PDF factorization which allows to group correlated dimensions together. To deal with very high dimensional spaces, a dimension reduction technique could be applied as a preprocessing step. The metric derived from the projection algorithm can then be used as a distance to search for the k-nearest neighbor in the remaining dimensions.

10.3 A unification between variational methods and information theory: coding applications

The kNN framework allows to minimize energies issued from information theory. These measures have a physical sense as a amount of information for coding. Minimizing this amount of information, one can find optimal solutions of motion estimation (optical flow) or partitions of videos (segmentation) in a coding sense. This motion or partition will be optimal in terms of coding rate. A possible application would be region-based video coding, where the region of interest would have an high coding rate.

Indeed, variational methods and video compression algorithms are often used together but applied sequentially. The variational problem gives an optimal solution in the sense of a criterion (for instance motion estimation uses block matching with SAD criterion) and the video compression algorithm compresses the data in the sense of another criterion (quantization of the motion field uses minimum entropy criterion). There is no global optimization between the coding rate and the quality of the result of the variational problems, we believe that a joint optimization would be of great interest.

Moreover, video compression is a chain dealing with multiple features (motion, image wavelets sub bands), at the end of the chain, these high dimensional features go through an entropic coder. Minimizing the joint entropy of all these features before sending them to the entropic coder can decrease the number of the bits at the output of the coder.

A simplified motion segmentation problems proves that it is realistic to use variational solutions for video coding purpose, the kNN framework proved that information theory criteria, commonly use for coding, can not only be used, but enhanced the solution of variational problems. Joint methods are possible.

10.4 Other measures

Other types of measures born from information theory can also explored such as Bregman divergences [BMDG05, NBN07] and adaptive windows for PDF estimation, such as Voronoi diagrams.

10.5 Efficient k-th nearest neighbor search using GPU Implementation

In this thesis, complexity has been isolated in the kNN search. Approximate nearest neighbor search (ANN) rather than nearest neighbor search speeds up the computations. One can explore how bounded errors on neighbors neighbor search will imply a bound error on entropy. Another strategy is highly parallelized nearest neighbor search on GPU [GDB08] using the new NVIDIA library CUDA. The tracking algorithm presented in this thesis has already been implemented using GPU for industrial applications. Other algorithms defined in these thesis can also be easily implemented on GPU using this library.

Part V

Appendix

APPENDIX **A**

kNN FRAMEWORK: PERFORMANCE ON GAUSSIAN MIXTURES

In this Appendix, we compare the estimates of the “kNN framework” to their classical alternatives. The comparisons include PDF estimates, statistical measure estimates and derivative estimates of these measures.

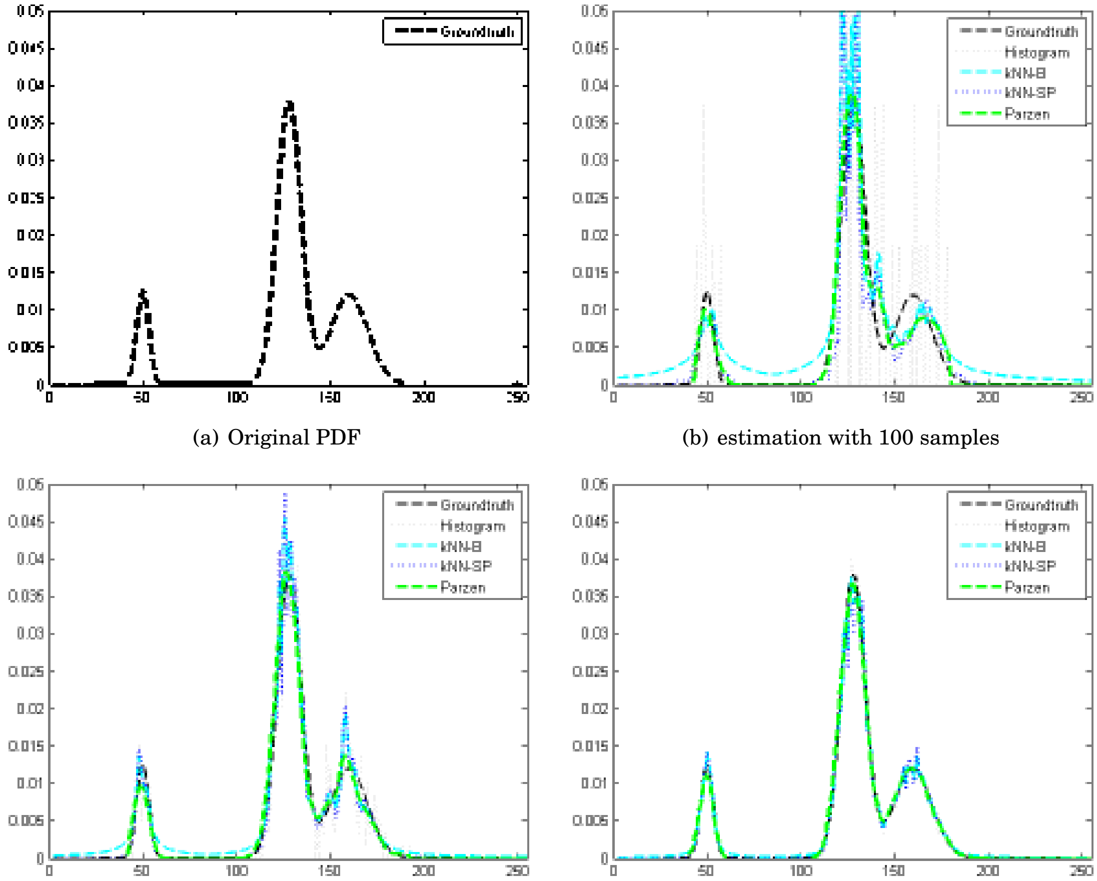
In the following experiments, “Histogram” is the histogram density estimate, “Parzen” denotes the fixed-size kernel density estimate with gaussian kernel and a bandwidth estimated with rule of thumb, “Parzen ind” means components were assumed independant. “kNN-B” is the balloon estimate with a uniform kernel and a bandwidth estimated with kNN, “kNN-SP” is the sample point estimate with a uniform kernel and a bandwidth estimated with kNN. When these names are used for an entropy estimate, it means that the mentioned kernel density estimate was plugged in an Ahmad-Lin estimate of entropy.

A.1 PDF estimation

Let us define a 1-D gaussian mixture.

$$f_U(u) = \sum_{i=1}^{3} w_i f_{\mu_i,\sigma_i}(u), u \in \mathbb{R} \tag{A.1}$$

The mixture is composed of three Gaussians of means 128, resp. 160, resp. 150, of standard deviation 40, 100, 10 and of weightings 0.6, 0.3, 0.1. Given a varying number of samples from this distribution we visually compare the density estimates of various estimates: Histogram, Parzen, Balloon, Sample point Fig. A.1. This is the ideal case for Parzen estimate which is performing well. Balloon estimate is less accurate and has an high order bias in the tails. It is clearly non competitive in 1-D dimensions on a classical distribution.



(a) Original PDF

(b) estimation with 100 samples

(c) estimation with 1,000 samples

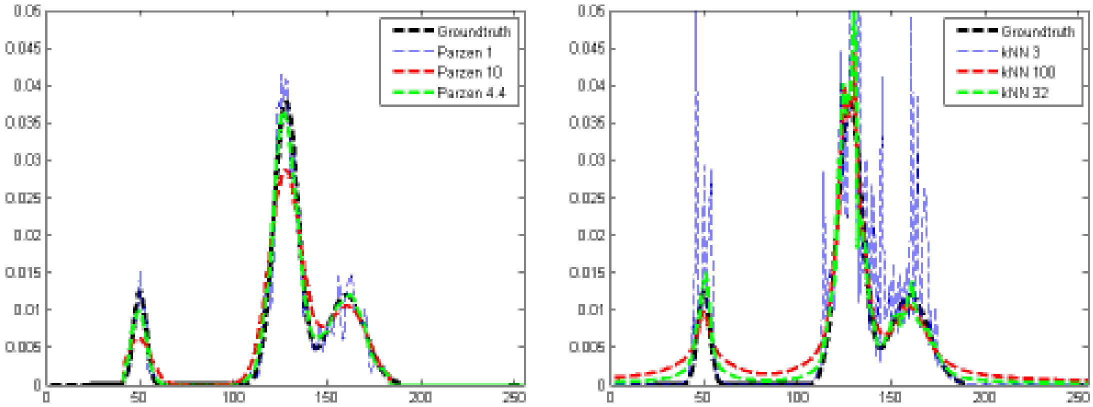
(d) estimation with 10,000 samples

Figure A.1: *PDF estimation performance on a 1-D Gaussian mixture for different sample sizes, from left to right: groundtruth, PDF generated from 100 samples, 1,000 samples, 10,000 samples*

Robustness to bandwidth parameter is now studied on Fig. A.2. The only parameter in Parzen estimate is the standard deviation, called bandwidth of the kernel whereas in the balloon estimate the only parameter is k, to compute the k-th nearest neighbor. In Parzen, the bandwidth h computed with plugin rule is 4.4 on this example, we also plotted the PDF for $h = 1$ and $h = 10$. In kNN balloon, the parameter k is choosen to be $\sqrt{1000} = 32$, we also plotted the PDF for $k = 3$ and $k = 100$. Small values of k are sensitive to noise.

A.2 Entropy estimation

As analytic formula of the entropy of a Gaussian mixture does not exist, instead a highly sampled quadrature numerical integration is performed on the PDF in order to find a good approximation of true entropy. We run 100 generations on five different sample of sizes $10^i, i = 2..5$ and plot the mean of entropy obtained. This experience is done on one dimensional and three dimensional mixtures. The correlation matrices of three dimensional Gaussians are generated though three random correlation matrices as follow: we generate three 3×3 random matrices X, each component being between 0.1 et 1 following a uniform law, and we compute a correlation matrix $\Sigma = XX^T$. We also multiply the diagonal components by 3 to enhance



(a) Parzen (b) kNN

Figure A.2: *PDF robustness on a 1-D Gaussian mixtures for different values of parameters (means over 100 generations for each sample size)*

the condition number of the matrix. We compare estimates of entropy on Fig. A.3 on a 1-D experiment and 3-D experiment.

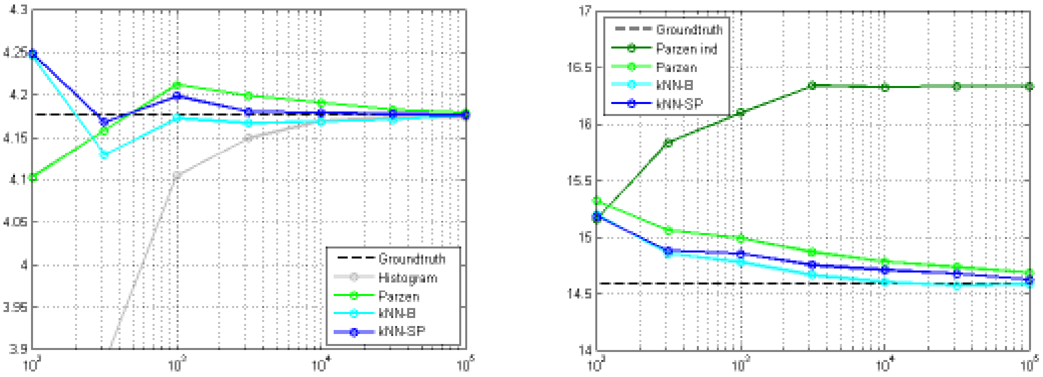


Figure A.3: *Entropy estimation performance on a 1-D Gaussian mixtures for different sample size (means over 100 generations for each sample size, Entropy estimation performance on a 3-D Gaussian mixtures for different sample size (means over 100 generations for each sample size)*

Robustness to parameter is tested on Tab. A.1, parameter k in both balloon and sample point estimates appears to be less critical than Parzen. Indeed, k is not a fixed value but a range of smoothing which adapts to the data.

Even with a bad estimation of the PDF Fig. A.2, kNN provides a good entropy estimate

A.3 Kullback-Leibler divergence estimation

We compute the Kullback-Leibler divergence between two distributions Fig. A.4. Various number of samples of two Gaussian mixtures with 3 overlapping and non overlapping modes are generated. As there is again no analytical solution to compute the exact Kullback-Leibler divergence between two Gaussian mixtures, we compute a highly sampled quadrature inte-

real entropy: **4.1763**	”Undersmooth”	”Correct”	”Oversmooth”
Entropy estimated with Parzen	4.1512	4.2095	4.3180
Entropy estimated with kNN	4.1710	4.1732	4.1778
Entropy estimated with Sample-Point	4.1720	4.1754	4.1771

Table A.1: *Entropy estimation performance on a 1-D Gaussian mixtures, 1,000 samples, ”Undersmooth” bandwidth h=1 for Parzen, k=3 for kNN, ”Correct” bandwidth: h=plugin par Parzen , h=32 for kNN, Oversmooth bandwidth h=10 for Parzen, k=100 for kNN*

gration over the PDF. We compare estimates of Kullback-Leibler divergence based on three different PDF estimates: the Parzen estimate, the balloon estimate and the sample point estimate. The same test is then performed on 3-D Gaussian mixtures selected randomly as in the entropy estimate experiments. We compare three Kullback-Leibler divergence estimates for various sample sizes.

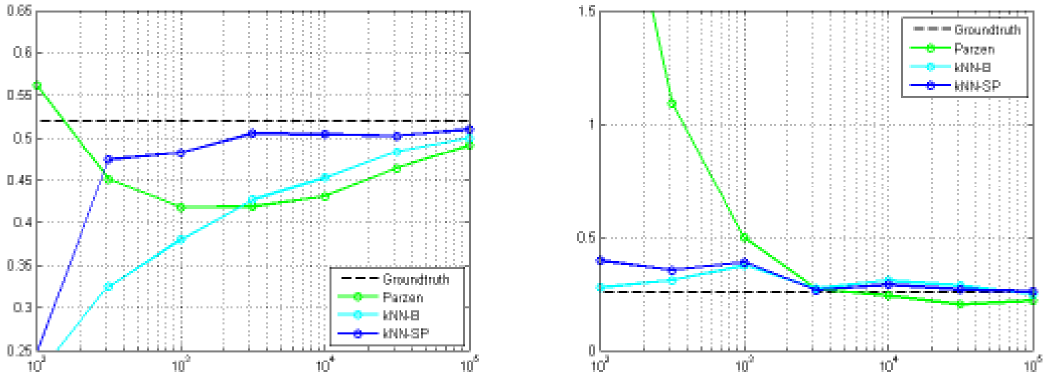


Figure A.4: *Kullback estimation performance on a 1-D Gaussian mixtures (left) and 3-D Gaussian mixtures (right) for different sample size (means over 100 generations for each sample size)*

In order to test parameter estimation, let us suppose we must match two Gaussian PDFs Fig. A.5. The first example is two 1-D Gaussians with slightly different means. The variance of the first Gaussian is known and we search for the optimal variance for the second Gaussian. The second example is two 5-D correlated Gaussians with (slightly different variances). We are searching for the mean of the sample minimizing the Kullback-Leibler divergence.

A.4 Mean-shift estimation

Analytical expression of mean shift $\nabla f/f$ of a Gaussian function is a linear function. We compare three diferent mean-shift estimates described in this chapter with the analytical formula Fig.A.6. Far outside of the support of the samples, both Parzen and sample-point mean-shift estimates are equal to 0. Indeed, there are no samples in the Parzen or sample-point mean shift estimates. Sample-point mean-shift is non-zero further than Parzen because it uses large bandwidth in the tails. However kNN window always contains at least k samples, it is thus never equal to zero.

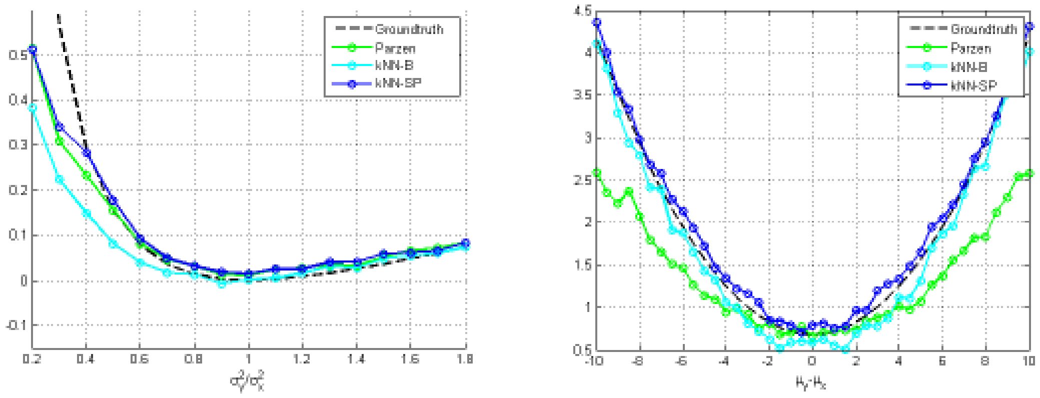


Figure A.5: *Kullback minimization performance between two Gaussians: from left to right, 1-D Gaussian with varying variance, 5-D correlated Gaussian with varying means*

A Gaussian mixture generates 1000 samples, an initialization point is choosen, Mean-shift converges to the closest mode Fig A.7. Mean-Shift with Fixed-size bandwidth (Parzen) converges slower than variable-size bandwidth balloon Mean-Shift.

