Statistical similarity measures and k nearest neighbor estimators teamed up to handle high-dimensional descriptors in image and video processing

Eric Debreuve

July 8, 2009

Statistical similarity measures and k nearest neighbor estimators teamed up to handle high-dimensional descriptors in image and video processing

Accreditation to Supervise Research
"Habilitation à diriger des recherches"
Université de Nice-Sophia Antipolis, France

Eric Debreuve, CNRS CR1

Jury composition

Vicent **Caselles**,	Professor	· Universitat Pompeu Fabra, Barcelona	Reviewer
Jean-Michel **Morel**,	Professor	· Ecole Normale Supérieure de Cachan	Reviewer
Patrick **Pérez**,	Research director	· Inria, Rennes	Reviewer
Gilles **Aubert**,	Professor	· Université de Nice-Sophia Antipolis	President
Michel **Barlaud**,	Professor	· Université de Nice-Sophia Antipolis	Examiner
Giovanni **Poggi**,	Professor	· Università Federico II, Napoli	Examiner

8 juillet 2009

Mesures de similarité statistiques et estimateurs par k plus proches voisins : une association pour gérer des descripteurs de haute dimension en traitement d'images et de vidéos

Habilitation à diriger des recherches
Université de Nice-Sophia Antipolis, France

Eric Debreuve, CNRS CR1

Composition du jury

Vicent **Caselles**,	Professeur	· Universitat Pompeu Fabra, Barcelona	Rapporteur
Jean-Michel **Morel**,	Professeur	· Ecole Normale Supérieure de Cachan	Rapporteur
Patrick **Pérez**,	Directeur de recherche	· Inria, Rennes	Rapporteur
Gilles **Aubert**,	Professeur	· Université de Nice-Sophia Antipolis	Président
Michel **Barlaud**,	Professeur	· Université de Nice-Sophia Antipolis	Examinateur
Giovanni **Poggi**,	Professeur	· Università Federico II, Napoli	Examinateur

Cast

Vincenzo **Angelino**
Sandrine **Anthoine**
Gilles **Aubert**
Sílvia **Barja**
Michel **Barlaud**
Sylvain **Boltz**
Pierre **Comon**
Vincent **Garcia**
Muriel **Gastaud**
Ariane **Herbulot**
Jean-Paul **Marmorat**
Paolo **Piro**
Luc **Pronzato**
Anonymous **Reviewer of [Deb+07]**
Tristan **Roy**
Eric **Wolsztynski**

Special Thanks To

The Reviewers
The Jury

This document presents, at different levels of detail, some developments that were made jointly with students as part of either an official supervision, an unofficial supervision, or an informal collaboration. The co-supervisions were shared with Professor Barlaud.

	Position at the time of collaboration	*Position subsequently held or current position*	*Responsibility / context*
Vincenzo **Angelino**	PhD thesis	PhD thesis	Official co-supervision
Sílvia **Barja**	Last year project, UPC, Barcelona (Erasmus – 7 months)	Engineer at Telefónica, Spain	Supervision
Sylvain **Boltz**	PhD thesis	Post-Doctoral position, Computer Science department, UCLA, with Prof. Stefano Soatto	Unofficial co-supervision
Vincent **Garcia**	PhD thesis	Post-Doctoral position, LIX, Ecole Polytechnique, with Prof. Frank Nielsen	Official co-supervision
Muriel **Gastaud**	PhD thesis	Post-Doctoral position, ENST, Paris, with Prof. Henri Maître	Unofficial co-supervision
Ariane **Herbulot**	PhD thesis	Research/Teaching position, LAAS/Université de Toulouse III	Collaboration
Paolo **Piro**	PhD thesis	PhD thesis	Collaboration
Tristan **Roy**	Master project, Master MVA, ENS de Cachan	PhD thesis in mathematics, UCLA	Co-supervision

Table of Content

INTRODUCTION

Chapter 1

Context and preliminary remarks

Context

The purpose of this document is to describe a variational framework to express and solve various image and video processing tasks handling features efficiently and in the same way whether they are low or high-dimensional. This introductory chapter presents some motivations and hints that will be developed in the subsequent parts.

1.1 Low to midlevel image and video processing tasks

This document deals with image and video processing tasks that may represent building bricks for content analysis or understanding. Mainly three problems will be studied: segmentation, denoising, and tracking. Image inpainting, optical flow computation, and content-based indexing and retrieval (which might be considered a midlevel task since it attempts to answer the fuzzy question "Do these images look alike?") will be briefly mentioned.

1.2 Similarity

1.2.1 A central notion

Many image and video processing problems can be solved by optimizing some cost functions. The notion of similarity is often behind these functions. It can be a "self-similarity" when a coherence is searched for within an object, or a "cross-similarity" between two objects, images, or videos. Image restoration and segmentation typically call upon self-similarity and content-based indexing and retrieval depend on the definition of a cross-similarity. Intermediately, tasks performed on video such as restoration, segmentation, tracking, and optical flow computation rely upon a similarity of an objet or a scene view with itself as observed on another

frame. This notion can be decomposed into two components: a description and a comparison function.

1.2.2 Description

An object, image, or video description can result from a modeling of the given item class. The difficulty is to select a few parameters to represent a class which, actually, can contain a wide variety of elements. Instead, a description can be formed by a set of examples or samples of the item class. The potential limitations are whether the samples are really representative of the class and whether the set is large enough to be statistically significant. In both cases, model or samples, it is usually more realistic to restrict the field of application of a task.

1.2.3 Comparison function

Note that the frontier between the descriptive aspect and the comparison aspect might be subjective. For example, if similarity between two grayscale images is obtained by the Battacharya coefficient of their respective histogram, the image description can be its histogram and the comparison function the coefficient expression. The description can also be the set of image pixels and the comparison function the combination of histogram estimation and coefficient computation.

1.2.4 Invariance

Invariance is a crucial property [Kad&Bra01, Kad+04, Mik&Sch05, Tuy&Mik07]. It is the equivalent of the generalization property of a classifier in the context of similarity. This property is often attached to the description. A well-known example is the Scale Invariant Feature Transform (SIFT) [Low04, Mor&Yu09]. However, the comparison function plays also a role in terms of invariance. Indeed, if two images A and B are described by their ordered set of pixels, their descriptions are not invariant to any alteration. Comparing them using a sum of squared differences (SSD), no invariance is introduced. On the contrary, the entropy of the difference image $A - B$ introduces invariance to many transformations since, due to the absence of geometrical constraint, many difference images have the same entropy.

To avoid the constraint of a fixed descriptive model, we will focus on example-based descriptions. In this context, comparison functions involving statistical or information measures appear as a coherent choice since it allows each example to be seen as a realization of a random variable. As described above, a similarity measure refers to a comparison function evaluated for some object, image, or video descriptions. For simplicity, the expression “similarity measure” will also be used to denote the comparison function.

1.3 PDF-based similarity measures

Similarity = $\mathcal{F}$(observations). Let x and y be two discretized images or videos defined in a domain $\mathcal{D}$. Because there is rarely an exact model for a given problem, a solution $\hat{x}$ is usually expressed as the minimizer of a function, or functional, $\mathcal{E}$ of the observed data y and a candidate x

$$\hat{x} = \arg\min_x \mathcal{E}(x, y), \tag{1.1}$$

the minimization aspect accounting for the mismatch between the model and reality. Often, the similarity $\mathcal{E}$ involves x and y in the form of the norm of a residual: $|y - \mathcal{M}x|$, where $\mathcal{M}$ is a transformation representing the model. A classical example is the least square solution

$$\hat{x} = \arg\min_x \ |y - \mathcal{M}x|^2 \ . \tag{1.2}$$

Taking a statistical point of view, one might want to find the image which maximizes the likelihood of the observed data

$$\hat{x} = \arg\max_x f(y|x) \ . \tag{1.3}$$

If the model mismatch is entirely attributed to some additive noise b, then

$$y = \mathcal{M}x + b \ . \tag{1.4}$$

Let b be a Gaussian white noise of mean zero and variance σ^2 independent of x. Therefore, given the model (1.4),

$$g(b) = \alpha \ \exp -\frac{|y - \mathcal{M}x|^2}{2\sigma^2} \tag{1.5}$$

where α is a normalization constant. Since b and x are independent, $g(b)$ is equal to $g(b|x)$, which is itself equal to $g(y - \mathcal{M}x|x) = g(y|x) := f(y|x)$. As a consequence, solving (1.3) is equivalent to maximizing (1.5) with respect to x, whose solution is clearly equal to (1.2). In conclusion, under some assumptions, the maximum likelihood solution is identical to the least square solution. Similarly, one can deduce that the maximum a posteriori solution corresponds to a regularized least square solution.

Similarity = $\mathcal{F}$(PDF of the observations). Without any assumption on f, the maximum likelihood solution (1.3) can be rewritten as follows

$$\hat{x} = \arg\min_x -\frac{1}{N} \log f(y - \mathcal{M}x|x) \tag{1.6}$$

where N is the measure $|\mathcal{D}|$ of $\mathcal{D}$. Assuming that the components of $r := y - \mathcal{M}x$ are independent, then

$$f(r|x) = \prod_{s \in \mathcal{D}} f(r_s|x) \tag{1.7}$$

and

$$\hat{x} = \arg\min_x -\frac{1}{N} \sum_{s \in \mathcal{D}} \log f(r_s|x) \tag{1.8}$$

which, if $f(r_s|x)$ is estimated using the Parzen method, is the Ahmad-Lin approximation of the differential entropy [Ahm&Lin76] of r conditional on x. Thus, it appears that entropy can be seen as a generalization of some classical energies such as the least squares. Since the latter is commonly encountered in image and video processing, it seems that entropy and entropy-based measures can be useful as well. Actually, it is already known that methods derived from measures of information theory can be efficient for, *e.g.*, restoration [Awa&Whi06, Ang+08b, Ang+08a], segmentation [Kim+05, Una+05, Bol+08], registration [Vio&Wel97, Plu+03, Cra+08], and tracking [Elg+03, Fre&Zha04, Bol+09].

1.4 Dealing with high-dimensional features

1.4.1 Features

A feature can be defined as a vector describing an image or a video locally around a given position and scale, *e.g.*, the pixel color, the ordered set of colors within a patch [Lee+03, Car+08], local color histograms [Kad&Bra01, Kad+04], a SIFT-based descriptor [Low04, Mor&Yu09], etc [Mik&Sch05, Tuy&Mik07]. Features combined together form a description.