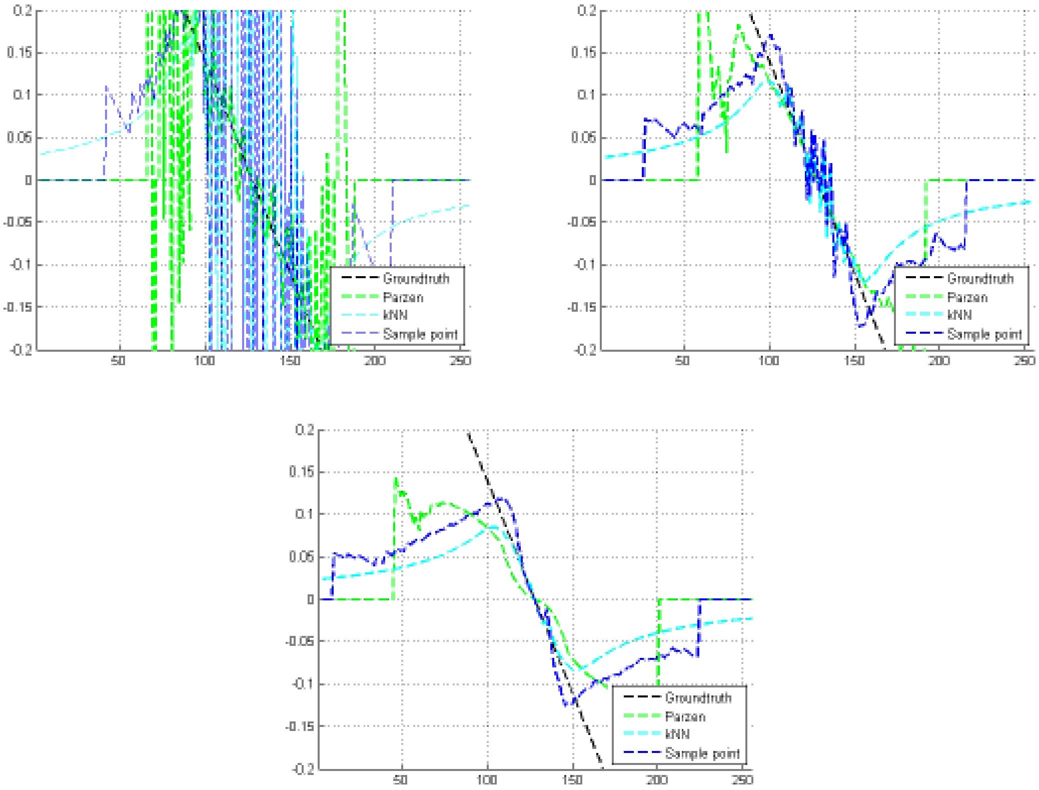


Figure A.6: *Mean shift estimation of a Gaussian, left undersmooth, right optimal, bottom oversmooth*

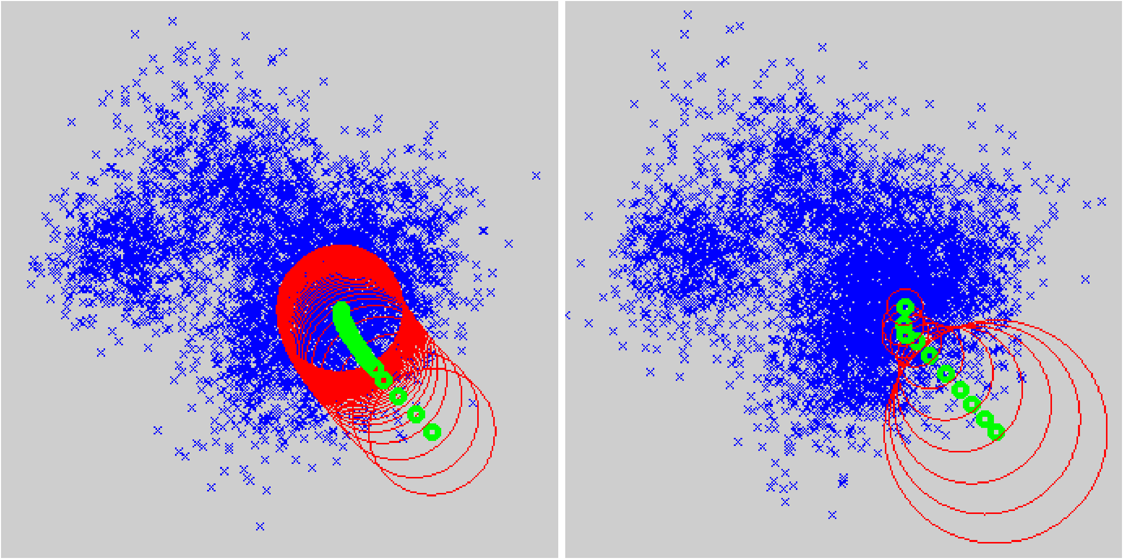


Figure A.7: *Mean shift illustration, fixed size bandwidth (left), variable bandwidth (right)*

APPENDIX **B**

ROBUSTNESS OF ENTROPY TO NOISE AND DEGRADATIONS

In this appendix, we compare an entropy-based similarity measures with state of the art strict geometric methods (from classical sum of square difference (SSD) to more robust estimators as sum of absolute differences (SAD). Experimental studies focus on robustness to various noise and natural motion estimation difficulties (occlusions, variations of illuminance, . . .)

Visual comparisons

We compare an entropy-based similarity measures with state of the art strict geometric methods (from classical sum of square difference (SSD) to more robust estimators as sum of absolute differences (SAD). A complete experimental study on robustness to various noise and natural matching difficulties (occlusions, variations of illuminance, . . .) can be found in Appendix B.

We show some visual results, we plot the criteria value in a 12×12 translation window around the true position of a reference template in sequence “Edberg”. Both SAD and SSD criteria are non convex around the solution while entropy based criterion Pz-H and kNN-H are convex around the solution.

Natural difficulties

As the performances of SAD or SSD drops when the data are not normally distributed we expect a significant gain when the residual error is assumed not gaussian (in the presence of occlusions or high texture for instance). In this section, we compare SSD and SAD measures with an entropy similarity measure based on parzen Pz-H (4.15) and with an entropy similarity measure based on kNN kNN-H (5.11) . We present results on real standard sequences that present typical difficulties for image matching

Occlusions

We first consider the *Erik* sequence, which consists in a head moving on a static background. We test the methods on a close-up on blocks around the left edge of the face as shown in

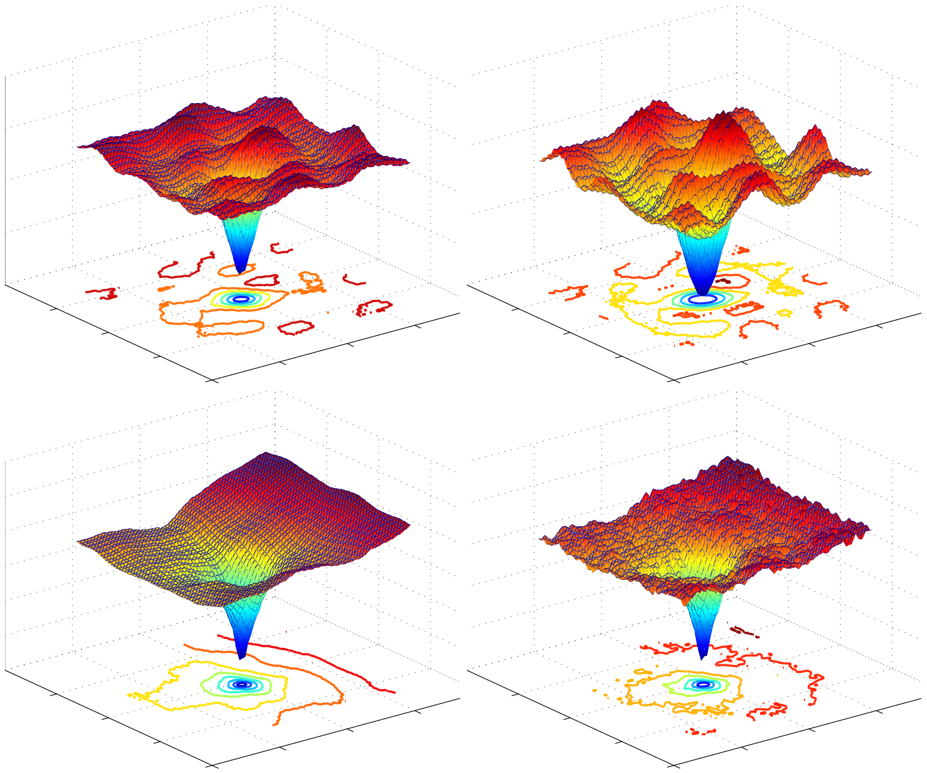


Figure B.1: *Minimization for two φ functions fails, entropy succeeds SAD, SSD, Pz-H, kNN-H*

Figure 2. Between I_n and I_{n+1}, the face (foreground) occludes the curtain (background). Here SAD does not find the correct match for the block from I_n (the mid-left block in Fig. 2, left) and tries to match it further downwards. kNN-H performs well since it finds no motion in the background (there are no motion vectors for the left blocks in Fig. 2, right).

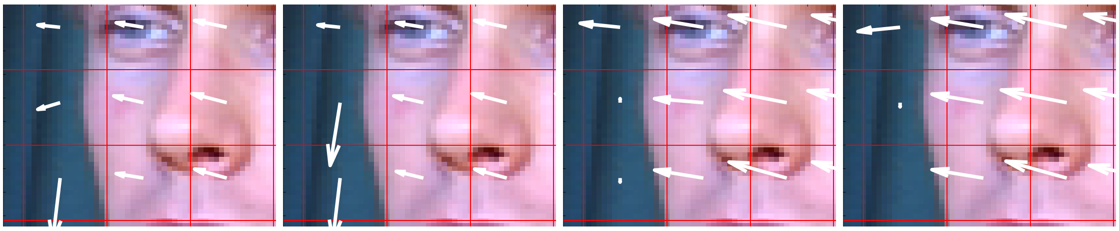


Figure B.2: *Occlusions: close-up of* Erik *processed with, from left to right SSD, SAD, Pz-H and kNN-H*

Highly textured images

Here we focus on the highly-textured public in the background of the standard *Edberg* sequence. Motion estimation on a highly textured image is very sensitive to even slight errors in the motion vector (which happen naturally for instance for non-uniform translations or motion vectors quantizations) and to suboptimal motion search, as the common criteria are non-convex. Figure 3 illustrates how SAD fails to find a uniform motion in the public (left) whereas kNN-H gives satisfaction (right).

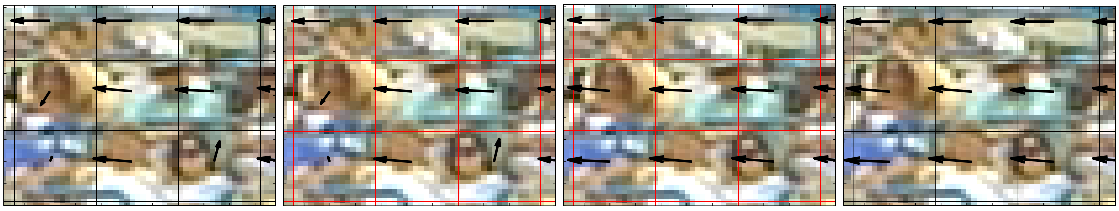


Figure B.3: High texture: from left to right SSD, SAD, Pz-H and kNN-H applied to *Edberg*; close-up of the public

Other significant improvements can be observed for other kinds of difficulties, such as motion blur, variations of illuminance, or different motions in a same block. These results are avaliable upon request.

Some extreme robustness experiments

Let us now consider a synthetic color sequence, for which the ground truth is known, in order to perform exact error measurements. The YUV color sequence considered here consists in the superposition of two images: an horizontal translation of a face (Foreman's) towards the left handside and a background (a view of Boston) travelling also horizontally but towards the right handside. There is no other movement in the scene; for instance the face does not move. The main difficulty with this sequence resides in frequent occurrences of occlusions (at the edges of Foreman's face or on the borders of the image) and in the texture of the object and the background. As the sequence considered here is a pure travelling, there are not enough occlusion problems to illustrate the robustness property of the entropy estimators. We thus consider the same sequence, that we name *Original*, but randomly altered as shown in Figure B.4 by the addition of

- *Flash*: variations of brightness between two successive frames (like a camera flash) of 40 units of luminance;
- *Noise*: "salt and pepper" noise covering 10% of a frame;
- *Patch*: black patches (similarly to scratches on a film);
- *Altered*: the 3 previous noises all at the same time, but with 5 units of variation of luminance instead of 40.

The results in terms of mean and standard deviation of the angular error are shown in Table 2. In addition to applying the SSD, Pz-H and kNN-H criteria to 3D data, we also consider their 1D alternative applied to the luminance (Y) channel of the image, as well as the SAD criterion, as these may turn out to be reasonable alternative estimation procedures in practice. Overall, the (3D) kNN-H method outperforms all other procedures and it seems more robust to the perturbations that were introduced. The Pz-H estimator fails hard, which only highlights that selection of the kernel bandwidths is a critical problem in the presence of perturbations (it is even more important with the 3D version). The kNN-H estimator does not suffer from these aspects. 1D kNN-H may provide a reasonable alternative since it is computationally less demanding and provides good results.

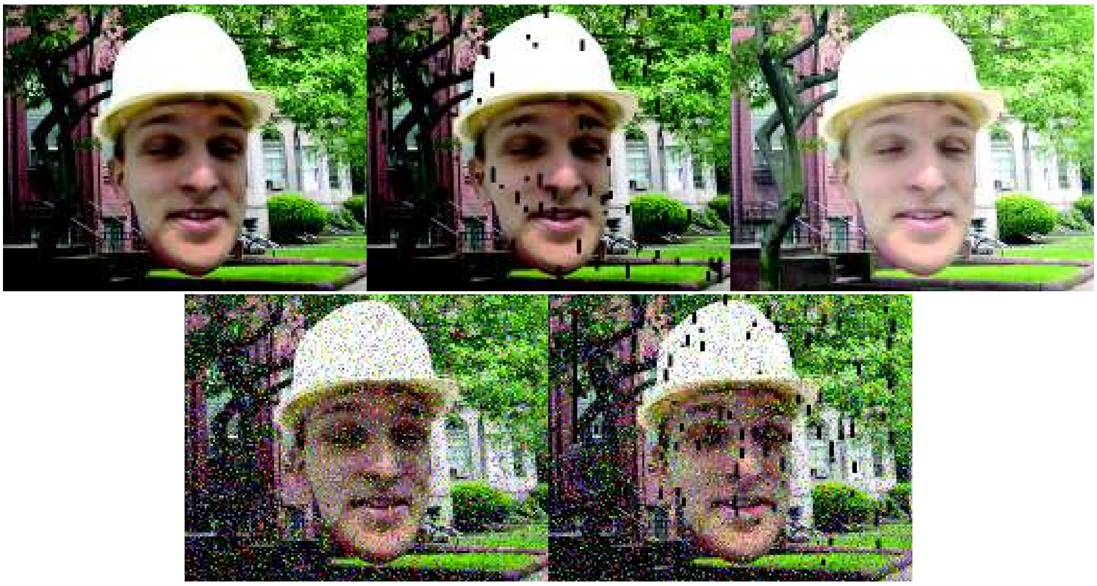


Figure B.4: The ”Foreman-Boston” sequence and altered with (from left to right, top to bottom), flash, noise and all three together

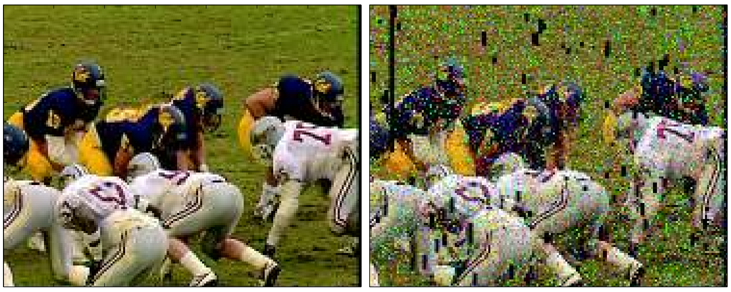


Figure B.5: *Sequence ”Football” and its altered version*

Sequence		*Original*	*Flash*	*Snow*	*Patches*	*All*
SAD	$\bar{\omega}$	8.9	57.1	14.4	13.7	26.9
	σ_ω	28.2	42.9	39.0	33.0	33.7
SSD	$\bar{\omega}$	9.5	45.1	33.3	22.4	44.9
	σ_ω	27.7	47.7	36.4	38.7	41.1
Pz-H ($\hat{h}$)	$\bar{\omega}$	6.7	7.0	39.8	12.5	44.8
	σ_ω	22.1	21.8	40.0	26.8	42.2
Pz-H ($h = 3$)	$\bar{\omega}$	5.83	6.23	7.98	6.77	10.30
	σ_ω	22.24	22.12	27.46	24.13	29.68
kNN-H	$\bar{\omega}$	**4.4**	**5.0**	**5.6**	**4.5**	**9.7**
	σ_ω	**21.0**	**21.0**	**22.2**	**21.1**	**28.5**
SAD (1D)	$\bar{\omega}$	9.1	67.2	15.8	16.91	28.5
	σ_ω	27.8	41.6	33.5	35.3	40.5
SSD (1D)	$\bar{\omega}$	10.6	48.7	31.3	22.5	43.6
	σ_ω	29.4	47.3	34.0	38.5	41.7
Pz-H ($\hat{h}$**, 1D)**	$\bar{\omega}$	9.1	12.8	27.4	17.7	35.1
	σ_ω	27.1	30.3	34.6	36.5	40.6
Pz-H ($h = 3$**, 1D)**	$\bar{\omega}$	8.10	11.88	11.07	9.65	11.53
	σ_ω	26.28	31.54	30.60	28.39	30.79
kNN-H (1D)	$\bar{\omega}$	8.8	11.5	9.4	11.0	11.6
	σ_ω	27.7	31.3	27.7	30.9	32.5

Table B.1: Motion flow error on Foreman-Boston altered with several methods: comparison with ground truth. Mean μ and standard deviation σ^2 of angular error are given in degrees.

Criterion	Mean $\bar{\omega}$	Std-dev σ_ω
SAD (1D)	11.15	23.01
SAD	11.28	**22.91**
SSD (1D)	29.2	28.9
SSD	31.4	30.4
Pz-H ($\hat{h}$**, 1D)**	23.8	24.0
Pz-H ($\hat{h}$)	35.3	32.8
Pz-H ($h = 3$**,1D)**	**8.53**	**19.46**
Pz-H ($h = 3$)	**10.30**	**21.32**
kNN-H (1D)	**8.2**	**22.0**
kNN-H	**7.7**	**21.0**

Table B.2: Motion flow error on altered Football: comparison with the unaltered sequence

Experiment on a real sequence: Football

The different motion estimators considered are now applied to a real sequence, "Football". As there is no ground truth on this sequence and as the distortion error of the DFD cannot be lower than the value provided by SSD, we have to define another test for robustness. First, compute the motion on the original video with several criteria; then alter the video with a “salt and pepper noise” with 10% coverage and random black patches, and finally compute the motion on this altered sequence with the same criteria and compare them to the original results. Under these heavy perturbations, only kNN-H in both its 1D and 3D versions remain close enough to the original estimate, while the others criteria all fail, as shown in Table 3.

APPENDIX **C**

GRADIENT OF THE KULLBACK-LEIBLER DIVERGENCE: MEAN-SHIFT-BASED EXPRESSION AND KNN IMPLEMENTATION

C.1 Preliminary note

In the context of tracking, Mean-Shift is often used to refer to a Mean-Shift-based algorithm. Here, it refers to the original meaning [FH75] of approximation of $\nabla f / f$ using the shift from the mean of neighboring samples

$$\frac{\nabla f(s)}{f(s)} \simeq \frac{d+2}{h^2}(\mu_F(s) - s) \tag{C.1}$$

where

$$\mu_F(s) = \frac{1}{n} \sum_{t \in \mathcal{N}_F(s)} t \tag{C.2}$$

is the mean of the samples (which happens to be n in number) contained in a window $\mathcal{N}_F$ of radius h centered at s. If f is a normal distribution with mean μ and variance h^2, then the Mean-Shift has the following, simple analytical expression

$$\frac{\nabla f(s)}{f(s)} = \frac{\mu - s}{h^2}. \tag{C.3}$$

C.1.1 Derivative

The Kullback-Leibler divergence is equal to

$$\mathfrak{D}_{\text{KL}}(T_\varphi, R) = H^\times(T_\varphi, R) - H(T_\varphi) \tag{C.4}$$

where the cross-entropy $H^{\times}(T_\varphi, R)$ is equal to

$$\begin{aligned} H^{\times}(T_\varphi, R) &= -\int_{\mathbb{R}^d} f_{T_\varphi}(s) \ \log f_R(s) \ \mathrm{d}s && \text{(C.5)} \\ &= \mathrm{E}_{T_\varphi}[-\log f_R] && \text{(C.6)} \\ &\simeq -\frac{1}{|T_\varphi|} \sum_{s \in T_\varphi} \log f_R(s), && \text{(C.7)} \end{aligned}$$

and the differential entropy $H(T_\varphi)$ can be approximated by the Ahmad-Lin estimator [AL76]

$$\hat{H}_{\mathrm{AL}}(T_\varphi) = -\frac{1}{|T_\varphi|} \sum_{s \in T_\varphi} \log f_{T_\varphi}(s). \qquad \text{(C.8)}$$

In (C.8), the PDF is by definition equal to

$$f_{T_\varphi}(s) = \frac{1}{|T_\varphi|} \sum_{t \in T_\varphi} K_h(s-t). \qquad \text{(C.9)}$$

The same estimation (replacing T_φ with R) will be used in (C.7).

Therefore, we have

$$\begin{aligned} \mathcal{E}(\varphi) &= -|T_\varphi| \, \mathfrak{D}_{\mathrm{KL}}(T_\varphi, R) && \text{(C.10)} \\ &\simeq \sum_{s \in T_\varphi} \log f_R(s) - \log f_{T_\varphi}(s). && \text{(C.11)} \end{aligned}$$

Note that $\sum_{s \in T_\varphi} a(s)$ is a convenient notation to designate $\sum_{n=1}^{|T_\varphi|} a(T_\varphi(n))$ where $T_\varphi(n)$ is the n^{th} sample in T_φ. Moreover, $|T_\varphi|$ is constant for all candidate regions in a given frame. Consequently, taking the derivative of (C.11) with respect to φ does not require to care about the interval of summation. Let the transformation φ be a translation (u, v) combined with a scaling by α. Sample set T_φ is equal to

$$T_\varphi = \{(I_{\mathrm{tgt}}(x+u, y+v), x/\alpha, y/\alpha), \ (x, y) \in \Omega\}. \qquad \text{(C.12)}$$

The derivative of (C.11) with respect to $\varphi = (\alpha, u, v)$ is equal to

$$\begin{aligned} \nabla \mathcal{E}(\varphi) &= \sum_{s \in T_\varphi} \frac{1}{f_R(s)} \frac{1}{|R|} \sum_{t \in R} \frac{\partial}{\partial \varphi} K_h(s-t) \\ &\quad - \frac{1}{f_{T_\varphi}(s)} \frac{1}{|T_\varphi|} \sum_{t \in T_\varphi} \frac{\partial}{\partial \varphi} K_h(s-t) && \text{(C.13)} \\ &= \sum_{s \in T_\varphi} \frac{1}{f_R(s)} \frac{1}{|R|} \sum_{t \in R} \mathcal{D}_s(T_\varphi) \ \nabla K_h(s-t) \\ &\quad - \frac{1}{f_{T_\varphi}(s)} \frac{1}{|T_\varphi|} \sum_{t \in T_\varphi} \frac{\partial}{\partial \varphi} K_h(s-t) && \text{(C.14)} \end{aligned}$$

where

$$\begin{aligned} \mathcal{D}_s(T_\varphi) &= \left[\begin{matrix} 0 & 0 \\ \nabla I_{\mathrm{tgt}}^{Y} \begin{pmatrix} s_x + u \\ s_y + v \end{pmatrix} & \nabla I_{\mathrm{tgt}}^{U} \begin{pmatrix} s_x + u \\ s_y + v \end{pmatrix} \end{matrix} \right. \\ &\qquad \left. \begin{matrix} 0 & -\frac{1}{\alpha^2}[s_x \quad s_y] \\ \nabla I_{\mathrm{tgt}}^{V} \begin{pmatrix} s_x + u \\ s_y + v \end{pmatrix} & \left[\, 0 \,\right]_{[2\times 2]} \end{matrix} \right]. && \text{(C.15)} \end{aligned}$$

Matrix $\mathcal{D}_s$ has p lines corresponding to the number of parameters of the motion model φ and d columns corresponding to the dimension of the feature space (here, (Y, U, V, x, y)). After some steps, one gets

$$\begin{aligned}\nabla\mathcal{E}(\varphi) &= \sum_{s\in T_\varphi} \mathcal{D}_s(T_\varphi)\left(\frac{\nabla f_R(s)}{f_R(s)} - \frac{\nabla f_{T_\varphi}(s)}{f_{T_\varphi}(s)}\right) \\ &\quad + \frac{1}{|T_\varphi|}\sum_{s\in T_\varphi} \mathcal{D}_s(T_\varphi)\sum_{t\in T_\varphi}\frac{\nabla K_h(t-s)}{f_{T_\varphi}(t)} \qquad \text{(C.16)} \\ &= \sum_{s\in T_\varphi} \mathcal{D}_s(T_\varphi)\left(\frac{\nabla f_R(s)}{f_R(s)} - \frac{\nabla f_{T_\varphi}(s)}{f_{T_\varphi}(s)}\right. \\ &\quad \left. + \frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\nabla K_h(t-s)}{f_{T_\varphi}(t)}\right). \qquad \text{(C.17)}\end{aligned}$$

C.1.2 Term interpretation

Let us focus on the following term of (C.16)

$$\mathcal{A}(s) := \frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\nabla K_h(t-s)}{f_{T_\varphi}(t)}. \qquad \text{(C.18)}$$

When the number of samples $|T_\varphi|$ tends toward infinity, $\mathcal{A}$ tends toward

$$\mathcal{A}_\infty(s) = \int_{\mathbb{R}^d} f_{T_\varphi}(t)\,\frac{\nabla K_h(t-s)}{f_{T_\varphi}(t)}\,\mathrm{d}t. \qquad \text{(C.19)}$$

Since K_h is radially symmetric, we have

$$\forall x \text{ and } y \in \mathbb{R}^d \text{ such that } x = -y, \nabla K_h(x) = -\nabla K_h(y). \qquad \text{(C.20)}$$

Therefore, (C.19) convergences (at least weakly) toward zero.

C.1.3 kNN-based expression of the derivative

The first sum in (C.16) involves two terms that can be approximated using the Mean-Shift (C.1). The expression of the mean (C.2) can be replaced with its kNN equivalent [FH75]

$$\mu_B(s) = \frac{1}{k}\sum_{t\in\mathcal{N}_B(s)} t. \qquad \text{(C.21)}$$

where $\mathcal{N}_B(s)$ is a balloon neighborhood 5.35 centered at point s. In the second sum in (C.16), the PDF f_{T_φ} can also be replaced with its kNN expression (5.5). Therefore, using the Mean-Shift approximation, the derivative of the Kullback-Leibler divergence can be written as a kNN-based expression

$$\begin{aligned}k\,\nabla\mathcal{E}(\varphi) &= \sum_{s\in T_\varphi}\mathcal{D}_s(T_\varphi)\left(\frac{d+2}{\rho_k^2(R,s)}\sum_{t\in\mathcal{N}_B(R,s)}(t-s)\right. \\ &\quad -\frac{d+2}{\rho_k^2(T_\varphi,s)}\sum_{t\in\mathcal{N}_B(T_\varphi,s)}(t-s) \\ &\quad \left.+v_d\sum_{t\in T_\varphi}\rho_k^d(T_\varphi,t)\,\nabla K_{\rho_k(T_\varphi,t)}(t-s)\right) \\ &\qquad \text{(C.22)}\end{aligned}$$

where $K_{\rho_k(T_\varphi,t)}(\cdot - s)$ is a window of radius $\rho_k(T_\varphi, t)$ centered at s.

C.1.4 Term interpretation

Let us now focus on the term of (C.22) corresponding to the term studied in Section C.1.2

$$\mathcal{A}_{\mathrm{kNN}}(s) = \sum_{t\in T_\varphi} \rho_k^d(T_\varphi, t)\ \nabla K_{\rho_k(T_\varphi,t)}(t-s). \tag{C.23}$$

Window $K_{\rho_k(T_\varphi,t)}(\cdot - s)$ at t is equal to $1/(\rho_k^d(T_\varphi, t)\ v_d)$ if $|t-s| \le \rho_k(T_\varphi, t)$ and zero otherwise. A finite difference approximation can be used to write

$$\nabla K_{\rho_k(T_\varphi,t)}(t-s) = \begin{cases} \frac{1}{\rho_k^d(T_\varphi,t)\ v_d}\ \frac{s-t}{|s-t|} & \text{if } |s-t| = \rho_k(T_\varphi, t) \\ 0 & \text{otherwise} \end{cases}. \tag{C.24}$$

Therefore, term (C.23) can be approximated by

$$\mathcal{A}_{\mathrm{kNN}}(s) \simeq \frac{1}{v_d} \sum_{\substack{t\in T_\varphi \\ |t-s|=\rho_k(T_\varphi,t)}} \frac{s-t}{\rho_k(T_\varphi,t)}. \tag{C.25}$$

Finally,

$$\begin{aligned} k\ \nabla\mathcal{E}(\varphi) \quad \simeq \quad & \sum_{s\in T_\varphi} \mathcal{D}_s(T_\varphi) \left(\frac{d+2}{\rho_k^2(R,s)} \sum_{t\in\mathcal{N}_B(R,s)} (t-s) \right. \\ & - \frac{d+2}{\rho_k^2(T_\varphi,s)} \sum_{t\in\mathcal{N}_B(T_\varphi,s)} (t-s) \\ & \left. - \sum_{\substack{t\in T_\varphi \\ |t-s|=\rho_k(T_\varphi,t)}} \frac{t-s}{\rho_k(T_\varphi,t)} \right). \end{aligned} \tag{C.26}$$

APPENDIX D

GRADIENT OF THE OPTICAL FLOW ENERGY: MEAN-SHIFT-BASED EXPRESSION AND KNN IMPLEMENTATION

In the following part, for differentiation and for future possible extensions (in Chapter 10) we write a generic formula for E_H that can be either $E_{\mathrm{H-Data}}$ or $E_{\mathrm{H-Smooth}}$.

$$E_H(\mathrm{v}) = -\sum_{\mathrm{m}\in D} \ln f_t(t_m) \tag{D.1}$$

where $t_{\mathrm{m}} = t(\mathrm{m}, \mathrm{v}(\mathrm{m}))$ is a generic notation and can be either ϵ_m or $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$. Indeed we can write E_H as $E_{\mathrm{H-Data}}$ (7.8) with $t = \epsilon$ and as E_{Smooth} (7.13) with $t_m = [\mathrm{m}, \mathrm{v}(\mathrm{m})]$. A kernel estimate of f_t can be written as

$$f_t(t_{\mathrm{m}}) = \frac{1}{|D|} \sum_{\mathrm{n}\in D} K_h(t_{\mathrm{m}} - t_{\mathrm{n}}) \tag{D.2}$$

In order to minimize E_H, we differentiate energy E_H (D.1) with respect to the optical flow v.

As the energy (D.1) has a double dependency in v (this double dependency is in t, which hides v as $t_m = t(\mathrm{m}, \mathrm{v}(\mathrm{m}))$), the derivative can be separated into two terms:

$$\nabla_{\mathrm{v}} E_H(\mathrm{v}(\mathrm{m})) = \mathcal{A}(t_{\mathrm{m}}) + \mathcal{B}(t_{\mathrm{m}}) \tag{D.3}$$

with

$$\mathcal{A}(t_{\mathrm{m}}) = \frac{\nabla_t f(t_{\mathrm{m}})}{f(t_{\mathrm{m}})} \nabla_{\mathrm{v}} t_{\mathrm{m}} \tag{D.4}$$

and

$$\mathcal{B}(t_{\mathrm{m}}) = -\frac{\nabla_{\mathrm{v}} t_{\mathrm{m}}}{|D|} \sum_{\mathrm{n}\in D\backslash \mathrm{m}} \frac{\nabla_t K_h(t_{\mathrm{n}} - t_{\mathrm{m}})}{f(t_{\mathrm{n}})}. \tag{D.5}$$

The second term $\mathcal{B}$ is from the construction of the density function which depends on t (7.9). However, if we suppose the density function changes slowly when changing one feature from t (because there are enough sample points to estimate the PDF) we can write $\mathcal{B}(\mathrm{m}) \approx 0$. Moreover, one can note that using the sample mean entropy [VW97] instead of the resubstitution estimate of entropy [AL76], this term would be equal to zero.

Using the mean shift simplifications detailed in Chapter 5.5, we can simply estimate $\mathcal{A}(t_{\mathrm{m}})$.

Replacing (5.41) in (D.4), we have an expression of $\nabla_{\mathrm{v}} E_H$

$$\nabla_{\mathrm{v}} E_H(\mathrm{v}) = \nabla_{\mathrm{v}} t_{\mathrm{m}} . \frac{d_t + 2}{d_k(t_{\mathrm{m}})^2} [\mu_B(t_{\mathrm{m}}) - t_{\mathrm{m}}]. \tag{D.6}$$

where μ_B is the mean in a balloon neighborhood around t_{m} of size $d_k(t_{\mathrm{m}})$ the distance to the k-th nearest neighbor of t_{m}. We can now write $\nabla_{\mathrm{v}} E_{\mathrm{H-Data}}$ with $t_m = \epsilon_m$ and $\nabla_{\mathrm{v}} E_{\mathrm{Smooth}}(\mathrm{v})$ with $t_m = [\mathrm{m}, \mathrm{v}(\mathrm{m})]$.