1.4.2 Order of magnitude

Unless otherwise noted, the developments presented in this document apply to feature spaces of arbitrary dimension. In practice, though, the dimension of the features results from the acquisition process (grayscale, color...) and the transformation/rearrangement of the observations (patches...) for the purpose of a specific task. For example, in tracking, the features are of dimension 5 to 13; in denoising, the feature dimension can exceed 50. Whether these dimensions can be considered as high is, probably, mostly a matter of context. First, it is relative to the number of samples available. In tracking, this number can be rather small since it is given by the size of the (user-selected) region-of-interest (ROI). Second, it depends on the purpose the samples are to be used for. When it comes

to estimate statistical measures (entropies, divergences...), the denomination of high-dimensional features makes sense since *classical* approaches sometimes already show their limits when the dimension gets higher than 2 or 3 [Ter&Sco92].

1.5 Organization of this document

This document is composed of the present introduction, three main parts, a short conclusion, and six appendices. Lists of figures, tables, and bibliographical references are also provided at the end. Part I deals with the measures of information theory, namely entropy and the Kullback-Leibler divergence, that will be used in the following. Part II presents these measures in the k nearest neighbor framework. The introduction and these first two parts aim at describing the general approach proposed to study some image and video processing problems. Then, part III details three applications in this framework: segmentation using the shape derivative, nonlocal denoising using conditional entropy, and ROI tracking using the Kullback-Leibler divergence. Finally, inpainting, optical flow computation, and content-based indexing and retrieval are briefly mentioned as other tasks that can be investigated with the proposed point of view. Note that bibliographical references appear as follows:

- [AutYY]: A work published in 19YY or 20YY by a single author;
- [Au1&Au2YY]: A work published by two authors;
- [Aut+YY]: A work published by three authors or more.

Part I

A FRAMEWORK BASED ON INFORMATION THEORY MEASURES

Chapter 2

Entropy: a hypothesis-free data consistency and regularization functional

Context

Many problems of image and video processing can be expressed as the minimization of a data consistency residual and a term of mismatch with respect to a priori constraints. Traditionally, these functionals are based on penalization functions such as the ones defined for robust estimation, sometimes referred to as φ-functions. From a statistical point of view, recurring to these functions is equivalent to implicitly making assumptions on the probability density functions (PDFs) of the residual and the model mismatch, *e.g.*, Gaussian, Laplacian, or other parametric laws – for the square function, the absolute value, or other φ-functions, respectively. Alternatively, it is interesting to adapt to (an estimation of) the true PDF. This nonparametric approach implies to define functionals which take PDFs as input.

Entropy has been proposed in this context since, as a measure of dispersion of a PDF, its minimization leads the residual or model mismatch values to concentrate around narrow modes, the highest one normally corresponding to the annihilation of the residual or mismatch, the others corresponding to inevitable outliers.

2.1 Classical variational approach

2.1.1 Implicit assumption on the PDF

The solution to image and video processing problems can often be formulated as follows

$$\hat{x} = \arg\min_{x} \varphi_d(y - \mathcal{M}x) + \lambda\varphi_r(\nabla x) \tag{2.1}$$

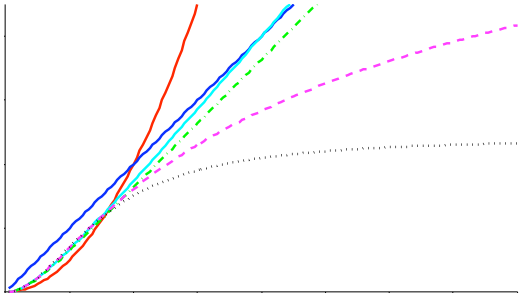

Figure 2.1 – Some functions proposed in robust estimation.
Red • Tikhonov: u^2, Blue • $\mathcal{L}^1$: $|u|$, Cyan • Differentiable approximation of $\mathcal{L}^1$: $\sqrt{u^2+\epsilon^2}-\epsilon$, Dashed green • Green: $\log(\cosh(u))$, Dashed magenta • Hebert & Leahy: $\log(1+u^2)$, **Dotted black** • Geman & McClure: $u^2/(1+u^2)$.

where φ_d and φ_r correspond to data consistency and regularization, respectively, and λ is the regularization parameter. Let φ_d be the $\mathcal{L}^p$-norm operator. Under some hypotheses (see Section 1.3), one can note that (2.1) corresponds to the Maximum A Posteriori solution if the noise follows a generalized Gaussian law[#1] of shape parameter p and the a priori on the solution is given by a Gibbs probability density function (PDF). These laws being defined by a small set of parameters, (2.1) can only adapt to the data to a limited extent. Moreover, such parametric assumptions may not be flexible enough to efficiently deal with outliers.

2.1.2 Dealing with outliers

Let φ refer to either φ_d or φ_r. As far as optimization is concerned, the simplest choice is $\varphi(u) = u^2$. Of course, this is known to inefficiently deal with outliers. Alternatively, one can pick φ among the set of functions proposed in robust estimations – see Fig. 2.1. Even though these functions reduce the bias introduced by outliers, they are sensitive to the values of the outliers nonetheless. Moreover, they still represent an implicit assumption on the underlying distribution of the data – see Section 2.1.1. As an illustration, let A be an image defined on a domain $\mathcal{D}$. Let $\hat{A}$ be an estimation of A by some procedure. Figure 2.2 shows the different norms $\int_{\mathcal{D}} \varphi(A-\hat{A})$ and the entropy of $A-\hat{A}$ in several situations involving outliers. (The $\mathcal{L}^2$ norm has not been plotted since it is known that it performs poorly in the presence of outliers.) It appears that the entropy is quite insensitive to the mean value of the outliers – see Fig. 2.2.a. Indeed, if the PDF of the residual has a main peak around zero and a peak corresponding to outliers, without intersection between their support, then the entropy does not depend on the outlier peak position. The entropy also behaves very well when the proportion of outliers increases (see Fig. 2.2.b), *focusing* on the main mode. Among the robust functions reminded here,

[#1] Since at convergence, the residual $y - \mathcal{M}x$ is ideally equal to the noise, any assumption on the noise also applies to the residual.

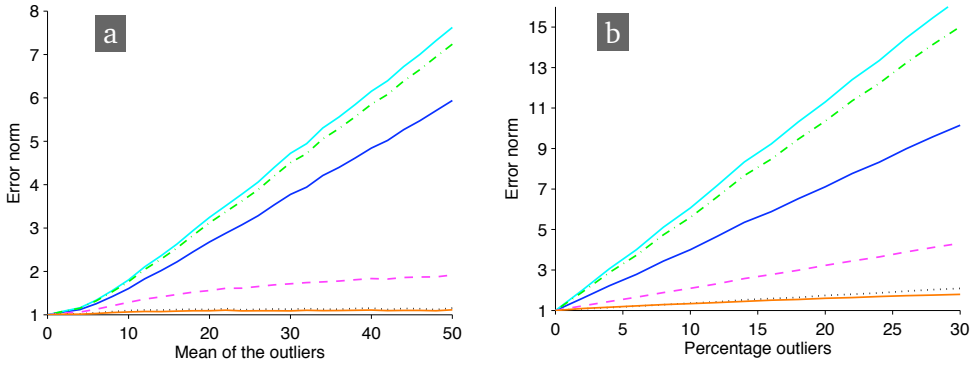

Figure 2.2 – Effect of outliers (u is the residual $A - \hat{A}$).
Blue • $\mathcal{L}^1$: $\int_{\mathcal{D}} |u|$, Cyan • Differentiable approximation of $\mathcal{L}^1$: $\int_{\mathcal{D}} \sqrt{(u^2 + \epsilon^2)} - \epsilon$, Dashed green • Green: $\int_{\mathcal{D}} \log(\cosh(u))$, Dashed magenta • Hebert & Leahy: $\int_{\mathcal{D}} \log(1 + u^2)$, **Dotted black** • Geman & McClure: $\int_{\mathcal{D}} u^2/(1 + u^2)$, Orange • Entropy of u
[a] The residual is composed of 15% of outliers following a Gaussian law of variance 36 and a mean (displayed on the horizontal axis) growing from zero to 50. The remaining 85% of the residual was drawn from a Gaussian law of mean zero and variance 1.
[b] The setup is similar to [a] except that the mean is fixed equal to 25 and the proportion of outliers (displayed on the horizontal axis) grows from 0 to 30.
The entropies and each of the norms have been divided by their respective values when the mean is equal to zero for [a], and when the proportion of outliers is equal to zero for [b]. The errors are therefore relative.

the function proposed by Geman & McClure performs nearly as well as the entropy on these examples. This might be explained by the fact that it has an asymptote (at one). Yet, as for the other functions, it has a parameter (not mentioned so far) to tune the transition between residual values small enough to be considered equal to zero and values large enough to be regarded as outliers – See Fig. 2.3.

It is notoriously difficult to truly cope with outliers. However, entropy seems to be a good replacement for robust functions, first because it is apparently not too sensitive to outliers and second because it is parameter-free.[#2]

2.2 Entropy as a hypothesis-free functional

2.2.1 Working with actual PDFs

The first advantage of working with entropy and other related statistical measures is to deal with outliers in terms of frequency of occurrence as opposed to value. As already mentioned, this eliminates the need for a threshold to distinguish between normal values and outliers. In addition, if the PDF(s) is/are estimated nonparametrically, then the measure makes no assumption on the data or, otherwise

[#2] Actually, no estimator is really parameter-free. Yet entropy does not involve any hard or soft thresholding in dealing with outliers. In this sense, it differs from classical solutions.

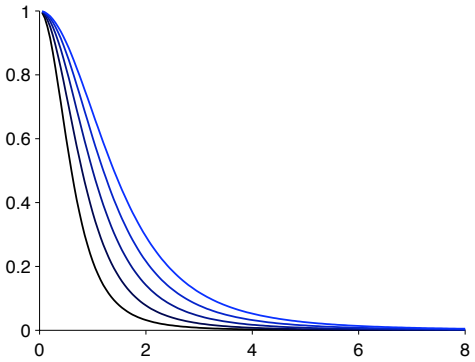

Figure 2.3 – Function $\varphi'_{\mathrm{G\&McC}}(u/\delta)/(2u/\delta)$ which characterizes the transitional behavior of the Geman & McClure robust function between normal (*i.e.*, close to zero) values of its argument u/δ (function close to one) and argument values considered as outliers (function close to zero). The region of transition is tuned by δ. The shades of blue of the curves reflect the value of δ (ranging from **0.6** to 1.4 times a reference value). The value of this parameter has a significant influence on the soft thresholding.

stated, adapts to them. This flexibility should limit the estimation bias compared to approaches resorting to parametric assumptions. More generally, the idea is to study the potential of another class of functionals which take a PDF or several PDFs as input, providing a unifying point of view for data consistency and regularization based on information theory.