Replacing t with the corresponding function, we have an expression for $\nabla_{\mathrm{v}} E_{\mathrm{H-Data}}$ and $\nabla_{\mathrm{v}} E_{\mathrm{Smooth}}$.In $\nabla_{\mathrm{v}} E_{\mathrm{H-Data}}$, t_m is replaced by $\epsilon_m = \epsilon(\mathrm{m}, \mathrm{v}(\mathrm{m}))$ and $\nabla_{\mathrm{v}} t$ by $\nabla_{\mathrm{v}} \epsilon$, which is a matrix $2 \times d_\epsilon$, $d_\epsilon = 6$ being the dimension of the image features u, ∇f being a vector $d_\epsilon \times 1$. In $\nabla_{\mathrm{v}} E_{\mathrm{Smooth}}$, t_m is replaced by $[\mathrm{m}, \mathrm{v}(\mathrm{m})]$ and $\nabla_{\mathrm{v}} t$ by $\mathbb{P}_2$, the projection operator $\mathbb{P}_2 .[\mathrm{m}, \mathrm{v}(\mathrm{m})] = \mathrm{v}(\mathrm{m})$.

APPENDIX **E**

GRADIENT OF AN ACTIVE CONTOUR SEGMENTATION

In this section we will focus on the general derivation of a bayesian segmentation problem with nonparametric distributions as formulated in (2.37). The energy (2.37)can be written in a more general form as

$$E(\Gamma) = -\frac{1}{|\Omega|} \int_{\Omega} \log f_{\Omega}(u(v(\Gamma), \mathrm{m}))\ \mathrm{dm} \tag{E.1}$$

where

$$\begin{cases} f_{\Omega}(r) = \dfrac{1}{|\Omega|} \displaystyle\int_{\Omega} K_{\sigma}(u(v(\Gamma), \mathrm{m}) - r)\ \mathrm{dm} \\ v(\Gamma) \ = \arg\min\limits_{w} -\dfrac{1}{|\Omega|} \displaystyle\int_{\Omega} \log f_{\Omega}(u(w, \mathrm{m}))\ \mathrm{dm} \end{cases} . \tag{E.2}$$

where $u(v(\Gamma), \mathrm{m}) = I(\mathrm{m})$ in [HSD$^+$06], in a motion estimation problem $u(v(\Gamma), \mathrm{m}) = I_n(\mathrm{m}) - I_{n+1}(\mathrm{m} + v(\Gamma))$.

Note that, for simplicity, residual u has been defined for a translation motion model. However, the following development is valid for any functions.

The definition of the shape derivative of (E.1) is based on a domain transformation T whose amplitude continuously depends on a parameter τ such that $T(\Omega, \tau = 0)$ is equal to Ω and $T(\Omega, \tau)$ is equal to $\Omega(\tau)$ [DZ01, HR04, JBBA03, ABFJB03]. Functions of Ω, or Γ, can then be rewritten as functions of τ. In this context, the shape derivative of

$$E(\Gamma) = \int_{\Omega} G(\Gamma, \mathrm{m})\ \mathrm{dm} \tag{E.3}$$

is equal to

$$\mathrm{d}E(\Gamma, F) = \frac{\mathrm{d}E}{\mathrm{d}\tau}(\tau = 0) = \int_{\Omega} \frac{\partial G}{\partial \tau}(\tau = 0, \mathrm{m})\ \mathrm{dm} - \int_{\Gamma} G(\Gamma, s)\ N(s) \cdot F(s)\ \mathrm{d}s \tag{E.4}$$

where F is a vector field defined on Γ and linked to T, s is the arclength parameter of Γ, $G(\Gamma, s)$ is a short notation for $G(\Gamma, \Gamma(s))$, and N is the inward unit normal of Γ.

Let us define $\mathcal{E}$ as follows

$$\mathcal{E}(\Gamma, w) = -\frac{1}{|\Omega|} \int_{\Omega} \log f_{\Omega}(u(w, \mathrm{m}))\ \mathrm{dm}\ . \tag{E.5}$$

Hence,

$$E(\Gamma) = \mathcal{E}(\Gamma, v(\Gamma)) \tag{E.6}$$

and

$$v(\Gamma) = \arg\min_{w} \mathcal{E}(\Gamma, w) \ . \tag{E.7}$$

Then, the shape derivative of (E.1) is equal to

$$\begin{aligned} \mathrm{d}E(\Gamma, F) &= \frac{\mathrm{d}\mathcal{E}}{\mathrm{d}\tau}\ (\tau, v(\tau))|_{\tau=0} & \text{(E.8)} \\ &= \frac{\partial\mathcal{E}}{\partial\tau}\ (\tau, v(\tau))|_{\tau=0} + \frac{\partial\mathcal{E}}{\partial w}\ (\tau, v(\tau))|_{\tau=0}\ \frac{\mathrm{d}v}{\mathrm{d}\tau}(\tau = 0) \ . & \text{(E.9)} \end{aligned}$$

Recalling that $\tau = 0$ corresponds to Γ and according to (E.7), the second term in (E.9) is equal to zero. Therefore, expression (E.9) is equal to

$$\mathrm{d}E(\Gamma, F) = \frac{\partial\mathcal{E}}{\partial\tau}\ (\tau, v(\tau))|_{\tau=0} \ . \tag{E.10}$$

Note that the derivative of $\mathcal{E}$ is taken with respect to the first variable, $v(\tau)$ being considered as a constant (including in all the following calculations). We compute the shape derivative of $\frac{1}{|\Omega|}$, the shape derivative of $|\Omega|$ is equal to

$$\begin{aligned} \mathrm{d}(|\Omega|)(\Gamma, F) &= \mathrm{d}\left(\int_{\Omega} \mathrm{dm}\right)(\Gamma, F) & \text{(E.11)} \\ &= \frac{\mathrm{d}}{\mathrm{d}\tau} \int_{\Omega(\tau)} \mathrm{dm}\Bigg|_{\tau=0} & \text{(E.12)} \\ &= \left[\int_{\Omega} \frac{\partial 1}{\partial\tau}(\tau = 0, \mathrm{m})\ \mathrm{dm} - \int_{\Gamma} N(s) \cdot F(s)\ \mathrm{d}s\right] & \text{(E.13)} \\ &= -\int_{\Gamma} N(s) \cdot F(s)\ \mathrm{d}s \ . & \text{(E.14)} \end{aligned}$$

The shape derivative of $\frac{1}{|\Omega|}$ is equal to

$$\begin{aligned} \mathrm{d}(1/|\Omega|)(\Gamma, F) &= \frac{\mathrm{d}}{\mathrm{d}\tau}\ \frac{1}{|\Omega(\tau)|}\Bigg|_{\tau=0} & \text{(E.15)} \\ &= -\frac{1}{|\Omega|^2}\ \frac{\mathrm{d}}{\mathrm{d}\tau} \int_{\Omega(\tau)} \mathrm{dm}\Bigg|_{\tau=0} & \text{(E.16)} \\ &= \frac{1}{|\Omega|^2} \int_{\Gamma} N(s) \cdot F(s)\ \mathrm{d}s \ . & \text{(E.17)} \end{aligned}$$

The classical rule for differentiating a product leads to

$$\begin{aligned} \mathrm{d}E(\Gamma, F) &= \frac{E(\Gamma)}{|\Omega|} \int_{\Gamma} N(s) \cdot F(s)\ \mathrm{d}s \\ &\quad -\frac{1}{|\Omega|}\ \frac{\mathrm{d}}{\mathrm{d}\tau} \int_{\Omega(\tau)} \log f_{\Omega}(u(v(\Gamma), \mathrm{m}))\ \mathrm{dm}\Bigg|_{\tau=0} & \text{(E.18)} \\ &= \frac{1}{|\Omega|} \left[\int_{\Gamma} E(\Gamma)\ N(s) \cdot F(s)\ \mathrm{d}s - \mathcal{A}\right] & \text{(E.19)} \end{aligned}$$

where f_{Ω} is also seen as a function of τ

$$f_{\Omega}(u(v(\Gamma), \mathrm{m})) = \frac{1}{|\Omega(\tau)|} \int_{\Omega(\tau)} K_{\sigma}(u(v(\Gamma), y) - u(v(\Gamma), \mathrm{m}))\ \mathrm{d}y \ . \tag{E.20}$$

Remember that $v(\Gamma)$ is considered as a constant and not as a function of τ as a result of the decoupling (E.6). Therefore, for clarity, $u(v(\Gamma),\cdot)$ will be denoted by $u(\cdot)$.

Term $\mathcal{A}$ can be computed by applying the general rule (E.4) successively

$$\mathcal{A} = \int_\Omega \frac{\partial \log f_\Omega}{\partial \tau}(\tau = 0, \mathrm{m})\ \mathbf{d}\mathrm{m} - \int_\Gamma \log f_\Omega(u(s))\ N(s)\cdot F(s)\ \mathbf{d}s \qquad \text{(E.21)}$$

$$= \int_\Omega \mathcal{B}\ \mathbf{d}\mathrm{m} - \int_\Gamma \log f_\Omega(u(s))\ N(s)\cdot F(s)\ \mathbf{d}s\ . \qquad \text{(E.22)}$$

Then,

$$\mathcal{B} = \frac{\frac{\partial f_\Omega}{\partial \tau}(\tau = 0, \mathrm{m})}{f_\Omega(u(\mathrm{m}))} \qquad \text{(E.23)}$$

$$= \frac{1}{f_\Omega(u(\mathrm{m}))} \left[\frac{f_\Omega(u(\mathrm{m}))}{|\Omega|} \int_\Gamma N(s)\cdot F(s)\ \mathbf{d}s \right.$$
$$\left. + \frac{1}{|\Omega|} \frac{\mathbf{d}}{\mathbf{d}\tau} \int_{\Omega(\tau)} K_\sigma(u(y) - u(\mathrm{m}))\ \mathbf{d}y \bigg|_{\tau=0} \right] \qquad \text{(E.24)}$$

$$= \frac{1}{|\Omega|} \left[\int_\Gamma N(s)\cdot F(s)\ \mathbf{d}s + \frac{\mathcal{C}}{f_\Omega(u(\mathrm{m}))} \right]\ . \qquad \text{(E.25)}$$

Finally,

$$\mathcal{C} = \int_\Omega \frac{\mathbf{d}K_\sigma(u(y) - u(\mathrm{m}))}{\mathbf{d}\tau}(\tau = 0)\ \mathbf{d}y$$
$$- \int_\Gamma K_\sigma(u(s) - u(\mathrm{m}))\ N(s)\cdot F(s)\ \mathbf{d}s \qquad \text{(E.26)}$$

$$= - \int_\Gamma K_\sigma(u(s) - u(\mathrm{m}))\ N(s)\cdot F(s)\ \mathbf{d}s \qquad \text{(E.27)}$$

since $K_\sigma(\dots)$ does not depend on τ. Gathering all the intermediate results together, the shape derivative of (E.1) is equal to

$$\mathbf{d}E(\Gamma, F) = \frac{1}{|\Omega|} \int_\Gamma \Bigg(E(\Gamma) - 1 + \log f_\Omega(u(s))$$
$$+ \frac{1}{|\Omega|} \int_\Omega \frac{K_\sigma(u(s) - u(\mathrm{m}))}{f_\Omega(u(\mathrm{m}))}\ \mathbf{d}\mathrm{m} \Bigg)\ N(s)\cdot F(s)\ \mathbf{d}s\ . \qquad \text{(E.28)}$$

APPENDIX **F**

MOTION SEGMENTATION

F.1 Disambiguation using spatial information

Energy (8.43) is well suited for segmenting objects over a textured background. However, it might cause segmentation to include homogeneous or quasi-homogeneous areas of the background. Indeed, this type of areas has a low residual even if compensated with the motion estimated for the object, at least as long as the motion-compensated object domain remains in the homogeneous area. Therefore, the energy might increase only negligibly when expanding in such areas. Since the notions of object and background are arbitrary and can be swapped for one another, one can note that an equivalent undersegmentation phenomenon can occur if the object contains homogeneous areas near its boundary.

On the other hand, the entropy of the object color increases if the object domain includes some background since it adds new colors to the object[1] and, therefore, increases the dispersion of its color distribution. Consequently, the joint entropy of the residual and the color also increases.

F.2 "Independence" between residual and color

Let us consider the following sequence model

$$I_{n+1}(x) = I_n(T(x)) + n(x) \tag{F.1}$$

where T is a transformation and n is a Gaussian white noise. The residual is equal to

$$e_n(v(x), x) = I_n(x) - I_{n+1}(v(x)) \; . \tag{F.2}$$

If the transformation T exists and the motion is perfectly estimated, then v is equal to T^{-1} and $e_n(v(x), x) = -n(T^{-1}(x))$, which is independent of I_n. However, model (F.1) is an approximation: in general, there is no such transformation T, frame I_{n+1} being a projection on a two-dimensional plane of a three-dimensional scene. Often, some parts of objects in I_n become invisible in I_{n+1} while others become visible. Therefore, frame I_{n+1} cannot be deduced

[1] If the background has the same color as the object near the boundary, there is no objective information to find the object boundary.

entirely from I_n. In the unpredictable areas, the residual is by definition independent of I_n. Overall, whether a transformation T exists or not, if the motion $v(\Gamma)$ is fairly well estimated, then the assumption of independence should be acceptable.

F.3 Energy derivative

Temporal energy

The temporal energy is equal to

$$E_t(\Gamma) = -\frac{1}{|\Omega|} \int_\Omega \log f_t(e_n(v(\Gamma), x)) \, \mathrm{d}x \tag{F.3}$$

where

$$\begin{cases} f_t(r) &= \dfrac{1}{|\Omega|} \displaystyle\int_\Omega K_\sigma(e_n(v(\Gamma), x) - r) \, \mathrm{d}x \\ e_n(v(\Gamma), x) &= \min_{\mathrm{abs}}(I_n(x) - I_{n+1}(x + v(\Gamma)), I_n(x) - I_{n-1}(x - v(\Gamma))) \\ v(\Gamma) &= \arg\min\limits_w -\dfrac{1}{|\Omega|} \displaystyle\int_\Omega \log f_t(e_n(w, x)) \, \mathrm{dm} \end{cases} . \tag{F.4}$$

Note that, for simplicity, residual e_n has been defined for a translation motion model. However, the following development is valid for any motion model.

Using the result of Appendix E the shape derivative of (F.3) is equal to

$$\begin{aligned} \mathrm{d}E_t(\Gamma, F) \quad = \quad & \frac{1}{|\Omega|} \int_\Gamma \bigg(E_t(\Gamma) - 1 + \log f_t(e_n(s)) \\ & + \frac{1}{|\Omega|} \int_\Omega \frac{K_\sigma(e_n(s) - e_n(\mathrm{m}))}{f_t(e_n(\mathrm{m}))} \, \mathrm{dm} \bigg) \; N(s) \cdot F(s) \, \mathrm{d}s \; . \end{aligned} \tag{F.5}$$

Spatial energy

The spatial energy is equal to

$$E_s(\Gamma) = -\frac{1}{|\Omega|} \int_\Omega \log f_s(I_n(x)) \, \mathrm{dm} \tag{F.6}$$

where

$$f_s(r) = \frac{1}{|\Omega|} \int_\Omega K_\sigma(I_n(x) - r) \, \mathrm{dm} \; . \tag{F.7}$$

Following the same approach as in Section F.3, it can be shown that the shape derivative of (F.6) is equal to

$$\begin{aligned} \mathrm{d}E_s(\Gamma, F) \quad = \quad & \frac{1}{|\Omega|} \int_\Gamma \bigg(E_s(\Gamma) - 1 + \log f_s(I_n(s)) \\ & + \frac{1}{|\Omega|} \int_\Omega \frac{K_\sigma(I_n(s) - I_n(x))}{f_s(I_n(x))} \, \mathrm{dm} \bigg) \; N(s) \cdot F(s) \, \mathrm{d}s \; . \end{aligned} \tag{F.8}$$

Shape derivative of $|\Omega|$ and $\frac{1}{|\Omega|}$

The shape derivative of $|\Omega|$ is equal to

$$\begin{aligned}
\mathrm{d}(|\Omega|)(\Gamma, F) &= \mathrm{d}\left(\int_\Omega \mathrm{dm}\right)(\Gamma, F) && \text{(F.9)}\\
&= \left.\frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\Omega(\tau)} \mathrm{dm}\right|_{\tau=0} && \text{(F.10)}\\
&= \left[\int_\Omega \frac{\partial 1}{\partial \tau}(\tau = 0, x)\ \mathrm{dm} - \int_\Gamma N(s)\cdot F(s)\ \mathrm{d}s\right] && \text{(F.11)}\\
&= -\int_\Gamma N(s)\cdot F(s)\ \mathrm{d}s\ . && \text{(F.12)}
\end{aligned}$$

The shape derivative of $\frac{1}{|\Omega|}$ is equal to

$$\begin{aligned}
\mathrm{d}(1/|\Omega|)(\Gamma, F) &= \left.\frac{\mathrm{d}}{\mathrm{d}\tau}\frac{1}{|\Omega(\tau)|}\right|_{\tau=0} && \text{(F.13)}\\
&= -\frac{1}{|\Omega|^2}\left.\frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\Omega(\tau)} \mathrm{dm}\right|_{\tau=0} && \text{(F.14)}\\
&= \frac{1}{|\Omega|^2}\int_\Gamma N(s)\cdot F(s)\ \mathrm{d}s\ . && \text{(F.15)}
\end{aligned}$$

F.4 Piecewise motion decomposition

The following development should give some intuitions to study the validity of the piecewise motion decomposition. As will be clear from the concluding remarks, it does not provide a full and rigorous analysis.

The frame I_n is divided into blocks B_i of identical size. Let Ω_i be the intersection of Ω with B_i and let Γ_i be the boundary $\partial\Omega_i$ of Ω_i (see Fig. 8.8). For clarity, $v(\Gamma_i)$ will be denoted by v_i. Energy (F.3) is replaced with

$$E_t^{\mathrm{local}}(\Gamma) = -\frac{1}{|\Omega|}\int_\Omega \log f_t(e_n(v_1, \ldots, v_k, x))\ \mathrm{dm} \tag{F.16}$$

where

$$\left\{\begin{array}{l}
f_t(r) = \frac{1}{|\Omega|}\int_\Omega K_\sigma(e_n(v_1, \ldots, v_k, x) - r)\ \mathrm{dm}\\
e_n(v_1, \ldots, v_k, x) = \mathrm{min_{abs}}(I_n(x) - I_{n+1}(x + v_i), I_n(x) - I_{n-1}(x - v_i))\\
\ \text{if } x \in \Omega_i\\
\\
v_i = \arg\min_w -\frac{1}{|\Omega|}\int_{\Omega_i} \log f_t(e_n(v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_k, x))\ \mathrm{dm}
\end{array}\right. . \tag{F.17}$$

Note that the motions $v_j, j \neq i$, in the energy minimized to solve for v_i are irrelevant constants since they are not used in the computation of the residual e_n on Ω_i.

Let us define $\mathcal{E}_t^i$ as follows

$$\mathcal{E}_t^i(\Gamma, w_1, \ldots, w_k) = -\frac{1}{|\Omega|}\int_{\Omega_i} \log f_t(e_n(w_1, \ldots, w_k, x))\ \mathrm{dm}\ . \tag{F.18}$$

According to the remark on the residual above, it can be concluded that $\mathcal{E}_t^i$ is independent of $w_j, j \neq i$.

Energy (F.16) is equal to,

$$E_t^{\mathrm{local}}(\Gamma) = \sum_i \mathcal{E}_t^i(\Gamma, v_1, \ldots, v_k)) \tag{F.19}$$

and

$$v_i = \arg\min_w \mathcal{E}_t^i(\Gamma, v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_k)\ . \tag{F.20}$$

Then, the shape derivative of (F.16) is equal to

$$\begin{aligned} \mathrm{d}E_t^{\mathrm{local}}(\Gamma, F) &= \sum_i \frac{\mathrm{d}\mathcal{E}_t^i}{\mathrm{d}\tau}\ (\tau, v_1(\tau), \ldots, v_k(\tau))|_{\tau=0} & \text{(F.21)} \\ &= \sum_i \frac{\partial\mathcal{E}_t^i}{\partial\tau}\ (\tau, v_1(\tau), \ldots, v_k(\tau))|_{\tau=0} \\ &\quad + \sum_i \sum_j \underbrace{\frac{\partial\mathcal{E}_t^i}{\partial w_j}\ (\tau, v_1(\tau), \ldots, v_k(\tau))|_{\tau=0}}_{\mathcal{A}_j^i}\ \frac{\mathrm{d}v_j}{\mathrm{d}\tau}(\tau = 0)\ . & \text{(F.22)} \end{aligned}$$

Recalling that $\tau = 0$ corresponds to Γ (and, therefore, $v_i(\tau = 0) = v_i$), $\mathcal{A}_j^i$ is equal to zero if j is equal to i because of (F.20). Moreover, according to the independence of $\mathcal{E}_t^i$ with respect to $w_j, j \neq i$, $\mathcal{A}_j^i$ is also equal to zero if j is not equal to i. Therefore, expression (F.22) is equal to

$$\mathrm{d}E_t^{\mathrm{local}}(\Gamma, F) = \sum_i \frac{\partial\mathcal{E}_t^i}{\partial\tau}\ (\tau, v_1(\tau), \ldots, v_k(\tau))|_{\tau=0}\ . \tag{F.23}$$

By definition, the shape derivative is based on a domain transformation T operating on Ω (see Appendix F.3). Energy $\mathcal{E}_t^i$ is an integral over Ω_i. Its shape derivative is naturally related with the restriction of T to Ω_i. However, f_t is still an integral over Ω. Keeping that in mind, the approach of Section F.3 can be followed to determine the shape derivative of (F.16)

$$\begin{aligned} \mathrm{d}E_t^{\mathrm{local}}(\Gamma, F) &= \frac{E_t^{\mathrm{local}}(\Gamma)}{|\Omega|} \int_\Gamma N(s) \cdot F(s)\ \mathrm{d}s \\ &\quad - \frac{1}{|\Omega|} \sum_i \frac{\mathrm{d}}{\mathrm{d}\tau} \int_{\Omega_i(\tau)} \log f_t(e_n(v(\Gamma), x))\ \mathrm{dm} \Bigg|_{\tau=0} & \text{(F.24)} \\ &= \frac{1}{|\Omega|} \left[\int_\Gamma E_t^{\mathrm{local}}(\Gamma)\ N(s) \cdot F(s)\ \mathrm{d}s - \sum_i \mathcal{A}_i \right]\ . & \text{(F.25)} \end{aligned}$$

For clarity, $e_n(v_1, \ldots, v_k, \cdot)$ will be denoted by $e_n(\cdot)$.

Term $\mathcal{A}_i$ is equal to

$$\begin{aligned} \mathcal{A}_i &= \int_{\Omega_i} \frac{\partial \log f_t}{\partial\tau}(\tau = 0, x)\ \mathrm{dm} - \int_{\Gamma_i} \log f_t(e_n(s))\ N_i(s) \cdot F(s)\ \mathrm{d}s & \text{(F.26)} \\ &= \int_{\Omega_i} \mathcal{B}\ \mathrm{dm} - \int_{\Gamma_i} \log f_t(e_n(s))\ N_i(s) \cdot F(s)\ \mathrm{d}s & \text{(F.27)} \end{aligned}$$

where N_i is the inward unit normal of Γ_i. Term $\mathcal{B}$ is identical to the corresponding term (E.25) in Appendix E, *i.e.*,

$$\mathcal{B} = \frac{1}{|\Omega|} \int_\Gamma \left(1 - \frac{K_\sigma(e_n(s) - e_n(\mathrm{m}))}{f_t(e_n(\mathrm{m}))} \right)\ N(s) \cdot F(s)\ \mathrm{d}s\ . \tag{F.28}$$

Gathering all the intermediate results together, the shape derivative of (F.16) is equal to

$$\mathrm{d}E_t(\Gamma, F) = \frac{1}{|\Omega|}\left[\int_\Gamma \left(E_t^{\text{local}}(\Gamma) - 1 + \frac{1}{|\Omega|}\int_\Omega \frac{K_\sigma(e_n(s) - e_n(\mathrm{m}))}{f_t(e_n(\mathrm{m}))}\,\mathrm{dm}\right)\right.$$

$$\left. N(s)\cdot F(s)\,\mathrm{d}s - \underbrace{\sum_i \int_{\Gamma_i} \log f_t(e_n(s))\; N_i(s)\cdot F(s)\,\mathrm{d}s}_{\mathcal{S}}\right]. \tag{F.29}$$

Let B_i and B_j be 2 adjacent blocks with boundaries Γ_i and Γ_j, respectively. On their common boundary, $\log f_t(e_n)$ and F are uniquely defined. However, N_i and N_j have opposite directions, each pointing inward relatively to its (oriented) boundary. Therefore, the sum of the integrals over Γ_i and Γ_j in $\mathcal{S}$ on this common boundary is equal to zero. When considering all the blocks, the only portions of integral that remain of $\mathcal{S}$ are the ones which are not in common with any other block boundary. These portions sum to Γ. The normals N_i on these portions are equal to N. In conclusion, shape derivative (F.29) is identical to (F.5): it seems that this hierarchical motion decomposition approach can be safely used with minimal changes to the implementation (only the residual computation changes). However, one condition has not been mentioned so far. The shape derivative framework is valid for smooth contours. In particular, the presence of the contour normal in the expressions implicitly requires that the contour be at least continuously differentiable. Unfortunately, the contours Ω_i of the proposed partition of Ω are not smooth, independently of the smoothness of Ω. Actually, any paving of Ω using patches contains multiple junctions. As a consequence, the previous development is theoretically invalid. Nevertheless, the set of singularities is finite and it might be possible to rigorously confirm the result by studying the limit of a related, smooth setting similar to some works on classification [SBFGJ00]. Moreover, in practice, the (wrongly) obtained result can be easily implemented since it does not involve these singularities.

F.5 Parametric assumptions

F.5.1 Residual

If the residual e_n is a spatially uncorrelated random field with a Laplacian distribution with mean μ_e and scale σ, the probability of having a given field, conditional to a motion v, is equal to

$$p(e_n|v) = \frac{1}{(2\sigma)^{|\Omega|}} \prod_{x\in\Omega} \exp -\frac{|e_n(v,x) - \mu_e|}{\sigma}\ . \tag{F.30}$$

The maximum log-likelihood estimation of v is given by

$$\arg\min_v \sum_{x\in\Omega} |e_n(v,x) - \mu_e|\ . \tag{F.31}$$

In practice, choosing μ_e different from zero can only be motivated by a global change of illumination occurring between frames I_n and I_{n+1}. Making the assumption that the global illumination remains constant, μ_e will be set to zero. Therefore, estimation (F.31) is equivalent to

$$\arg\min_v \sum_{x\in\Omega} |I_n(x) - I_{n+1}(x+v)| \tag{F.32}$$

which is the SAD criterion.

F.5.2 Color

If the color I_n is a spatially uncorrelated random field with a Gaussian distribution with mean μ_I and standard deviation σ, the probability of having a given field, conditional to a motion v, is equal to

$$p(I_n|v) = \frac{1}{\sqrt{2\pi}\ \sigma} \prod_{x \in \Omega} \exp - \frac{(I_n(x) - \mu_I)^2}{2\sigma^2}\ . \tag{F.33}$$

The maximum likelihood estimation of v is then equivalent to minimizing the Sum of Squared Differences (SSD)

$$\arg\min_v \sum_{x \in \Omega} (I_n(x) - \mu_I)^2\ . \tag{F.34}$$

In practice, μ_I can be approximated by the mean of I_n in Ω [JBBA03].

APPENDIX G

PUZZLE: OBJECT DRIVEN VIDEO CODING

The goal of this section is to propose a simple method to estimate motion segmentation in video sequences for video coding purposes. Most video coders, including MPEG-like coders, recent wavelet-based coders [CAAB04] and H264 coders, use simple motion estimators based on block matching (BM) algorithms. These algorithms classically split the image into blocks of regular size (called Macroblocks (MBs)). Even though these algorithms are fast and quite accurate, they still have some issues : indeed, since the subdivision into blocks does not match the positions of the moving objects, some blocks overlap regions with different motions, which creates blocking artifacts in the coded-decoded sequence. To overcome this problem, H264 coders test various modes for coding MBs, adapted for different cases and selects the most efficient, one for each MB, efficient in terms of final Rate Distortion (RD) *i.e.* the ratio between the amount of distortion in the final video brought by a mode, and the coding rate of the mode. Among the possible modes in competition are "intra", Fig. G.1(a) which skips motion estimation and codes the MB frame by frame; "inter", Fig. G.1(b) which performs motion estimation and compensation on the whole MB; "inter+4v", Fig. G.1(c) which splits the MB into 4 smaller blocks and performs motion estimation and compensation. Our contribution is a new coding mode for coding MBs, called "split", Fig. G.1(d) derived from a motion segmentation algorithm. The split mode splits the MB into 2 regions.

Indeed, one should consider a segmentation of the moving objects. Unfortunately, most of the image segmentation techniques based on active contours [ABFJB03, CKS97, CV01, CS05, JBBA03] are not fully automatic, and are too complex to be implemented within a video coder. For this reason, we propose a simplified active contour approach to estimate motion and segmentation simultaneously in a local context. More precisely, we divide the image into MBs in which segmentation is performed independently. We suppose that there are at most two regions with different motion in each MB.

In order to decide whether each MB should be splitted or not, a block selection process will be run. The selected blocks are now distinct joint motion segmentation problems and we will define a cost functional to solve them simultaneously. Finally, we will show how this model allows to avoid occlusion.

An active contour description of the segmentation is usually quite finer than what is actually needed: since the blocks are relatively small, a high order spline description of a border is not really necessary, while a simple straight line border would be almost as much effective but much cheaper in terms of coding rate. Moreover, an active contour algorithm is not realistic for video coding applications as it suffers from a possible infinite number of iterations. We kept the same framework and energy but implemented a simplified minimization, with lines instead of splines, and testing a finite number of configurations. This technique is unable to provide a global segmentation of the image, but this information is not mandatory for coding purposes. Finally this mode was implemented in the Orange (the French national operator) H264 video coder.

This section is organized as follows: Section G.1 presents motivations, Section G.2 gives the motion segmentation algorithm, Section G.3 defines a simplified algorithm of the split mode for industrial purpose, Section G.4 presents results on the split mode, finally Section G.5 concludes.

G.1 Basic Idea and Motivations

G.1.1 Blocks versus Regions

A simple block-based description of the motion in a scene is often not accurate enough, because in some area a finer description of the motion can be necessary. This happens usually at object borders, or when small moving objects are present in the scene.

A common solution in these cases is to further divide the MB into four smaller sub-blocks, see Fig. G.1(c). This case is what we call inter+4v in this section. In fact, with this strategy, each MB is analyzed, and according to some suitable criterion, a decision is made whether this MB should be divided or not into four sub-blocks. If the MB must be divided, for each sub-block we can run the same BM algorithm used for the larger block, that is for example, a SAD-based search into the reference frame.

When the inter+4v approach is chosen, for each MB we should send the information signaling whether the MB has been divided or not, and, according to this information, one or four motion vectors.

This approach is simple and has the advantage that the BM algorithm for the sub-blocks can be the same used for the MB. Moreover in principle, the sub-blocks can be further divided.

The main drawback of this approach is that the division of a MB in four symmetric sub-block is completely arbitrary, and nothing assures that it is well suited to describe the motion of the objects. A common situation in which this approach is inefficient is when there is an object moving on a (fixed or moving) background. If the current MB is amid the object and the background, the inter+4v description can be ineffective because:

1. it is not assured that the subdivision separates the object and the background; and

2. four vectors are used when there are only two movements to describe.

From these considerations it is clear that a better description of the movements could be given if we knew where the object and the background are. If we know the curve describing the border between them, we could have a very effective representation of the motion in the scene, like in Fig. G.1(d). In this case, the motion is described by the curve and the two motion vectors. A parametric description of the curve can be very effective. For example, the curve in Fig. G.1(d) is a spline, and it is completely described by its four control points, highlighted with small circles.

Even when this segmentation strategy (called split throughout this section) is chosen, the general algorithm is similar to the one described for the inter+4v case: first we use a suitable criterion to decide if a MB must be splitted or not, and if it has to be splitted, a suitable BM algorithm is performed on it. This model might not perform better than the block-based model with small blocks in all cases. However it should be a good alternative specially on object borders, as shown in figure G.1

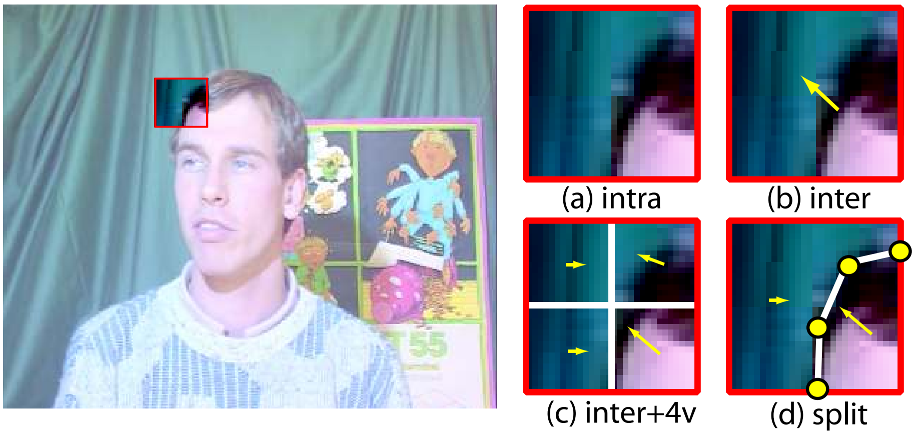


Figure G.1: *4 modes in competition for coding Macroblocks in video coding, three classical modes with a new one (a) "intra": no motion estimation is performed, (b) "inter": a global motion estimation is performed on all the macroblock (c) "inter+4v": motion estimation is split into 4 smaller blocks (d) "split": a new mode adapted for object boundaries is introduced*

G.1.2 Transmission cost

Let us compare the transmission cost between two methods inter+4v and split : the first method divides a block into 4 smaller blocks (4 motion vectors); whereas the second method divides it into 2 regions (4 control points and 2 motion vectors), figure G.1. The precision of the motion vectors is typically 1/8 pixel precision, and the precision of control points is one pixel. Moreover 2 out of 4 control points are on the border of the block so these points are represented by only one parameter. The region-based representation can be coded using 5 vectors: 2 control points, 1 vector made of 2 parameters and 2 region motion vectors, versus 4 vectors with smaller blocks. The transmission cost in each MB is then : 2 motion vectors at 1/8 pixel precision plus 3 control points position at pixel precision. With the smaller blocks method it would be 4 motion vectors at 1/8 pixel precision. In a coder, this represents $64bits$/MB for our method versus $64bits$/MB for the block-based method. Assuming that not all blocks are divided by a spline, we even obtain a smaller motion information in the compressed video with the bonus of a greater spline precision Fig. G.1.