2.2.2 Usefulness of image entropy

Entropy is a measure of dispersion of a random variable. The differential entropy $H(f_U)$, or equivalently $H(U)$, of a continuous random variable U of R^d with PDF f_U writes

$$H(U) = -\int_{R^d} f_U(t)\ \log f_U(t)\mathrm{d}t. \tag{2.2}$$

In practice, entropy can be used as a piecewise constant constraint. As an illustration, let A be an image composed of m regions of n distinct average gray levels, $m \geq n$. Let the pixels corresponding to a given gray level be drawn from a normal law of the appropriate mean and a variance equal to σ^2. Taking for example $n = 3$, Fig. 2.4 plots the average entropy of realizations of A as a function of σ^2. The entropy decreasing when the variance gets smaller, one can infer that minimizing the entropy of A will lead to a piecewise constant image. This is clear if $n = 1$ since the entropy is then equal to $\log(\sigma\ \sqrt{(2\pi\mathrm{e})})$, which tends toward $-\infty$ when σ tends toward zero, *i.e.*, when the image becomes constant. Therefore, entropy can be used as a regularization function to enforce a well-known constraint normally

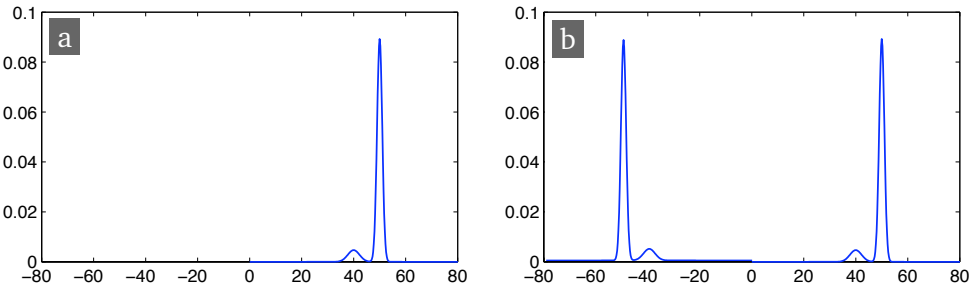

Figure 2.5 – The residual PDF [a] has a low entropy because of its small dispersion. To ensure that the entropy is minimal only when, in addition to the small dispersion, the residual is close to zero, the entropy can be computed on the symmetrized version of the PDF. Clearly, the symmetrized PDF [b] does not satisfy a minimal entropy constraint anymore.

related to total variation. However, as opposed to this latter one, there is no issue concerning the differentiation of entropy.[#3]

Regarding data consistency, entropy can also play the role of model mismatch as is. Indeed, the model mismatch, or residual, should be equal to zero expect for some outliers. This can efficiently be described as a piecewise constant image. Naturally, a minimal entropy constraint cannot guarantee that one of the peaks (the main one) is centered on zero.[#4] Yet, with the action of regularization, if the entropy is minimal, *i.e.*, the residual is piecewise constant, then the main peak is certainly centered on zero. Otherwise, the initialization was probably chosen too far away from the solution – in an iterative resolution process. Potential solutions classically include choosing a better initialization, increasing the weight of the regularization term, and combining both. Now, if one really needs to enforce that the entropy be minimal only when the main peak is narrow *and centered on zero*, the entropy can be computed on the symmetrized version of the residual PDF – see Fig. 2.5.

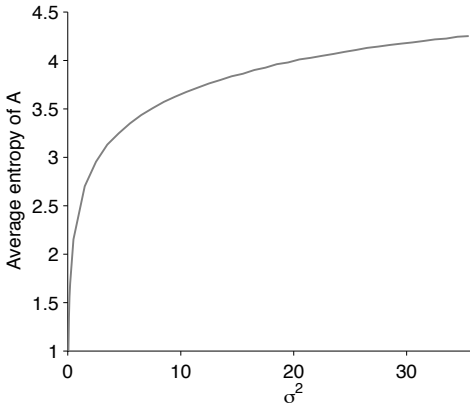

Figure 2.4 – Average entropy of realizations of A as a function of σ^2. The entropy decreases when the variance gets smaller.

Whether for regularization or for data consistency, the piecewise constancy as imposed by entropy might not be usual. Indeed, since entropy accounts for

[#3] In fact, it is more accurate to say that there is a simple answer to the differentiation issue of entropy – see Section 2.2.4.

[#4] Note that a residual equal to zero does not imply that the estimated solution is satisfying anyway – see Section 10.2.2.

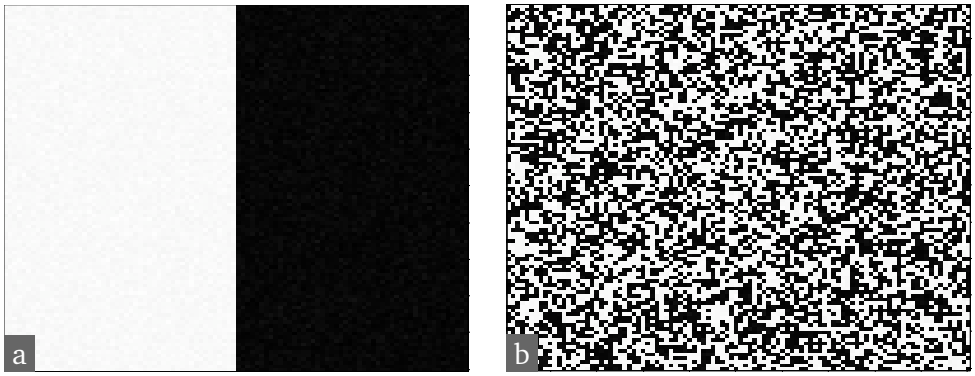

Figure 2.6 – Two images with equal entropy ([b] was obtained by scrambling the pixels of [a]). As is clear from these images, piecewise constant and minimal (pixel-level) entropy are not synonymous.

Feature ↓ \ Fig. →	2.6.[a]	2.6.[b]
Pixel	**2.1**	**2.1**
Pixel+position	**7.9**	8.6
Patch 3×3	**13.7**	20.1
Patch 5×5	**38.2**	134.6

Table 2.1 – Entropies of Figs. 2.6.[a] and 2.6.[b] for several feature spaces. Although at a pixel level the undesirable image has the same entropy as a potential piecewise constant solution, its entropy becomes higher for feature spaces that involve contextual information. With these feature spaces, a minimal entropy constraint seems more coherent with piecewise constancy. *Note that these results were computed using the k nearest neighbor (kNN) entropy estimator presented later in Section 5.2.1. The entropy of some random variables taken jointly being smaller than or equal to the sum of the entropies of the variables taken individually, the "Patch 3×3" entropies should be less than or equal to $(3 \times 3) \times 2.1 = 18.9$ and the "Patch 5×5" entropies should be less than or equal to $(5 \times 5) \times 2.1 = 52.5$. The results of Fig. 2.6.[b] show that the entropy was over-estimated.*

frequency of occurrence independently of position, one can be surprised by what an image of minimal entropy looks like – see Fig. 2.6. A workaround consists in defining the feature space U as a random vector by enriching the random variable of pixel color with contextual information. For example, the pixel coordinates ($U = (c, x, y)$, with c being gray level or color) or the color of neighboring pixels can be added ($U = (c, c_1, c_2 \dots c_n)$, where c_i is the color of the i-th pixel in a chosen neighborhood mask, this set composed of c and its neighborhood being referred to as a patch) – see Tab. 2.1. Of course, when using coordinates as in (c, x, y), U becomes a mix of random and deterministic information. Applying the present statistical framework to such a feature space is not objectively justifiable. For the sake of curiosity, Section 3.2 proposes a brief analysis related to the consequence of doing so. Furthermore, in practice, it proved to be efficient, particularly for tracking – see Chapter 9.

2.2.3 Is PDF estimation necessary?

Apparently when looking at (2.2), the computation of entropy requires a PDF estimation. Section 4.1 reminds several kernel-based methods. The Parzen method lacks local adaptability and the bandwidth estimation is problematic. Conversely, the sample point method adapts to the local sample density. Nevertheless, the k nearest neighbor (kNN) framework (see Section 2.2.4) has the advantage of allowing to derive PDF-based statistical estimators such as entropy and the Kullback-Leibler divergence that do not explicitly involve PDFs. Although kNN PDF estimations are usually quite noisy and considered better than Parzen and sample point at high dimensions only [Ter&Sco92], some derived estimators, in particular entropy, have good properties even in one dimension – see Section 2.2.4.

2.2.4 Two essential ingredients

The mean shift. Because the entropy is to be minimized and because this minimization will often be performed by gradient descent, the gradient of a PDF over the PDF (sometimes called normalized density gradient)

$$\frac{\nabla f_U}{f_U} = \nabla \log f_U \tag{2.3}$$

will be needed – see the $\log$-term in (2.2). PDFs have either a finite support or they tend toward zero at infinities, hence the question of stability or even existence of (2.3). Fortunately, it has been noted that this term can be approximated at t using the vector formed by t and the average within an ellipsoid centered at t of some samples drawn according to f_U [Fuk&Hos75, Fuk90]. This vector is referred to as the mean shift since it represents a shift from a local mean. This type of approximation was later popularized by a mean shift-based tracking algorithm [Com+00, Col+05] and by its use for denoising or segmentation [Com&Mee02]. The essential information is the normalized gradient direction,[#5] which both estimations [Fuk&Hos75] and [Com+00] share, differing only by a multiplicative constant. For a spherical neighborhood, the mean shift of [Fuk&Hos75, Fuk90][#6] writes as

$$\frac{\nabla f_U}{f_U}(t) \simeq \frac{d+2}{r^2}\,(\bar{s} - t) \tag{2.4}$$

where d is the dimension of the samples, r is the radius of the neighborhood, and $\bar{s}$ is the mean (or a weighted mean) of the, say n, samples of a sample set $\{U\}$ that

[#5] As such, the mean shift clearly allows detection of modes of PDFs [Com&Mee02].
[#6] In [Fuk90], see page 534.

fall within the neighborhood. Typically,

$$\bar{s} = \frac{1}{n} \sum_{\substack{s \in \{U\} \\ |s-t| \leq r}} s \ . \tag{2.5}$$

It is easy to check that, if f is a univariate normal distribution with mean μ and variance σ^2, then the normalized density gradient has the following analytical expression

$$\frac{\nabla f}{f}(t) = \frac{\mu - t}{\sigma^2} \ . \tag{2.6}$$

Compared to (2.4), one can note the analogy between *(i)* the constant $d + 2/r^2 = 3/r^2$ depending on the neighborhood radius r and $3/3\sigma^2$ depending on $\sqrt{3}\sigma$, which can be seen as the Gaussian "radius", and *(ii)* the mean $\bar{s}$ and the expected value μ, although the former is close to the latter only when t is close to μ.