G.2 Motion segmentation

G.2.1 Criterion definition

A region in the frame will be defined by its motion, also called optical flow. Let $I(m, i)$ be a video sequence, m the spatial coordinates, i the frame number, and v the optical flow between image i and image $i + 1$. v is a vector field representing an apparent motion related to a local gray-scale coherence between two consecutive images [HS81, NE86, WS01, DGBA05] .

$$(I(\mathrm{m}, i) - I(\mathrm{m} + v, i + 1))^2 = 0 \tag{G.1}$$

As presented in the chapter 7 on optical flow, (G.1) has several solutions, since many points in an image have the same gray-scale value. Therefore, the problem of computing the motion of a point must be regularized. First, (G.1) can be extended to a domain surrounding this point, second, we assume that the optical flow v constant over a region Ω. Therefore, region Ω should be a minimizer of the following energy (G.2):

$$\left\{\begin{array}{rcl} E(\Gamma) & = & \displaystyle\int_\Omega (I(\mathrm{m}, i) - I(\mathrm{m} + v(\Gamma), i + 1))^2 \,\mathrm{dm} \\ v(\Gamma) & = & \displaystyle\arg\min_v \int_\Omega (I(\mathrm{m}, i) - I(\mathrm{m} + v, i + 1))^2 \,\mathrm{dm} \end{array}\right. \tag{G.2}$$

The cost functional E (G.2) is minimized to solve motion and segmentation problems simultaneously.

For higher robustness, the functional is defined on a set of two frames surrounding the image of interest, a forward and a backward frame (previous equation is forward only), and we constrain the motions v computed backward and forward to be equal i.e uniform motion assumption:

$$\begin{array}{rcl} k(\mathrm{m}, \mathrm{v}) & = & (I(\mathrm{m}, i) - I(\mathrm{m} + \mathrm{v}, i + 1))^2 \\ & + & (I(\mathrm{m}, i) - I(\mathrm{m} - \mathrm{v}, i - 1))^2 \end{array} \tag{G.3}$$

$$\left\{\begin{array}{rcl} E(\Gamma) & = & \displaystyle\int_\Omega k(\mathrm{m}, \mathrm{v}(\Gamma)) \,\mathrm{dm} \\ \mathrm{v}(\Gamma) & = & \displaystyle\arg\min_{\mathrm{v}} \int_\Omega k(\mathrm{m}, \mathrm{v}) \,\mathrm{dm} \end{array}\right. \tag{G.4}$$

A classical block matching method is used to compute motion. However the matching is performed with regions instead of blocks, and more precisely a fast suboptimal matching algorithm is used: the Diamond Search [ZM00].

G.2.2 Segmentation through region competition

In order to find the region that minimizes the cost functional, we use a region competition algorithm. For instance, the functional E for two regions including a regularization term can be written as follows:

As the two regions form a partition of the block there is only one unknown, the boundary Γ defining two regions Ω and $\overline{\Omega}$. We note $\partial\Omega = \Gamma$ and $\partial\overline{\Omega} = \Gamma^c$.

$$E(\Gamma) = \int_\Omega k(\mathrm{m}, v(\Gamma)) \,\mathrm{dm} + \int_{\overline{\Omega}} k(\mathrm{m}, v(\Gamma^c)) \,\mathrm{dm} + \int_{\partial\Omega} \beta \,\mathrm{d}t \tag{G.5}$$

The first and second terms are the energy (G.2) applied on the two regions of the block, and the last term is the regularization term where $\Gamma = \partial\Omega$ is the contour between the two regions and β a constant, Γ and Γ^c are identical up to a change of orientation: $\Gamma^c = \partial\overline{\Omega}$. Differentiating this functional is done through shape gradients. A shape gradient model [ABFJB03, DZ01, JBBA03] is used to make the energy depend on an evolution parameter τ:

$$\begin{aligned} \mathrm{d}E_t(\Gamma, F) &= \int_\Gamma (k(s, v(\Gamma^c)) - k(s, \mathrm{v}(\Gamma)))N(s).F(s)ds \\ &+ \int_\Gamma (-\beta\kappa)(N(s).F(s))ds \end{aligned} \tag{G.6}$$

F is the unknown local deformation of Γ and N is the inward unit normal to Γ. Details on this shape derivative can be found Appendix E (assuming the probability is Gaussian).

The derivative (G.6) must be negative to go towards the minimum of the functional. The evolution equation $\frac{\partial\Gamma}{\partial\tau}$ is then:

$$\frac{\partial\Gamma(s)}{\partial\tau} = (k(s, v(\Gamma^c)) - k(s, v(\Gamma)) + \beta.\kappa).N(s) \tag{G.7}$$

We use active parametric contours to model the boundary Γ. Γ is represented by an open spline, the first and last control points of the splines are located on a block border and their evolution are also projected to stay on the border. An explicit parametrization of the active contour is performed by interpolating a spline between the control points.

G.2.3 Implementation details

Let us first present initialization and block selection. A block of homogeneous motion does not require to be splited by a spline. The algorithm must select the blocks to be divided and, at the same time, must initialize a first spline in these blocks. This selection is a three-step procedure:

- First, every block is divided into 4 smaller blocks. Then are computed the motion vectors in these smaller blocks with a block matching algorithm. Finally we compute a normalized distance between each pair of vectors:

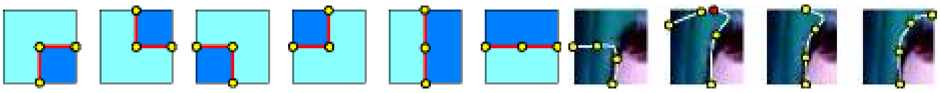
$$d = \max_{i=1..4, j>i} \frac{\|v_i - v_j\|}{\min(\|v_i\|, \|v_j\|)} \tag{G.8}$$

  and we threshold this value to choose if a block should be splitted by a spline or not.

- As a requirement for compression applications, we threshold the mean value of the prediction error at initialization, which helps to produce an effective segmentation for video coding rather than a regularized one.

- Finally, we threshold the same criterion (G.8) applied to the motion vectors of the two regions delimited by the spline

In addition, we use the first threshold to initialize a first spline in the block : the blocks (i, j) found by maximizing d in (G.8) defines the two classes of motions. The two other blocks are classified whether they are closer from the motion of i or the motion of j, closer in the sense of the same normalized distance. The motion classification leads to six different possible initializations made of control points splitting the blocks. Topology management assumes that a block is composed of at most two connex regions separated by a spline. However if the



(a) Six possible initializations (b) Topology Management

Figure G.2: *Implementation*

spline reaches a border, it splits the block in 3 regions; in this case the spline is cut into two parts and the shortest one is discarded, so that only two regions remain (See Fig. G.2).

One must take occlusion into account. As the problem is local we suppose there is only one kind of occlusion which happens only in backward or forward estimation. To correct this problem, a weighting between forward and backward estimations will be used. For the bidirectional prediction, the weighting is the same in both directions, we will adjust the weightings in forward prediction or backward prediction if occluded parts are detected. The new criterion with weightings is thus described in (G.9).

$$\begin{aligned} k(\mathrm{m}, v) &= c_f * \varphi((I(\mathrm{m}, i) - I(\mathrm{m} + v, i + 1))) \\ &+ c_b * \varphi((I(\mathrm{m}, i) - I(\mathrm{m} - v, i - 1))) \end{aligned} \tag{G.9}$$

where φ is a penalizing function, for simplicity it is chosen to be $\varphi(x) = x^2$. The occlusion detection method is now to be defined as well as the set of weightings. The constraint Block Matching algorithm gives us two values of the criterion (G.5): one forward and one backward. Comparing these two values, we can assume that if the forward (resp. backward) criterion value is some percentage higher than the other backward (resp. forward) criterion value there is an occlusion problem, so we set c_f (resp. c_b) to 0 and the other to 2. Otherwise, we use the constraint bidirectional method, so c_f and c_b are set to 1.

G.2.4 Experimental results

The proposed method was tested on the sequence "Eric", on the frames 18 to 24 which represent a quite uniform translation, needed by the bidirectional constraint. Let us analyze the result on the mid frame 21, the result is shown on the left of Fig. G.3. The method seems to perform well. Note that some blocks at the bottom of the frame were not divided by a spline because the background is quite homogeneous, so even with Eric's motion, the prediction error is lower than the energy's threshold. Left figure G.3 shows that segmentation splines are actually located a few pixels away from the object to be segmented. This is due to the background being occluded by Eric. Indeed, since criterion (G.3) is bidirectional, occluded background parts are on both sides of Eric. The occluded background on the Eric's sequence represents up to 10% of a block, which is much more important than in a classical algorithm on the whole image where occluded parts represent about 1% of the image. Visually, we observe an important diminution of the wrong classified pixels, the splines are much closer to Eric; we can also notice some improvements of the selection algorithm behavior; a spline at highlighted block wrongly removed by criterion (G.3), Fig. G.3 are now back in the video, Fig. G.3.

We compare the segmentation results with and without occlusion management. We count the wrong-classified points in the two cases and we compare the results with a manual segmentation. There are 1400 wrong pixels using our method without occlusion management

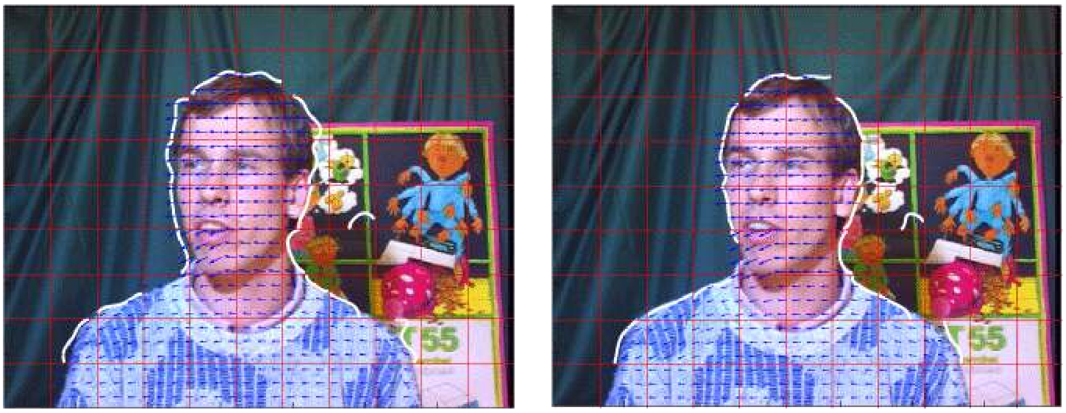


Figure G.3: *Macroblock spline segmentation: without occlusion and with occlusions management*

and only 1000 wrong pixels using our method with occlusion management, see figure G.3. In order to estimate the performances in terms of prediction error the proposed segmentation method was applied to the 8 frames of sequence Eric. Each frame was processed using the next and previous frame as detailed above. Tab. G.1 presents the prediction error energy per frame (PEEF), averaged over the 8 frames. The PEEF is defined as follows:

- inter+4v mode: the PEEF is equal to the sum of the prediction errors (from (G.3)) of the 4 blocks composing a MB, summed over each MB in which inhomogeneous motion was detected.
- split mode: the PEEF is equal to the sum of the prediction errors (from (G.3)) of the 2 regions composing a MB, summed over each MB in which inhomogeneous motion was detected, i.e splited by a spline.

By definition, in both cases, the same MBs are considered. The proposed method leads to a decrease of about $1/3$ of the average PEEF on the 8 frames. The results shown on the 8 frames in Tab. (G.1) in terms of PEEF, are not better because our error criterion (G.3) does not take account on occlusions. However using an adaptive filtering, as presented in [AAB05a, AAB05b], this accurate segmentation should provide better results.

Macroblock division	4 blocks	2 regions	gain in %
Method 1	59.9	39.1	33.90
Method 2	57.9	42.8	25.83

Table G.1: *Average PEEF on the split macro-blocks, first row: with occlusion management, second row: without occlusion management*

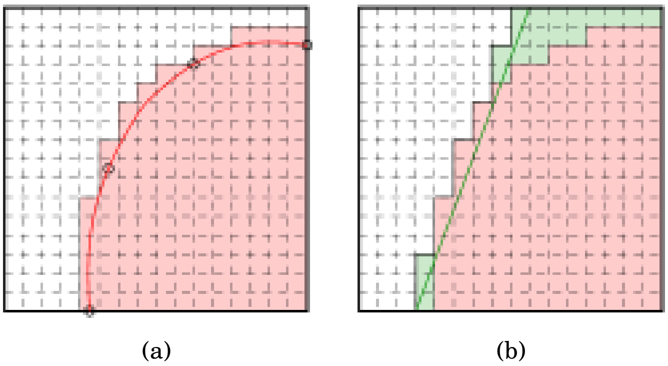
G.3 The split algorithm: simplified motion segmentation

As a gradient descent of active contours suffers from possible infinite number of iterations, we consider a finite alternative: instead of splitting the block with a spline, the block is splitted by a line.

G.3.1 Segmentation Method

We observe that a BM algorithm producing this segmentation information gives a very rich description of the motion, which however is quite expensive in terms of rate needed to represent it. In fact, in the example of Fig. G.1, we need to send two vectors and four control points for each MB. Such a description of the segmentation is usually quite finer that what is actually needed: since the blocks are relatively small, a high order spline description of a border is not really necessary, while a simple straight line border would be almost as much effective but much cheaper in terms of coding rate. This is visually justified in Fig. G.4. The spline contour originates the region in light red in the left part of the figure, while the straight line contour originates the region in green and red in the right part. As it can been seen, only a few pixels are misclassified by the straight line contour, which on the other hand is much less expensive to describe.

In order to further simplify the description of the split, the extremes of the segment can only be in some fixed positions on the MB perimeters. A parameter called *step* is introduced: its values corresponds to the number of allowed positions for the points on the MB perimeter.



(a)

(b)

Figure G.4: *Digital contours: (a) spline, (b) line.*

In conclusion, the algorithm used in this section produces a block-based segmentation; for each MB a decision is taken whether it should be divided or not, and in the case a positive decision is taken, the MB is segmented by a straight line. It follows that in the split case we have to send for each MB the information about the position of the split, and two motion vectors. Moreover, in the case the MB is splitted, the BM algorithm computes the four motion vectors related to the inter+4v case as well.

In conclusion, the operation flow of the algorithm is the following:

- For each MB a single motion estimation is computed, like in the inter mode.
- A test on the effectiveness of the inter mode can be carried out. If the test fails, the split algorithm is actually performed[1].
- When required, the split algorithm is performed. The parameter to be optimized are jointly the split position and the two motion vectors. Since the test space is very large, a suboptimal iterative research technique is performed: first some fixed split positions are tested, then they are refined.

[1] We note that this decision should be taken with a rate distortion optimization technique, so in a second moment we will force the algorithm to always perform the split algorithm, so we can choose among various modes.

- Finally, the algorithm produces a inter+4v estimation for the blocks where the test fails. It means that for these MBs we have also the finer motion description corresponding to smaller-sized blocks.

G.3.2 Criterion and split decision

The encoder must decide whether each MB should be splitted into two regions (or four blocks, in the inter+4v case) or not. This decision affects very much the compression performances and should be taken on a rate distortion-optimized basis. However, the first version of the algorithm used a different criterion. For each MB the ME algorithm computes the full-block backward and forward motion vectors. Then it computes the backward and forward motion-compensated predictions, which are used to evaluate the error (G.9) integrated on the macroblock. In the experiments, for robustness purposes, absolute value is preferred to square penalizing function and it is computed on the three color channels, the criterion is then a sum of absolute values over the three color channels (SAD-3D). The energy functional in each MB is then:

$$E(\Gamma_i) = \int_{\Omega_i} k(\mathrm{m}, \mathrm{v}(\Gamma_i))\, \mathrm{dm} + \int_{\overline{\Omega_i}} k(\mathrm{m}, \mathrm{v}(\Gamma_i^c))\, \mathrm{dm} \tag{G.10}$$

where Γ_i is a simple line, splitting the blocks into two regions Ω_i and Γ_i, where expression $k(\mathrm{m}, \mathrm{v}(\Gamma_i))$ is given in (G.9) with $\varphi(x) = |x|$.

Finally, the idea behind this algorithm is that the MBs in which the error is important should be splitted. However it is clear that such an approach is suboptimal. The decision to split a MB should be taken after a RD-optimization algorithm. Some segmentation map produced by this algorithm are shown here (see Fig. G.5).