The kNN framework. The intuition is clearly presented in [Fuk90, p. 255&268]. It can be summed up in a few words as follows: *(i)* the probability mass of a small region of volume v around t can be approximated by $f_U(t)\ v$; *(ii)* if N samples are drawn from f_U, this probability can also be approximated by k/N where k is the number of samples that fell in the small region; *(iii)* finally,

$$f_U(t) \simeq \frac{k}{N\ v} \ . \tag{2.7}$$

If v is independent of t, then k depends on t and (2.7) corresponds to the Parzen estimator. This raises the problem of estimating f_U wherever it is low. Indeed, unless N is indefinitely large, k may be equal to zero and, even if it is not, it might not be statistically significant. On the contrary, one can fix k and define v as the volume of the ball centered on t containing k samples among the N ones – in which case, v depends on t. This is the idea of the k nearest neighbor framework (kNN). Actually, in this context, it can be shown that (2.7) is biased [Fuk90, p. 270] and that $k - 1$ should be used instead of k.

Chapter 3

Entropy-based measures

Context

Chapter 2 provides some motivations for choosing the entropy as a mismatch measure in place of classical φ-functions. Yet, it might be interesting to have more freedom in how data consistency or regularization constraints are enforced. This could be done by comparing the observed probability density function (PDF) and a PDF model rather than dealing with the PDF of a residual. For example, the Bhattacharya coefficient could serve this purpose. However, to take advantage of the mean shift and the k nearest neighbor framework (kNN) framework (see Section 2.2.4), divergences such as the Kullback-Leibler divergence seem appropriate.

3.1 The Kullback-Leibler divergence

To fix the notations, let us write the Kullback-Leibler divergence

$$\mathfrak{D}_{\mathrm{KL}}(f,g) \doteq \int_{\mathbb{R}^d} f(t)\ \log\frac{f(t)}{g(t)}\,\mathrm{d}t \tag{3.1}$$

$$= H^{\times}(f,g) - H(f) \tag{3.2}$$

where $H^{\times}$ is the cross-entropy. The non-symmetric nature of this divergence can be characterized by comparing the Gaussians g_{za} and g_{zf} which minimize $\mathfrak{D}_{\mathrm{KL}}(f,g)$ and $\mathfrak{D}_{\mathrm{KL}}(g,f)$, respectively, where f is a given probability density function (PDF).

The problem of minimizing $\mathfrak{D}_{\mathrm{KL}}(f,g(\mu,\sigma))$ with respect to μ and σ can be easily solved analytically. For simplicity, the following developments are made for

univariate PDFs.

$$\min_{\mu,\sigma} \mathfrak{D}_{\mathrm{KL}}(f, g(\mu,\sigma)) \tag{3.3}$$

$$\Leftrightarrow \min_{\mu,\sigma} H^{\times}(f, g(\mu,\sigma)) \tag{3.4}$$

$$\Leftrightarrow \min_{\mu,\sigma} \log(\sqrt{2\pi}\sigma) \underbrace{\int_{\mathbb{R}} f(t)\ \mathrm{d}t}_{1} + \frac{1}{2\sigma^2} \int_{\mathbb{R}} f(t)\ (t-\mu)^2\ \mathrm{d}t\ . \tag{3.5}$$

Equating to zero the derivative of (3.5) with respect to μ, the first necessary condition writes

$$\mu = E[f] \tag{3.6}$$

where $E[f]$ is the expected value of f. Accounting for this result and carrying out the same development with σ, the second necessary condition is

$$\sigma^2 = \sigma_f^2 \tag{3.7}$$

where σ_f is the standard deviation of f. It can be checked that the Hessian matrix of $\mathfrak{D}_{\mathrm{KL}}(f, g(\cdot,\cdot))$ is equal to

$$D^{(2)}_{K-L} = \frac{1}{\sigma^4} \begin{bmatrix} \sigma^2 & 2\sigma(E[f]-\mu) \\ 2\sigma(E[f]-\mu) & 3\int_{\mathbb{R}} f(t)(t-\mu)^2\ \mathrm{d}t - \sigma^2 \end{bmatrix}\ . \tag{3.8}$$

At the unique potential optimum, it is equal to the positive definite matrix

$$\frac{1}{\sigma_f^2} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \tag{3.9}$$

which confirms that $(E[f], \sigma_f)$ is the unique global minimum of $\mathfrak{D}_{\mathrm{KL}}(f, g(\cdot,\cdot))$. This solution attempts to cover the whole support of f.[#1] In other words, the solution avoids to be close to zero wherever f is not, hence its name of zero-avoiding solution.

On the contrary, the problem of minimizing $\mathfrak{D}_{\mathrm{KL}}(g(\mu,\sigma), f)$ with respect to μ and σ seems less straightforward to solve

$$\min_{\mu,\sigma} \mathfrak{D}_{\mathrm{KL}}(g(\mu,\sigma), f) \tag{3.10}$$

$$\Leftrightarrow \min_{\mu,\sigma} \underbrace{-\int_{\mathbb{R}} g(\mu,\sigma)(t)\ \log f(t)\ \mathrm{d}t}_{\mathcal{A}} - H(g(\mu,\sigma)) \tag{3.11}$$

$$\Leftrightarrow \min_{\mu,\sigma} -\frac{1}{\sqrt{2\pi}\sigma} \int_{\mathbb{R}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right)\ \log f(t)\ \mathrm{d}t - \log(\sqrt{2\pi\mathrm{e}}\sigma)\ . \tag{3.12}$$

[#1] This is in accordance with the absolute continuity condition ensuring the validity of (3.1).

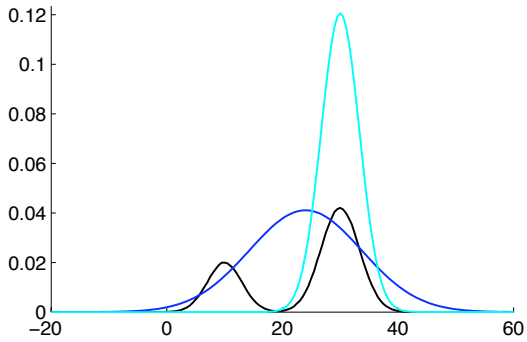

Figure 3.1 – Zero-forcing and zero-avoiding behaviors of the Kullback-Leibler divergence used as a cost function.
Black • Target PDF f, Blue • Gaussian $G(\hat{\mu},\hat{\sigma})$ minimizing $\mathfrak{D}_{\mathrm{KL}}(f, G(\mu,\sigma))$: zero-avoiding solution, Cyan • Gaussian $G(\hat{\mu},\hat{\sigma})$ minimizing $\mathfrak{D}_{\mathrm{KL}}(G(\hat{\mu},\hat{\sigma}), f)$: zero-forcing solution (obtained by numerical optimization).

Up to a multiplicative constant, the derivative of (3.12) with respect to μ is equal to

$$\int_{\mathbb{R}} \frac{\partial}{\partial \mu} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \log f(t)\, \mathrm{d}t \tag{3.13}$$

$$= \int_{\mathbb{R}} \frac{\partial}{\partial t} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \log f(t)\, \mathrm{d}t \tag{3.14}$$

$$= \underbrace{\left[\exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right)\right]_{-\infty}^{+\infty}}_{0} - \int_{\mathbb{R}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \frac{f'(t)}{f(t)}\, \mathrm{d}t\,. \tag{3.15}$$

Suppose that f has one dominant, radially symmetric mode and that $f'(t)/f(t)$ is finite for any t. Centering the Gaussian weighting in (3.15) at the mode maximizer μ_f^{dom} while assigning to σ a value close to this mode standard deviation σ_f^{dom} should both *(i)* greatly reduce the influence of the other modes in the integral and *(ii)* cancel $f'(t)/f(t)$ out when integrating over the dominant mode. The derivative (3.15) should therefore be close to zero, giving the intuition that $(\mu_f^{\mathrm{dom}}, \sigma_f^{\mathrm{dom}})$ is a potential solution to the minimization of the divergence. The corresponding Gaussian, by focusing on the dominant mode of f, has a reduced support compared to f.[#2] Then, this solution is ensured to be close to zero wherever f is, hence its name of zero-enforcing solution.

For the purpose of illustration, f was defined as a mixture of two univariate Gaussians. The zero-forcing and zero-avoiding solutions are presented in Fig. 3.1.

[#2] Again coherent with the absolute continuity condition.

3.2 How the Kullback-Leibler divergence and entropy differ

To compare two image descriptions A and B[#3] within the present statistical framework, two options have been mentioned: the entropy of the PDF f_R of the residual $R = A - B$ and the Kullback-Leibler divergence (or another similar divergence) of their respective PDFs f_A and f_B. In practice, the residual can be computed only if A and B have the same size whereas computation of the Kullback-Leibler divergence imposes no such requirements. However, the main difference lies in the geometric constraint where geometry refers here to the fact that spatial coordinates are taken into account in some way. Whether the descriptions involve geometry or not, the residual enforces a one-to-one correspondence between the description elements of A and B. Therefore, an involuntary geometric constraint is added.[#4] Then, some freedom is restored by the entropy measure. On the contrary, the Kullback-Leibler divergence between f_A and f_B discard any implicit geometric information that might have resulted from a special arrangement of the elements of the descriptions. In consequence, any desired geometric constraint must be included in the descriptions.

In the discrete domain, a description A is a set $\{a_i, i \in [1..|A|]\}$ of features. Let us assume that the index i is related to geometry. For example, it can represent the spatial coordinates of a pixel. Let θ be a set of parameters B depends on. A way to explicitly add geometry when estimating θ by minimization of the Kullback-Leibler divergence between f_A and f_B is to replace the features a_i with $(a_i^\mathsf{T}\ i)^\mathsf{T}$, and similarly for b_i. Let us see if this induces a behavior closer to the entropy of the residual.

$$\min_\theta \mathfrak{D}_{\mathrm{KL}}(f_A, f_{B(\theta)}) \tag{3.16}$$

$$\Leftrightarrow \min_\theta H^\times(f_A, f_{B(\theta)}) \tag{3.17}$$

$$\Leftrightarrow \min_\theta -\int_{\mathbb{R}^d} f_A(t)\ \log f_{B(\theta)}(t)\ \mathrm{d}t \tag{3.18}$$

$$\Leftrightarrow \min_\theta -E_{f_A}[\log f_{B(\theta)}]\ . \tag{3.19}$$

The solution of the minimization problem (3.19) can be approximated by

$$\min_\theta -\frac{1}{|A|}\ \sum_i \log f_{B(\theta)}((a_i^\mathsf{T}\ i)^\mathsf{T})\ . \tag{3.20}$$

Let $f_{B(\theta)}$ be estimated with a kernel-based method

$$f_{B(\theta)}(t) \approx \frac{1}{|B(\theta)|} \sum_i K_\sigma(t - (b_i^\mathsf{T}\ i)^\mathsf{T}) \tag{3.21}$$

[#3] For example, these descriptions can be the images themselves or a set of feature vectors computed at each pixel.

[#4] Of course, one might nonetheless take advantage of it.

where K is the centered Gaussian kernel of standard deviation σ. Then, (3.20) is equivalent to

$$\min_{\theta} -\frac{1}{|A|} \sum_i \log \frac{1}{|B(\theta)|} \sum_j K((a_i^\mathsf{T}\ i)^\mathsf{T} - (b_j^\mathsf{T}\ j)^\mathsf{T}) \tag{3.22}$$

$$\Leftrightarrow \min_{\theta} -\frac{1}{|A|} \sum_i \log \frac{1}{|B(\theta)|} \sum_j \exp -\frac{(i-j)^2}{2\sigma^2} K(a_i - b_j)\ . \tag{3.23}$$

This shows that, given the approximations that were made, adding a strong geometric constraint to the descriptions only weights the terms involved in the divergence estimation rather than turning it into a computation closer to the entropy of the residual. For comparison, following steps similar to those done for the Kullback-Leibler divergence, the entropy of the residual between the geometry-free descriptions writes[#5]

$$-\frac{1}{|A|} \sum_i \log \frac{1}{|B(\theta)|} \sum_j K((a_i - b_i) - (a_j - b_j))\ . \tag{3.24}$$

In the minimization of $\mathfrak{D}_{\mathrm{KL}}(f_{B(\theta)}, f_A)$, the entropy of $f_{B(\theta)}$ cannot be discarded as the one of f_A was in (3.19). However, the conclusion remains the same as above.

3.3 Entropy-based measures

To benefit directly from the results presented in Chapter 5, it is necessary to consider measures that can be written in terms of entropies. For example, if the symmetry property is of importance, the Jensen-Shannon divergence

$$\mathfrak{D}_{\mathrm{JS}}(f,g) = \frac{1}{2}\big(\mathfrak{D}_{\mathrm{KL}}(f,m) + \mathfrak{D}_{\mathrm{KL}}(g,m)\big), \tag{3.25}$$

where $m = 0.5(f+g)$, or the mutual information

$$I(f,g) = H(f) + H(g) - H(f,g) \tag{3.26}$$

can be used. In case the triangular inequality is also a requirement, several entropy-based metrics exist, *e.g.*,

$$\begin{cases} \mathfrak{D}_1(f,g) = H(f,g) - I(f,g) \\ \mathfrak{D}_2(f,g) = \sqrt{\mathfrak{D}_{\mathrm{JS}}(f,g)} \end{cases}. \tag{3.27}$$

[#5] Remember that $|A|$ must be equal to $|B(\theta)|$.

Part II

K NEAREST NEIGHBOR ESTIMATORS

Chapter 4

Basic ideas about kNN

4.1 Kernel-based approaches

4.1.1 PDF estimation

Kernel-based methods for probability density function (PDF) estimation make no assumption about the actual PDF. Consequently, the estimated PDF cannot be described in terms of a small set of parameters, as opposed to, *e.g.*, a Gaussian PDF defined by its mean and variance. Such methods are therefore qualified as non-parametric. Let U be a set of samples independently drawn with a given law. These estimators have the following general expression

$$f_U(t) = \frac{1}{|U|} \sum_{s \in U} K_{U,s,t}(t - s) \tag{4.1}$$

where $K_{U,s,t}$ is a multivariate kernel which bandwidth is a function of U, s, and t [Ter&Sco92] and $|U|$ is the cardinality of the sample set U. Three cases can be distinguished.

• $K_{U,s,t} = K_\sigma$**,** σ constant. This is the Parzen approach. For a uniform kernel, estimator (4.1) approximates the density at t with the relative number of samples $k(t)/|U|$ falling into the open ball of volume v_σ centered on t

$$f_U(t) = \frac{k(t)}{v_\sigma \ |U|} \ . \tag{4.2}$$

Unfortunately, the choice of the kernel bandwidth σ is critical [Sil86, Sco92]. If σ is too large, the estimate will suffer from a lack of resolution; if it is too small, the estimate will have a high statistical variability. Moreover, as the dimension of the feature space increases, the space sampling gets sparser – a problem known as the curse of dimensionality. 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 other words, the Parzen approach cannot adapt to the local sample density due to the fixed kernel bandwidth.

• $K_{U,s,t} = K_{U,s}$. This is the sample point approach [Ter&Sco92, Com03]. One bandwidth is chosen per sample s of U. Although it allows to adapt to the local sample density, the following k nearest neighbor (kNN) framework was preferred since it leads to interesting statistical estimators such as the Kullback-Leibler divergence.

• $K_{U,s,t} = K_{U,t}$. This is the balloon approach [Lof&Que65, Sai02]. The bandwidth is determined at each PDF estimation as a function of the location t. In the kNN framework, it is defined by the distance to the k-th nearest neighbor of t among the samples of U. For a uniform kernel, estimator (4.1) reads [Fuk90, p. 268]

$$f_U(t) = \frac{k}{\rho_k^d(t)\ v_d\ |U|} \tag{4.3}$$

where $\rho_k^d(t)\ v_d$ is the volume of the open ball centered on t with a radius of $\rho_k(t)$ equal to the distance to the k-th nearest neighbor of t in U excluding the sample located at t if any[#1], and v_d is the volume of the unit ball in $\mathbb{R}^d$. This approach appears to be dual to the Parzen approach (with uniform kernel): the kernel bandwidth adjusts so that the kernel includes k neighbors instead of counting the samples within a fixed range.

4.1.2 Mean shift

In a fixed-bandwidth context, the local mean in (2.4) writes

$$\bar{s} = \frac{1}{|U_\sigma(t)|} \sum_{\substack{s \in U \\ |s-t| \leq \sigma}} s \tag{4.4}$$

where t is a point of $\mathbb{R}^d$, σ is the bandwidth, and $U_\sigma(t)$ is the set of summation. As mentioned in Section 2.2.4, the mean shift is an approximation of the normalized gradient $\nabla f_U / f_U$ of the PDF f_U. As such, it can be used to reach the mode closest to a starting position t_0, unless t_0 is located in an area of low density. Indeed, the set of summation in (4.4) may be reduced to the singleton $\{t_0\}$. The mean shift is

[#1] Here, the principle of the kNN PDF estimate is only reported. Note however that, as one can guess, it suffers from some defects: it can be discontinuous, unbounded, and it does not integrate to one. Nevertheless, it can be useful by itself in high dimension [Ter&Sco92] and it represents the fundamental notion of other kNN-based estimators – see Chapter 5.

then equal to 0. This situation can be avoided by replacing the constant bandwidth σ with the distance $\rho_k(t)$ to the k-th nearest neighbor of t among the samples of U [Fuk&Hos75]

$$\bar{s} = \frac{1}{|U_\sigma(t)|} \sum_{\substack{s \in U \\ |s-t| \leq \rho_k(t)}} s \ . \tag{4.5}$$

Nonetheless, one might want to limit the influence of faraway neighbors, which can be done by turning the mean into a weighted average [Ang+08a]

$$\bar{s} = \sum_{\substack{s \in U \\ |s-t| \leq \rho_k(t)}} w_s \ s \tag{4.6}$$

where w_s is the weight of s (typically, a function of $|s-t|$) and $\sum_s w_s = 1$.

4.2 Interests of kNN

As mentioned in Section 4.1, the kNN approach is nonparametric, which ensures its generality, and locally adaptive, which reduces the effect of the curse of dimensionality. Moreover, the adaptability rule is simple: the local bandwidth selection amounts to a search for k-th nearest neighbors. Although it might be computationally costly, it is conceptually basic. This framework also allows to derive expressions of PDF-based measures (such as entropy or the Kullback-Leibler divergence) which do not explicitly depend on the underlying PDFs. Instead, they directly depend on samples. Importantly, for some kNN-based estimators, the choice of k does not seem to be critical. Finally, the principle is valid for any dimension d. Some experiments are provided in Section 5.4 to illustrate these claims.

Although the kNN framework was described in seminal works on PDF estimation [Fix&Hod51, Lof&Que65] and on the mean shift [Fuk&Hos75] a while ago, it has rarely been used in image processing so far, except for high-dimensional clustering [Geo+03].

Chapter 5

kNN entropy-based estimators

5.1 First approximations

The entropy (2.2) can be approximated by the Ahmad-Lin estimator [Ahm&Lin76]

$$H_{\mathrm{AL}}(U) = -\frac{1}{|U|}\sum_{s\in U}\log f_U(s) \tag{5.1}$$

where f_U is the Parzen estimation (4.1) of the actual probability density function (PDF).[#1] Approximation (5.1) converges in mean to the differential entropy of U.

The k nearest neighbor (kNN) PDF estimation (4.3) is biased and does not respect the fundamental PDF property of integrating to one. Nevertheless, these flaws get less critical as the dimensionality increases and the estimator has better overall performances in high dimensions than fixed bandwidth estimators [Ter&Sco92]. Let plug (4.3) into the Ahmad-Lin entropy estimation (5.1)

$$H_{\mathrm{AL}}(U) \overset{\text{kNN}}{=} -\frac{1}{|U|}\sum_{s\in U}\log\frac{k}{\rho_k^d(U,s)\ v_d\ |U|} \tag{5.2}$$

$$= \log\frac{v_d\ |U|}{k} + \frac{d}{|U|}\sum_{s\in U}\log\rho_k(U,s). \tag{5.3}$$

Moreover, the cross entropy (3.2) is equal to

$$H^{\times}(f_U, f_V) = -\,\mathrm{E}_U[\log f_V] \tag{5.4}$$

$$\simeq -\frac{1}{|U|}\sum_{s\in U}\log f_V(s). \tag{5.5}$$

[#1] Note that an entropy estimation following the same spirit has been proposed more recently [Vio&Wel97].