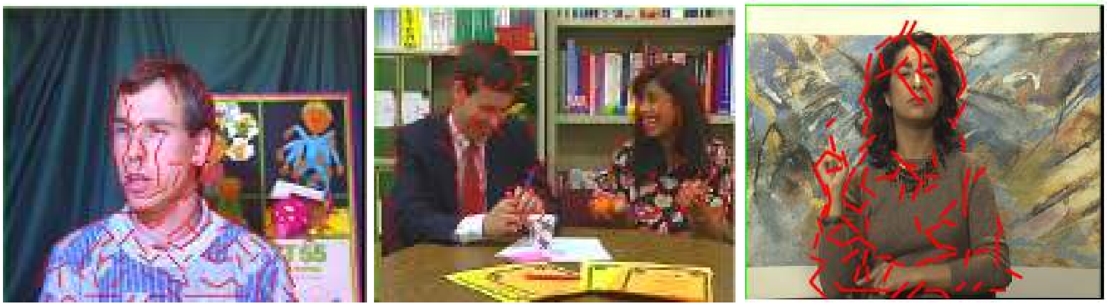


Figure G.5: *Examples of segmentation produced by the split algorithm*

G.3.3 Motion estimation for the object and the background

Once the decision to split the MB has be taken, the algorithm must find:

- the optimal segmentation of the MB in foreground and background;
- the optimal motion vectors.

Not all the possible splits are tested in order to find the best one. This mode method will be referred to as “split-fast”, an exhaustive version which tests all configurations with two given precisions (one for motion segmentation and one for estimation) will be referred to as “split-full”. The “split-fast” mode first tests 8 configuration, and the best among them is kept.

Mode	inter	inter+4v	split-fast	split-full
Fast	110570	128560	64038	63767

Table G.2: *Sum of criterion SAD-3D value over all blocks on "Boston" sequence)*

Mode	inter	inter+4v	split-fast	split-full
Fast	470910	421230	400910	396230

Table G.3: *Sum of criterion SAD-3D value over all blocks on "Football" sequence*

From it, we derive a subset of possible splits (obtained by small perturbation), and we look the best segmentation among these splits.

The first 8 initial segmentation of the MB are shown in Fig. G.6. For the i-th test configuration, the two motion vectors and the corresponding error block are computed. Then the criterion $E(\Gamma_i)$ is computed. At the end of this phase, we find the best initial configuration $j = \arg\min_i E(\Gamma_i)$. Then, this configuration is slightly perturbed in order to find a set of similar segmentations. For each of them the criterion E is computed, using the same vectors $\mathrm{v}(\Gamma_i)$ found at the previous step. When the best segmentation is found, the vectors are recomputed. This is necessary because the new segmentation is different (even though slightly) from the one the motion vectors have been computed for.

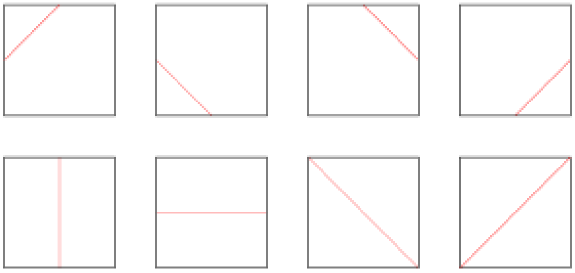


Figure G.6: *The initial split configuration tested by the ME algorithm*

G.3.4 Occlusion control

The motion estimation algorithm uses a technique for detecting the occlusions. If one between the backward and the forward errors is considerably larger than the other, its contribution is not considered in the ME procedure, *i.e.* only the backward or the forward contribution to the error block is considered.

G.4 Results

BM algorithm uses as criterion a bidirectional with SAD constraint in color (3-D) G.10, and uses interpolation in order to achieve a quarter pixel precision. The fast-split and full-split mode were experimented.

Results are shown on sequence "Boston", a synthetic video sequence with a head traveling, split segmentation results are shown on Fig. G.7.

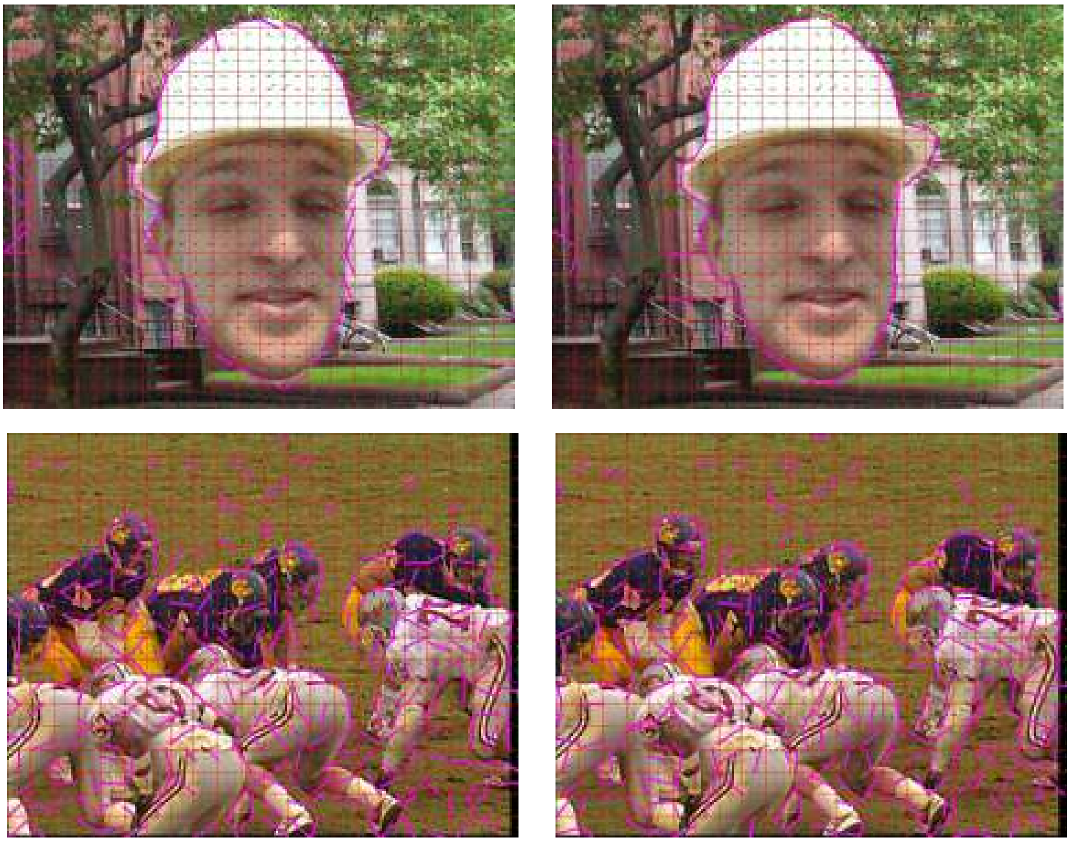


Figure G.7: *Full versus fast method in split mode: split-fast, left column, split-full, right column*

G.5 Conclusion

We have described a joint motion segmentation and motion estimation algorithm. We adopted a simplified approach using MBs in which the problem is solved independently in each MB. We presented interesting first results on video coding. Finally this new mode was implemented in an industrial video coder in competition with existing inter and inter+4v modes. As expected, the split mode was chosen as the optimal one at object boundaries on Fig. G.8.

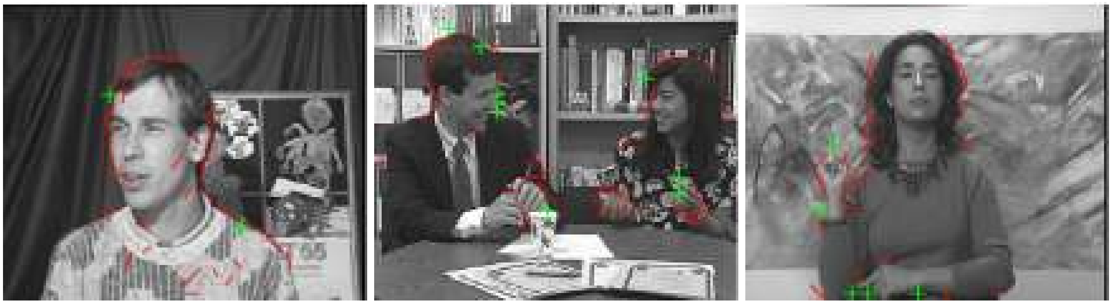


Figure G.8: *Integration in a video coder with modes in competition: inter+4v (green) and split modes (red)*

G.6 Conclusion

In this chapter, we demonstrated that using the kNN framework, one can derive an efficient multivariate framework for active contours based on information theory. This framework shows that even on low dimensional space such as RGB colors, using features jointly is efficient for segmentation purpose. Other features are combined than just color to integrate local statistics or motion cues. This information theoretic framework also allows to define high dimensional joint shape and appearance priors for segmentation. Finally, last chapter presented a simplified motion segmentation algorithm for video coding and was implemented in the video coder of the French national operator “Orange”.

Author Publications

Journal publications

[1] **S. Boltz**, A. Herbulot, E. Debreuve, M. Barlaud, and G. Aubert, "Motion and appearance nonparametric joint entropy for video segmentation," *in International Journal of Computer Vision (in press)*, 2008, IJCV.

[2] **S. Boltz**, E. Debreuve, M. Barlaud, "High dimensional statistical distance for tracking" *in revision at IEEE Transactions on Image Processing*, 2008, TIP.

[3] **S. Boltz**, E. Debreuve, M. Barlaud, "High-dimensional image segmentation" *in preparation for International Journal of Computer Vision*, 2008, IJCV.

[4] **S. Boltz**, E. Debreuve, M. Barlaud, "A unified statistical framework for optical flow" *in preparation for SIAM image science*, 2008, SIIMS.

International conferences with review committee

[5] **S. Boltz**, E. Debreuve, and M. Barlaud, "Joint appearance and shape for nonparametric segmentation," *in Human Motion, LNCS 4814 Workshop of IEEE Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007, ICCV'07.

[6] **S. Boltz**, E. Debreuve, and M. Barlaud, "A high dimensional framework for joint color-spatial segmentation," in *IEEE International Conference on Image Processing*, San Antonio, USA, 2007, ICIP'07.

[7] V. Garcia, **S. Boltz**, E. Debreuve, and M. Barlaud, "Outer-layer based tracking using entropy as a similarity measure," in *IEEE International Conference on Image Processing*, San Antonio, USA, 2007, ICIP'07.

[8] **S. Boltz**, E. Debreuve, and M. Barlaud, "High-dimensional statistical distance for region-of-interest tracking: Application to combining a soft geometric constraint with radiometry," in *IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007, CVPR'07.

[9] **S. Boltz**, E. Debreuve, and M. Barlaud, "knn-based high-dimensional kullback-leibler distance for tracking," in *International Workshop on Image Analysis for Multimedia Interactive Services*, Santorini, Greece, 2007, WIAMIS'07.

[10] A. Herbulot, **S. Boltz**, E. Debreuve, M. Barlaud, and G. Aubert, "Space-time segmentation based on a joint entropy with estimation of nonparametric distributions," in *International Conference on Scale Space Methods and Variational Methods in Computer Vision*, Ischia, Italy, 2007, SSVM'07: joint edition of the 6th Scale Space and the 4th VLSM.

[11] **S. Boltz**, A. Herbulot, E. Debreuve, and M. Barlaud, "Entropy-based space-time segmentation in video sequences," in *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, Graz, Austria, May 2006, SMVP'06 received the best paper award.

[12] V. Garcia, **S. Boltz**, E. Debreuve, and M. Barlaud, "Contour tracking for rotoscoping based on trajectories of feature points," in *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, Graz, Austria, May 2006, SMVP'06.

[13] **S. Boltz**, E. Wolsztynski, E. Debreuve, E. Thierry, M. Barlaud, and L. Pronzato, "A minimum-entropy procedure for robust motion estimation," in *IEEE International Conference on Image Processing*, Atlanta, USA, 2006, pp. 1249–1252, ICIP'06.

[14] A. Herbulot, **S. Boltz**, E. Debreuve, and M. Barlaud, "Robust motion-based segmentation in video sequences using entropy estimator," in *IEEE International Conference on Image Processing*, Atlanta, USA, 2006, pp. 1853–1856, ICIP'06.

[15] **S. Boltz**, E. Debreuve, and M. Barlaud, "A joint motion computation and segmentation algorithm for video coding," in *European Signal Processing Conference*, Antalya, Turkey, september 2005, EUSIPCO'05.

Bibliography

[AAB05a] T. ANDRÉ, M. ANTONINI et M. BARLAUD : Full occlusion management for wavelet-based video coding. *In European Signal Processing Conference*, Antalya, Turkey, september 2005.

[AAB05b] T. ANDRÉ, M. ANTONINI et M. BARLAUD : Puzzle temporal lifting for wavelet-based video coding. *In IEEE International Conference on Image Processing*, volume III, pages 213–216, Genova, Italy, sept 2005.

[ABFJB03] G. AUBERT, M. BARLAUD, O. FAUGERAS et S. JEHAN-BESSON : Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM Applied Mathematics*, 1(2):2128–2145, 2003.

[Abr82] I. ABRAMSON : On bandwidth variation in kernel estimates a square root law. *The Annals of Statistics*, 10:1217–1223, 1982.

[ADPB08] S. ANTHOINE, E. DEBREUVE, P. PIRO et M. BARLAUD : Using neighborhood distributions of wavelet coefficients for on-the-fly, multiscale-based image retrieval. *In Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2008.

[AFPA06] Vincent ARSIGNY, Pierre FILLARD, Xavier PENNEC et Nicholas AYACHE : Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, August 2006.

[AL76] I.A. AHMAD et P.E. LIN : A nonparametric estimation of the entropy for absolutely continuous distributions. *In IEEE Transactions on Information Theory*, volume 22, pages 372–375, 1976.

[AM93] S. ARYA et D. M. MOUNT : Approximate nearest neighbor searching. *In 4th Ann. ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 271–280, 1993.

[ATW06] S.P. AWATE, T. TASDIZEN et R.T. WHITAKER : Unsupervised texture segmentation with nonparametric neighborhood statistics. *In European Conference on Computer Vision*, pages 494–507, Graz, Austria, 2006.

[AW06] S.P. AWATE et R.T. WHITAKER : Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Transactions Pattern Analysis Machine Intelligence*, 28(3):364–376, 2006.

[BA96] M. J. BLACK et P. ANANDAN : The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[BBPW04] T. BROX, A. BRUHN, N. PAPENBERG et J. WEICKERT : High accuracy optical flow estimation based on a theory for warping. *In European Conference on Computer Vision*, pages 25–36, Prague, Czech Republic, 2004.

[BCM05] A. BUADES, B. COLL et J.-M. MOREL : A non-local algorithm for image denoising. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 60–65, San Diego, CA, USA, 2005.

[BD94] A. BERLINET et L. DEVROYE : A comparison of kernel density estimates. *Publications de l'Institut de Statistique de l'Université de Paris*, 38(3):3–59, 1994.

[BDB07] S. BOLTZ, E. DEBREUVE et M. BARLAUD : High-dimensional statistical distance for region-of-interest tracking: Application to combining a soft geometric constraint with radiometry. *In IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007.

[BDB08] S. BOLTZ, E. DEBREUVE et M. BARLAUD : A unified statistical framework for optical flow. *In submitted to European Conference on Computer Vision*, 2008.

[BHD$^+$07] S. BOLTZ, A. HERBULOT, E. DEBREUVE, M. BARLAUD et G. AUBERT : Motion and appearance nonparametric joint entropy for video segmentation. *International Journal of Computer Vision*, 2007. in press.

[BHDB06] S. BOLTZ, A. HERBULOT, E. DEBREUVE et M. BARLAUD : Entropy-based space-time segmentation in video sequences. *In ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, Graz, Austria, May 2006. SMVP'06 *received the best paper award*.

[BMDG05] A. BANERJEE, S. MERUGU, I. DHILLON et J. GHOSH : Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[BMP02] S. BELONGIE, J. MALIK et J. PUZICHA : Shape matching and object recognition using shape contexts. *IEEE Transactions Pattern Analysis Machine Intelligence*, 24(4): 509–522, 2002.

[BP07] A. BUGEAU et P. PÉREZ : Detection and segmentation of moving objects in highly dynamic scenes. *In IEEE International Conference on Computer Vision and Pattern Recognition*, Minneapolis, MI, June 2007.

[BRDW03] T. BROX, M. ROUSSON, R. DERICHE et J. WEICKERT : Unsupervised segmentation incorporating colour, texture, and motion. *In Computer Analysis of Images and Patterns*, volume 2756 de *LNCS*, pages 353–360. Springer, August 2003.

[BRS$^+$07] S. BAKER, S. ROTH, D. SCHARSTEIN, M.J. BLACK, J.P. LEWIS et R. SZELISKI : A database and evaluation methodology for optical flow. *In IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[BWD$^+$06] S. BOLTZ, E. WOLSZTYNSKI, E. DEBREUVE, E. THIERRY, M. BARLAUD et L. PRONZATO : A minimum-entropy procedure for robust motion estimation. *In IEEE International Conference on Image Processing*, pages 1249–1252, Atlanta, USA, 2006. ICIP'06.