Again, plugging the kNN PDF expression of f_V into (5.5) leads to

$$H^{\times}(U,V) \overset{\text{kNN}}{=} \log \frac{v_d\,|V|}{k} + \frac{d}{|U|}\sum_{s\in U}\log \rho_k(V,s). \tag{5.6}$$

Subtracting (5.3) from (5.6), the following Kullback-Leibler approximation is obtained

$$\mathfrak{D}_{\text{KL}}(U,V) \overset{\text{kNN}}{=} \log \frac{|V|}{|U|} + \frac{d}{|U|}\sum_{s\in U}\log \frac{\rho_k(V,s)}{\rho_k(U,s)}. \tag{5.7}$$

Actually, this estimator has a slight bias. Nevertheless, the above development can help understanding the philosophy of the following, unbiased version.

5.2 Unbiased versions

Since the Kullback-Leibler divergence can be expressed as the difference between a cross entropy and an entropy, let us first present unbiased estimators of these quantities in the kNN framework.

5.2.1 Entropy

The following unbiased and consistent (under weak conditions on the underlying PDF) entropy estimator was proposed [Koz&Leo87, Gor+05, Leo+08]

$$H_{\text{kNN}}(U) = \log(v_d(|U|-1)) - \psi(k) + \frac{d}{|U|}\sum_{s\in U}\log \rho_k(U,s) \tag{5.8}$$

where v_d is the volume of the unit ball in $\mathbb{R}^d$, $|U|$ is the cardinality of the sample set U, ψ is the digamma function Γ'/Γ, and $\rho_k(U,s)$ is the distance to the k-th nearest neighbor of s in U excluding the sample located at s if any. Informally, the main term in estimate (5.8) is equal to the mean of the log-distances to the k-th nearest neighbor of each sample. Note that (5.8) does not depend on the PDF f_U.

While the kNN PDF estimator is competitive in high dimensions only, the entropy estimator is accurate even in the univariate case [Gor+05, Leo+08]. Moreover, the choice of k does not appear to be really crucial (see Section 5.4), as opposed to the choice of σ in the Parzen method. Actually, when the kNN approach is used for parameter estimation [Bol+06] (see Eq. (9.1)), k must be greater than the number of parameters, it must tend toward infinity when $|U|$ tends toward infinity, and such that $k/|U|$ tends toward zero when $|U|$ tends toward infinity. An admissible choice is $k = \sqrt{|U|}$.

Note that an estimate of the Rényi entropy using a related graph-based kNN framework has also been proposed for learning [Cos&Her04].

5.2.2 Cross entropy

Similarly, the cross entropy (also called relative entropy or likelihood) of two sample sets U and V can be approximated by [Koz&Leo87]

$$H^{\times}_{\mathrm{kNN}}(U,V) = \log(v_d|V|) - \psi(k) + \frac{d}{|U|}\sum_{s\in U}\log\rho_k(V,s)\ . \tag{5.9}$$

Note again that estimator (5.9) 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 V of each sample of U. Since a sample s of U does not belong to V, the search for the k-th nearest neighbor *excluding* s itself does not in fact exclude any sample of V. This is why $|V|$ appears in (5.9) whereas $|U|-1$ appears in (5.8).

5.2.3 Divergence

The Kullback-Leibler divergence can then be approximated in the kNN framework, directly from the sample sets U and V, using the entropy and cross entropy estimators (5.8) and (5.9), respectively,

$$\mathfrak{D}_{\mathrm{KL}}(U,V) \overset{\text{kNN}}{=} H^{\times}_{\mathrm{kNN}}(U,V) - H_{\mathrm{kNN}}(U) \tag{5.10}$$

$$= \log\frac{|V|}{|U|-1} + \frac{d}{|U|}\sum_{s\in U}\log\frac{\rho_k(V,s)}{\rho_k(U,s)}\ . \tag{5.11}$$

It was proven that this estimator is consistent and asymptotically unbiased [Koz&Leo87, Gor+05, Leo+08].

5.3 Remark about the biased versions

Note that (5.11) only differs from (5.7) by $\log(|U|/|U-1|)$ in absolute value and that this difference tends toward zero when the number of target samples $|U|$ tends toward infinity. Actually, concerning entropy and cross entropy, a similar remark can be made. Besides the term $|U|-1$ in (5.8) instead of $|U|$ in (5.3) (corresponding to the bias just mentioned about the divergence), the entropy estimators (5.3) and (5.8), and the cross entropy estimators (5.6) and (5.9) only differ by $\log(k)-\psi(k)$ in absolute value. Functions ψ being very close to $\log$, this difference is also not very significant (see Tab. 5.1).

5.4 Illustrative experiments

5.4.1 PDF estimation

The kNN PDF estimator performs well only at high dimensions [Ter&Sco92]. This can be seen with some basic examples.

Value of k	3	4	5	10	20	30	40
$\log(k)$	1.09	1.39	1.61	2.30	2.99	3.40	3.69
$\log(k)-\psi(k)$	0.18	0.13	0.10	0.05	0.03	0.02	0.01

Table 5.1 – Bias of the entropy estimator (5.3) and the cross entropy estimator (5.6) as a function of k.

Let $U = \{s_i\}$ be a set of ordered samples of dimension $d = 1$. Let k be equal to 1 and suppose that t belongs to the interval $[s_j, s_{j+1}]$. Expression (4.3) becomes

$$f_U(t) = \begin{cases} \frac{1}{2(t-s_j)\ |U|} & \text{if } t \leq \frac{s_j+s_{j+1}}{2} \\ \frac{1}{2(s_{j+1}-t)\ |U|} & \text{otherwise.} \end{cases} \tag{5.12}$$

Therefore, f_U is U-shaped on $[s_j, s_{j+1}]$ and tends toward infinity at the bounds of this interval. Below the lowest sample and above the highest one (*i.e.*, in the tails of the PDF), f_U is only "half-U"-shaped. If $k > m$ where m is the cardinality of the largest subset of U containing equal samples, then f_U is always finite. However, it remains *piecewise U-shaped*, although this gets less obvious as k increases. A similar behavior also occurs at higher dimensions where Voronoi cells replace intervals.

Let U be a set of $|U| = 3000$ one-dimensional samples normally distributed. Figure 5.1 shows the kNN estimation of the PDF of the samples for several values of k. Besides the irregularity of the estimation for small values of k, the other known penalizing behavior is the overestimation in the tails.

5.4.2 Entropy estimation

Despite the poor quality of the kNN PDF estimation at low dimension, the kNN entropy estimator seems accurate starting from $d = 1$. This might be explained by the smoothing effect of the $\log$-distance averaging in (5.8). By the way, after smoothing and normalizing[#2] the PDF shown in Fig. 5.1-$k = 4$, the result of Fig. 5.2 is obtained. This tends to indicate that the estimation, although very noisy, has a correct average shape.

The kNN entropy estimator also seems reasonably stable with respect to k until fairly high dimensions as shown in Fig. 5.3. Naturally, these few plots only provide motivations for using kNN-based estimators. Firm conclusions cannot be drawn upon such didactic examples.

5.4.3 Kullback-Leibler divergence estimation

Let g_{ref} be a Gaussian of dimension d with marginal means chosen uniformly in the interval $\mu_{\text{ref}} = [5, 6]$ and a random diagonal covariance matrix with diagonal

[#2]The kNN PDF estimator does not guarantee that the estimate integrates to one.

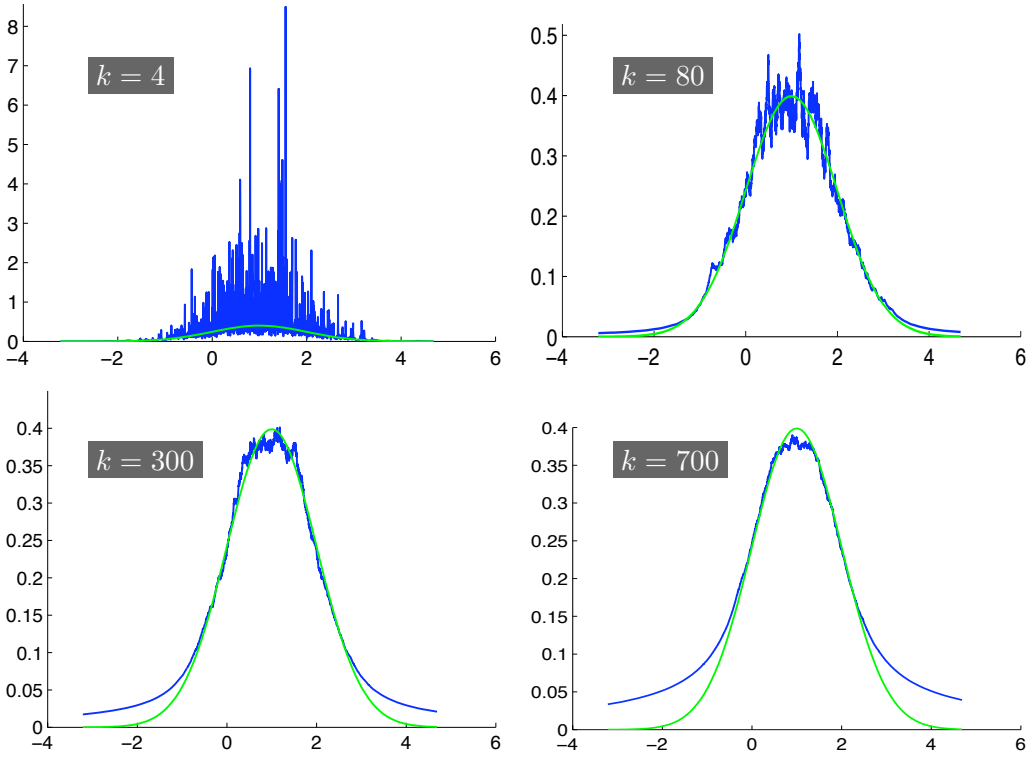

Figure 5.1 – kNN PDF estimation (4.3) from 3000 normally distributed samples for several values of k. Green • Actual Gaussian PDF, Blue • kNN estimation.

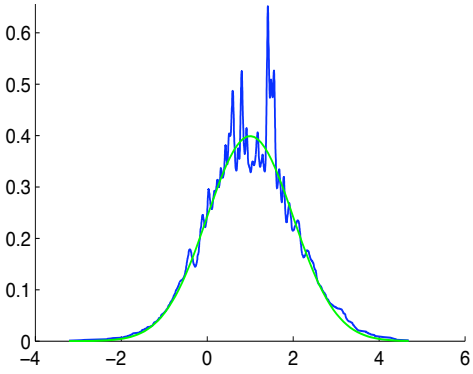

Figure 5.2 – kNN PDF estimation from 3000 normally distributed samples for $k = 4$ after smoothing and normalizing the estimate shown in Fig. 5.1. Green • Actual Gaussian PDF, Blue • kNN estimation.

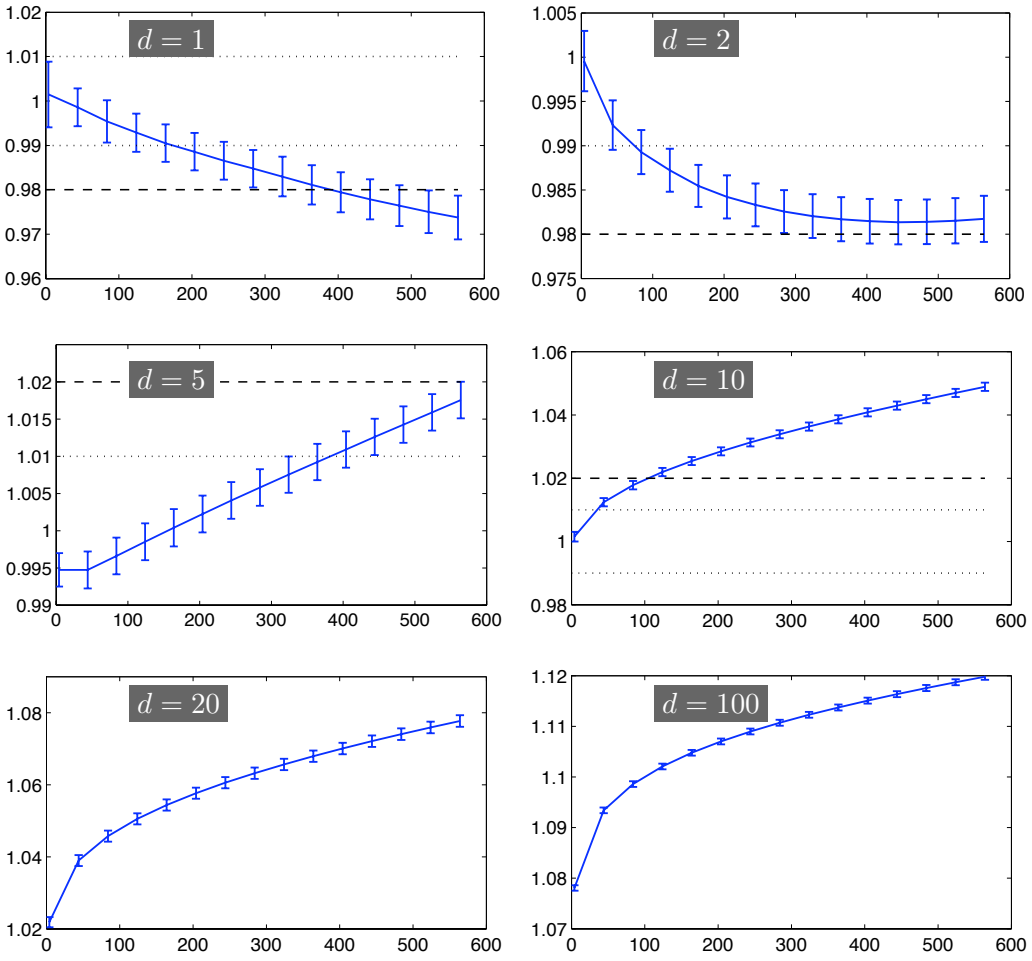

Figure 5.3 – kNN entropy estimation (5.8) from 3000 d-dimensional, normally distributed samples for values of k ranging from 4 to 564 (horizontal axes). For d and k given, 10 random diagonal covariance matrices Σ were generated. The vertical axes represent the kNN estimation divided by the corresponding true entropy $\log(\sqrt{(2\pi \mathrm{e})^d \det \Sigma})$ averaged over the 10 trials. The variations of this relative error are plotted as bars extending between $\pm$ its standard deviation. The dotted lines, respectively the dashed lines, indicate the $\pm 1\%$ error range, respectively $\pm 2\%$ error range.

elements chosen uniformly in the interval $\sigma^2_{\text{ref}} = [5, 7]$. Let g_n, $n \in [1..10]$, be Gaussians of dimension d similarly picked using intervals $\mu = [3, 4]$ and $\sigma^2 = [2, 4]$. Figure 5.4 shows the relative error of the estimation of $\mathfrak{D}_{\text{KL}}(g_n, g_{\text{ref}})$ by (5.11). The kNN estimator of the Kullback-Leibler divergence does not seem to show the same accuracy as the kNN estimator of entropy. Most notably, its performances deteriorate faster with the dimension. Yet, it appears to be relatively reliable.

The Kullback-Leibler divergence not being symmetric, similar experiments with exchanged variance intervals should be made. Thus, let the interval σ^2_{ref} be $[2, 4]$ now, and let the interval σ^2 be $[5, 7]$. Figure 5.5 shows the relative error of the estimation of $\mathfrak{D}_{\text{KL}}(g_n, g_{\text{ref}})$ by (5.11). It appears that, in these conditions, the kNN estimator of the Kullback-Leibler divergence is not very accurate nor very stable with respect to the neighboring order k, which contrasts even more with the entropy estimator. Since the Kullback-Leibler divergence estimator was defined as the difference between the kNN estimator of cross entropy and the kNN estimator of entropy, this could be an indication that the cross entropy estimator is not as accurate as the entropy estimator, or this could be an illustration that combining accurate estimators does not necessarily build an accurate estimator of the combined underlying quantities. Yet, because the Kullback-Leibler divergence, or other entropy-based measures, is to be used as a similarity measure, the actual value of the estimation is of limited importance compared to respecting the relative order of Kullback-Leibler divergences. As a partial answer to whether this property is verified (in the current context) by the kNN estimator of the Kullback-Leibler divergence, the results of Fig. 5.5 were plotted in a different way. For a given dimension d, 10 Kullback-Leibler divergences were considered. Let us sort the true values in ascending order. For a given k, let us apply to the estimations (5.11) the same rearrangement as the one undergone by the true values when sorted. If these estimations are then in ascending order, the estimator, if not adapted to determine absolute values, can still be regarded as suitable for comparison purpose. This alternative presentation of the results previously discussed is shown in Fig. 5.6. Overall, in these experiments (Fig. 5.6), the Kullback-Leibler divergence estimator exhibits a fairly good coherence with the true Kullback-Leibler divergence with globally monotonic estimations. The correlation is better at low dimension and decreases rapidly with the neighboring order k – see Fig. 5.7.

As a final remark, let us mention that the set of experiments which provided the best results (see Fig. 5.4) corresponds to $\mathfrak{D}_{\text{KL}}(g_1, g_2)$ where g_2 has a variance larger than the variance of g_1. Although the Gaussians have infinite support, it is tempting to note that, in general, this situation tends to be in accordance with the absolute continuity condition.

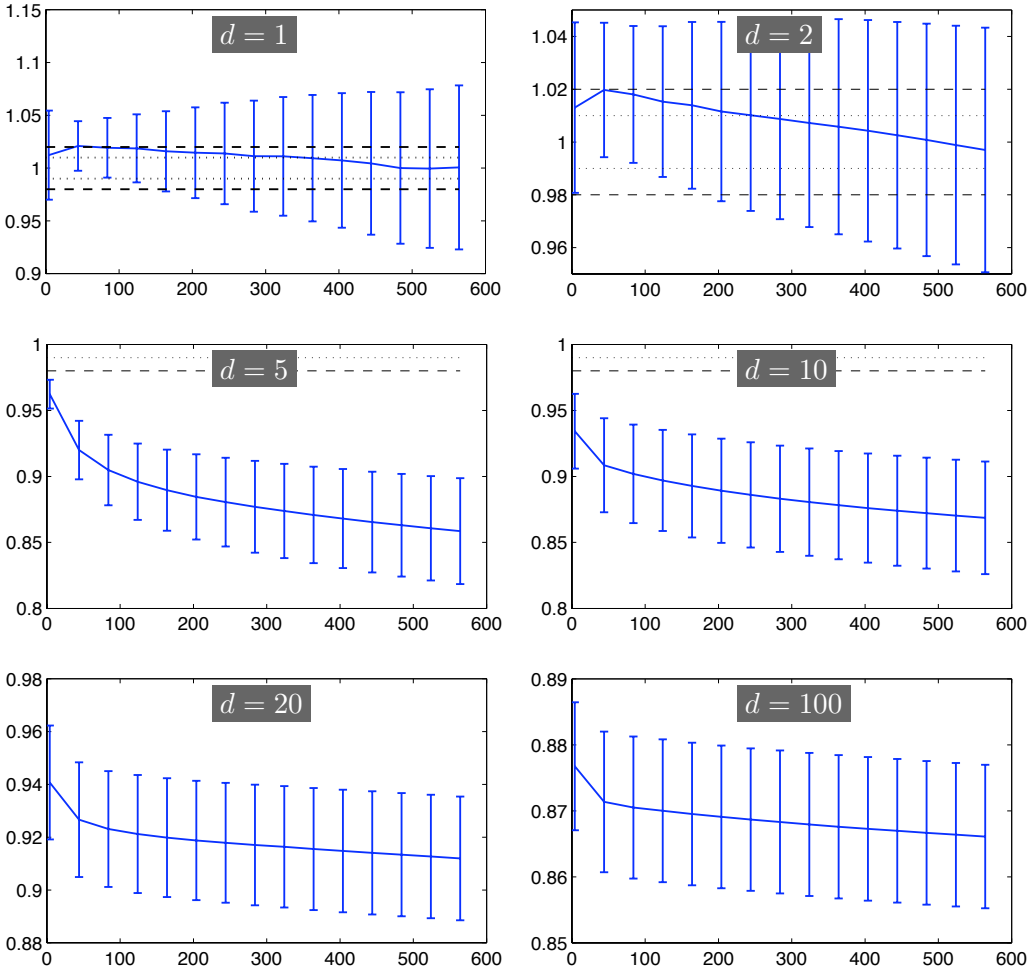

Figure 5.4 – kNN Kullback-Leibler divergence estimation (5.11) of $\mathfrak{D}_{\mathrm{KL}}(g_n, g_{\mathrm{ref}})$ from 3000 d-dimensional samples following g_{ref} and 3000 d-dimensional samples following g_n, $n \in [1..10]$. The estimation was performed for several values of d and for values of k ranging from 4 to 564 (horizontal axes). The vertical axes represent the kNN estimation divided by the corresponding true divergence $0.5(\log(\det(\Sigma_{\mathrm{ref}})/\det(\Sigma_n)) + \mathrm{trace}(\Sigma_{\mathrm{ref}}^{-1} * \Sigma_n) + ([\mu]_{\mathrm{ref}} - [\mu]_n)^{\mathsf{T}} \Sigma_{\mathrm{ref}}^{-1} ([\mu]_{\mathrm{ref}} - [\mu]_n) - d)$ averaged over the 10 trials at d and k fixed. The variations of this relative error are plotted as bars extending between $\pm$ its standard deviation.