[BWS05] A. BRUHN, J. WEICKERT et C. SCHNÖRR : Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[CAAB04] M. CAGNAZZO, T. ANDRÉ, M. ANTONINI et M. BARLAUD : A model-based motion compensated video coder with JPEG2000 compatibility. *In IEEE International Conference on Image Processing*, pages 2255–2258, Singapore, octobre 2004.

[CBFAB97] P. CHARBONNIER, L. BLANC-FÉRAUD, G. AUBERT et M. BARLAUD : Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997.

[CH04] J. COSTA et A. O. HERO : Manifold learning using euclidean k-nearest neighbor graphs. *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 988–991, 2004.

[CKS97] V. CASELLES, R. KIMMEL et G. SAPIRO : Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[CKS03] D. CREMERS, T. KOHLBERGER et C. SCHNÖRR : Shape Statistics in Kernel Space for Variational Image Segmentation. *Pattern Recognition*, 36(9):1929–1943, 2003.

[Com03] Dorin COMANICIU : An algorithm for data-driven bandwidth selection. *IEEE Transactions Pattern Analysis Machine Intelligence*, 25(2):281–288, 2003.

[COS06] D. CREMERS, S. J. OSHER et S. SOATTO : Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, September 2006.

[CRD07] D. CREMERS, M. ROUSSON et R. DERICHE : A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, April 2007.

[CRM00] D. COMANICIU, V. RAMESH et P. MEER : Real-time tracking of non-rigid objects using mean shift. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 142–151, Hilton Head Island, SC, 2000.

[CRM03] Dorin COMANICIU, Visvanathan RAMESH et Peter MEER : Kernel-based object tracking. *IEEE Transactions Pattern Analysis Machine Intelligence*, 25(5):564–575, 2003.

[CS03] D. CREMERS et S. SOATTO : Variational space-time motion segmentation. *In ICCV*, pages 886–893, 2003.

[CS05] D. CREMERS et S. SOATTO : Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.

[CSV03] T. CHAN, J. SHEN et L. VESE : Variational pde models in image processing. *Notice of Amer. Math. Soc.*, 50, 2003.

[CT91] Thomas M. COVER et Joy A. THOMAS : *Elements of information theory*. John Wiley and Sons, Inc., 1991.

[CV01] T. CHAN et L. VESE : Active contours without edges. *In IEEE Transactions on Image Processing*, volume 10, pages 266–277, 2001.

[CZT05] R. COLLINS, X. ZHOU et S.K. TEH : An open source tracking testbed and evaluation web site. *In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, Breckenridge, CO, http://www.vividevaluation.ri.cmu.edu/, 2005.

[DGBA05] E. DEBREUVE, M. GASTAUD, M. BARLAUD et G. AUBERT : A region-based joint motion computation and segmentation on a set of frames. *In WIAMIS*, Montreux, Swi, 2005.

[DL00] L.P. DEVROYE et G. LUGOSI : Variable kernel estimates: on the impossibility of tuning the parameters. *In* D. Mason E. GINÉ et J.A. WELLNER, éditeurs : *High-Dimensional Probability II*, pages 405–424. Springer-Verlag, New York, 2000.

[DPM06] C.C. DOREA, M. PARDÀS et F. MARQUÉS : Generation of long-term color and motion coherent partitions. *In IEEE International Conference on Image Processing*, pages 581–584, 2006.

[DZ01] M.C. DELFOUR et J.P. ZOLÉSIO : *Shape and geometries*. Advances in Design and Control, SIAM, 2001.

[EDD03] A. ELGAMMAL, R. DURAISWAMI et L. S. DAVIS : Probabilistic tracking in joint feature-spatial spaces. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 781–788, Madison, WI, 2003.

[ESK07] P. ETYNGIER, F. SÉGONNE et R. KERIVEN : Shape priors using manifold learning techniques. *In iccv*, 2007.

[FH51] E. FIX et J.L HODGES : Discriminatory analysis, non-parametric discrimination: consistency properties. Rapport technique, USAF School of aviation and medicine, Randolph Field, 1951. 4.

[FH75] K. FUKUNAGA et L.D. HOSTETLER : The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.

[FZ04] D. FREEDMAN et T. ZHANG : Active contours for tracking distributions. *IEEE Transactions on Image Processing*, 13(4):518–526, 2004.

[GBDB06] V. GARCIA, S. BOLTZ, E. DEBREUVE et M. BARLAUD : Contour tracking for rotoscoping based on trajectories of feature points. *In ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, Graz, Austria, May 2006. SMVP'06.

[GDB08] V. GARCIA, E. DEBREUVE et M. BARLAUD : Fast k nearest neighbor search using gpu. *In CVPR Workshop on Computer Vision on GPU*, Anchorage, Alaska, USA, June 2008.

[GLMI05] M. N. GORIA, N. N. LEONENKO, V. V. MERGEL et P. L. Novi INVERARDI : A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *J. Nonparametr. Stat.*, 17(3):277–297, 2005.

[GM95] C. GOURIEROUX et A. MONFORT : *Statistics and Econometric Models*, volume 1. Cambridge University Press, 1995.

[GSM03] Bogdan GEORGESCU, Ilan SHIMSHONI et Peter MEER : Mean shift based clustering in high dimensions: A texture classification example. *In IEEE International Conference on Computer Vision*, page 456, 2003.

[Gt] DPI GTTINGEN : Tstool toolbox for nearest neighbor statistics. http://www.dpi.physik.uni-goettingen.de/tstool/.

[HBDB06] A. HERBULOT, S. BOLTZ, E. DEBREUVE et M. BARLAUD : Robust motion-based segmentation in video sequences using entropy estimator. *In IEEE International Conference on Image Processing*, pages 1853–1856, Atlanta, USA, 2006. ICIP'06.

[HM99] Jinggang HUANG et David MUMFORD : Statistics of natural images and models. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1541–1547, Ft. Collins, CO, USA, 1999.

[HPM06] X. HUANG, N. PARAGIOS et D. METAXAS : Shape registration in implicit spaces using information theory and free form deformations. *IEEE Transactions Pattern Analysis Machine Intelligence*, 28(8):1303–1318, 2006.

[HR04] M. HINTERMULLER et W. RING : A second order shape optimization approach for image segmentation. *SIAM Applied Mathematics*, 64(2):442–467, 2004.

[HS81] B. K. P. HORN et B. G. SCHUNCK : Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HSD$^+$06] A. HERBULOT, S. Jehan-Besson S., DUFFNER, M. BARLAUD et G. AUBERT : Segmentation of vectorial image features using shape gradients and information measures. *Journal of Mathematical Imaging and Vision*, 2006.

[Ihl] A. IHLER : Kernel density estimation toolbox for matlab.

[JBBA03] S. JEHAN-BESSON, M. BARLAUD et G. AUBERT : DREAM^{2}S: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation. *International Journal of Computer Vision*, 53(1):45–70, 2003.

[KFY$^+$05] J. KIM, J. W. F. FISHER, A. YEZZI, M. ÇETIN et A. S. WILLSKY : A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, october 2005.

[KL87] L. KOZACHENKO et N. LEONENKO : On statistical estimation of entropy of random vector. *Problems Infor. Transmiss.*, 23(2):95–101, 1987.

[Kul59] S. KULLBACK : *Information theory and statistics*. John Wiley and Sons., New York, 1959.

[KWT87] M. KASS, A. WITKIN et D. TERZOPOULOS : Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[LFGWI00] M. E. Leventon, O. Faugeras, W. E. L. Grimson et W. M. Wells III : Level set based segmentation with intensity and curvature priors. *In IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis*, 2000.

[LGF00] M. Leventon, E. Grimson et O. Faugeras : Statistical shape influence in geodesic active contour. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1316–1323, Hilton Head Island, South Carolina, 2000.

[Lin91] J. Lin : Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145 – 151, 1991.

[LK81] B.D. Lucas et T. Kanade : An iterative image registration technique with an application to stereo vision. *In IJCAI81*, pages 674–679, 1981.

[Low04] D.G. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LPS05] N. Leonenko, L. Pronzato et V. Savani : A class of Renyi information estimators for multidimensional densities. PASCAL archive: http://eprints.pascal-network.org/archive/00001031/, 2005.

[LQ65] D. Loftsgaarden et C. Quesenberry : A nonparametric estimate of a multivariate density function. *Annals Math. Statistics*, 36:1049–1051, 1965.

[MAT07] B. Mory, R. Ardon et J.P. Thiran : Variational segmentation using fuzzy region competition and local non-parametric probability density functions. *In IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[MFTM01] D. Martin, C. Fowlkes, D. Tal et J. Malik : A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *In IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[Min05] Tom Minka : Divergence measures and message passing. *In Microsoft Research Technical Report (MSR-TR-2005-173)*, 2005.

[MP04] A. Mittal et N. Paragios : Motion-based background subtraction using adaptive kernel density estimation. *In IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 302–309, 2004.

[MS89] D. Mumford et J. Shah : Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989.

[NBN07] F. Nielsen, J.-D. Boissonnat et R. Nock : On bregman voronoi diagrams. *In ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[NE86] H. H. Nagel et W. Enkelmann : An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions Pattern Analysis Machine Intelligence*, 8:565 – 593, 1986.

[OB95] J.-M. Odobez et P. Bouthemy : Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6:348–365, 1995.

[OS88] S. OSHER et J.A. SETHIAN : Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[PADB08] P. PIRO, S. ANTHOINE, E. DEBREUVE et M. BARLAUD : Image retrieval via kullback-leibler divergence of patches of multiscale coefficients in the knn framework. *In International Workshop on Content-Based Multimedia Indexing (CBMI), London, UK, 18-20th June*, 2008.

[Par62] E. PARZEN : On the estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[PBBU05] F. PRECIOSO, M. BARLAUD, T. BLU et M. UNSER : Robust real-time segmentation of images and videos uisng a smoothing-spline snake-based algorithm. *IEEE Transactions on Image Processing*, 14(7):910–924, 2005.

[PD02a] N. PARAGIOS et R. DERICHE : Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, pages 249–268, 2002.

[PD02b] N. PARAGIOS et R. DERICHE : Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, 46(3):223–247, February 2002.

[PHVG02] P. PÉREZ, C. HUE, J. VERMAAK et M. GANGNET : Color-based probabilistic tracking. *In European Conference on Computer Vision*, volume 2350 de *LNCS*, pages 661–675, Copenhaguen, Denmark, 2002.

[PLS07] L. PRONZATO, N. LEONENKO et V. SAVANI : A class of renyi information estimators for multidimensional densities. *Annals of Statistics*, in press, 2007.

[RB05] S. ROTH et M.J. BLACK : On the spatial statistics of optical flow. *In IEEE International Conference on Computer Vision*, pages 42–49, Nice, France, 2005.

[RBD03] M. ROUSSON, T. BROX et R. DERICHE : Active unsupervised texture segmentation on a diffusion based feature space. *In IEEE International Conference on Computer Vision and Pattern Recognition*, june 2003.

[Ros56] M ROSENBLATT : Remarks on some nonparametric estimates of a density function. *Annals of Math. Statistics*, 27:832–837, 1956.

[RP02] M. ROUSSON et N. PARAGIOS : Shape priors for level set representations. *In European Conference on Computer Vision*, pages 78–92, 2002.

[Sai02] Stephan R. SAIN : Multivariate locally adaptive density estimation. *Comput. Stat. Data Anal.*, 39(2):165–186, 2002.

[SB97] S.M. SMITH et J.M. BRADY : Susan a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[SBFGJ00] C. SAMSON, L. BLANC-FÉRAUD, G.AUBERT et J.ZERUBIA. : A level set model for image classification. *International Journal of Computer Vision*, 3:187–197, 2000.

[Sco92] D.W. SCOTT : *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

[Sil86] B.W. SILVERMAN : *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[SM06] E. SHARON et D. MUMFORD : 2d-shape analysis using conformal mapping. *International Journal of Computer Vision*, 70(1):55–75, 2006.

[ST94] J. SHI et C. TOMASI : Good features to track. *In IEEE International Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[TPJ05] M. TARON, N. PARAGIOS et M.-P. JOLLY : Modelling shapes with uncertainties : Higher order polynomials, variable bandwidth kernels and non-parametric density estimation. *In IEEE International Conference on Computer Vision*, Beijing, Oct 2005.

[TS92] G. R. TERRELL et D. W. SCOTT : Variable kernel density estimation. *The Annals of Statistics*, 20:1236–1265, 1992.

[VBPB07] R. VENKATESH BABU, P. PÉREZ et P. BOUTHEMY : Robust tracking with motion estimation and local kernel-based color modeling. *Image Vis. Comput. In Press*, 2007.

[VW97] P. VIOLA et W. M. WELLS : Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[WK93] S. F. WU et J. KITTLER : A gradient-based method for general motion estimation and segmentation. *Journal of Visual Communication and Image Representation*, 4:25–38, 1993.

[WS01] J. WEICKERT et C. SCHNÖRR : Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3): 245–255, 2001.

[WTP05] E. WOLSZTYNSKI, E. THIERRY et L. PRONZATO : Minimum entropy estimation in semi parametric models. *Signal Processing*, 85:937–949, 2005.

[Yar85] Leonid P. YAROSLAVSKY : *Digital Picture Processing*. Springer-Verlag New York, Inc., 1985.

[YDGD03] C. YANG, R. DURAISWAMI, N. A. GUMEROV et L. DAVIS : Improved fast gauss transform and efficient kernel density estimation. *In IEEE International Conference on Computer Vision*, Nice, France, 2003.

[YJS06] A. YILMAZ, O. JAVED et M. SHAH : Object tracking: A survey. *ACM Computing Surveys*, 38(4):13, 2006.

[YTW02] A. YEZZI, A. TSAI et A. WILLSKY : A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13:195–216, March 2002.

[ZM00] S. ZHU et K.-K. MA : A new diamond search algorithm for fast block-matching motion estimation. *IEEE Transactions on Image Processing*, 9(2):287–290, 2000.

[ZY96] S. ZHU et A. YUILLE : Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions Pattern Analysis Machine Intelligence*, 18:884–900, septembre 1996.

Résumé: Cette thèse aborde le traitement d'images et de vidéos sous l'angle variationnel, c'est-à-dire sous forme d'une énergie dont le minimum est atteint pour la solution. La modélisation adoptée pour formaliser le problème et obtenir ces énergies peut être déterministe ou stochastique. Il est connu que la première approche est équivalente à la classe paramétrique de la seconde. Ce constat nous a amenés à faire le choix de la seconde approche a priori plus générale si l'on se débarrasse de l'hypothèse paramétrique. En contrepartie, il s'agit d'tre capable d'exprimer et d'estimer une énergie en fonction des données alors interprétées comme des échantillons d'une variable aléatoire. Ce premier obstacle est classiquement surmonté par l'emploi de méthodes à noyau fixe sur des lois marginales, autrement dit, en supposant les canaux de données indépendants entre eux. Or cet obstacle en cache deux autres : l'inhomogénéité de la répartition des échantillons dans leur espace d'appartenance et leur faible densité dans cet espace. Ces difficultés, ainsi que l'hypothèse d'indépendance mentionnée plus haut, sont d'autant plus pénalisantes que le modèle proposé pour les données est de grande dimension (canaux couleur, mais aussi ajouts d'autres canaux pour prendre en compte les structures locales des images). Au fondement d'estimateurs de mesures statistiques telle que l'entropie, l'idée du kième plus proche voisin permet de résoudre les difficultés évoquées en s'adaptant à la densité locale des données, en considérant les canaux conjointement, et ce quel que soit leur nombre. Dans ce contexte, nous proposons une approche statistique générale inspirée de la théorie de l'information, dédiée aux approches variationnelles car estimant efficacement des énergies en haute dimension, permettant le calcul de leur dérivée et pouvant estimer localement des probabilités. Ce cadre est appliqué aux trois problèmes de traitement d'images ou de vidéos que sont l'estimation de flot optique, le suivi d'objets vidéos et la segmentation. Ce point de vue, en permettant de limiter sinon de s'affranchir du problème de la dimension des données, autorise la définition de nouvelles mesures et lois de probabilités plus adaptées aux images naturelles. Certains travaux en suivi d'objets et en segmentation ont conduit à des implémentations industrielles.

Abstract: This thesis addresses variational formulation of image and video processing problems. This formulation expresses the solution through a minimization of an energy. These energies can be expressed as deterministic or stochastic. The first approach corresponding to the parametric class of the second one. The second class is then more general if we get rid of the parametric assumption. In return, the energy must be expressed as a function of the data considered as random variables. These functions are classically estimated with fixed-sized kernels on marginal distributions of the data, assuming the different channels are independent. These methods have two limitations, the inegal repartition and sparsity of the data in the space. These difficulties, as well as the independence assumption are enhanced when the data of the image are high dimensional (color channels, or other channels describing local patterns of natural images). At the foundation of statistics, the k-th nearest neighbor can solve these difficulties by locally adapting to the repartition of the data and treating the channels jointly. We propose a general statistical framework based on statistics and information theory. This new framework is dedicated to variational problems as it efficiently estimates, high dimensional energies, gradients of these energies and local probabilities. This framework is applied to three problems of image and video processing: optical flow, object tracking and segmentation. This framework circumvents the problem of dimensionality and allows us to introduce new measures and probabilities more adapted to natural images. Some results obtained have been applied in an industrial context.