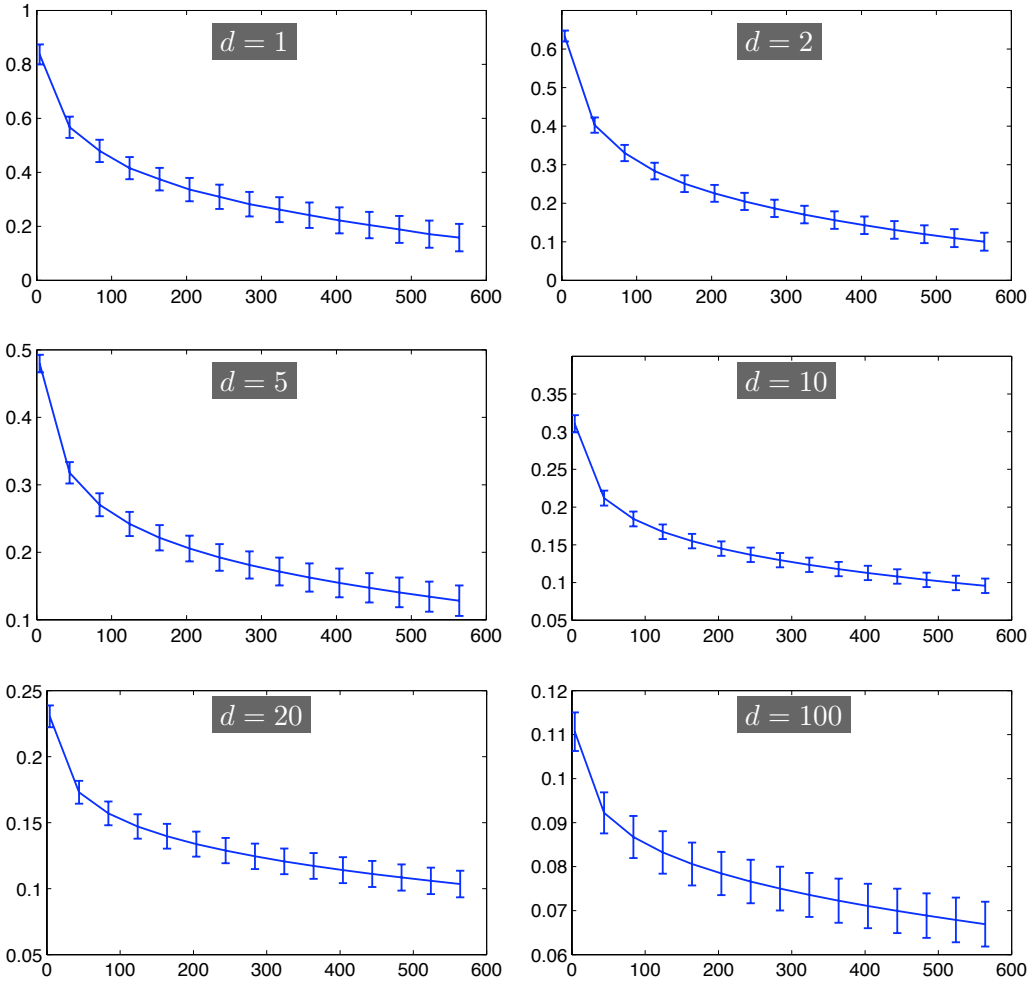

Figure 5.5 – kNN Kullback-Leibler divergence estimation (5.11) of $\mathfrak{D}_{\mathrm{KL}}(g_n, g_{\mathrm{ref}})$ from 3000 d-dimensional samples following g_{ref} and 3000 d-dimensional samples following g_n, $n \in [1..10]$. The estimation was performed for several values of d and for values of k ranging from 4 to 564 (horizontal axes). The vertical axes represent the kNN estimation divided by the corresponding true divergence $0.5(\log(\det(\Sigma_{\mathrm{ref}})/\det(\Sigma_n)) + \mathrm{trace}(\Sigma_{\mathrm{ref}}^{-1} * \Sigma_n) + ([\mu]_{\mathrm{ref}} - [\mu]_n)^{\mathsf{T}}\, \Sigma_{\mathrm{ref}}^{-1}\, ([\mu]_{\mathrm{ref}} - [\mu]_n) - d)$ averaged over the 10 trials at d and k fixed. The variations of this relative error are plotted as bars extending between $\pm$ its standard deviation.

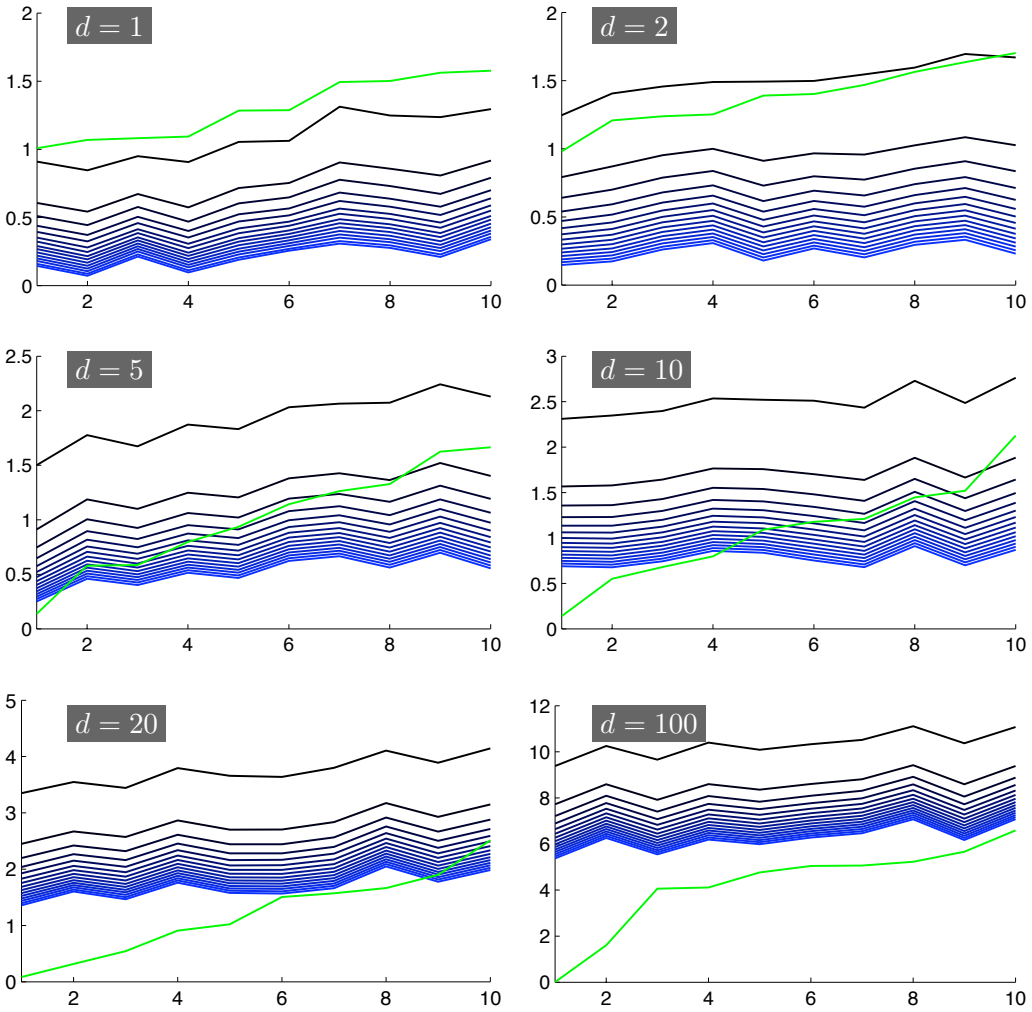

Figure 5.6 – kNN Kullback-Leibler divergence estimation (5.11) of $\mathfrak{D}_{\mathrm{KL}}(g_n, g_{\mathrm{ref}})$ from 3000 d-dimensional samples following g_{ref} and 3000 d-dimensional samples following g_n, $n \in [1..10]$. The estimation was performed for several values of d and for values of k ranging from 4 to 564. The horizontal axes represent the 10 trials at d and k fixed. The vertical axes represent the Kullback-Leibler divergence. For practical reasons, the true divergences have been decreased by 1, 3, 7, 15, and 89 for d equal to 2, 5, 10, 20, and 100, respectively.
Green • True values, Black • Estimations for $k = 4$, Blue • Estimations for $k = 564$. The shades of blue correspond to varying values of k between 4 and 564 with a step of 40.

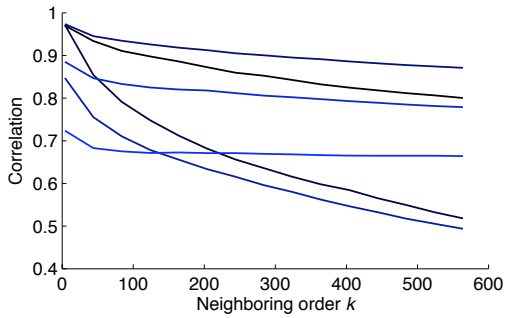

Figure 5.7 – Correlation between the kNN Kullback-Leibler divergence estimations shown in Figs. 5.5 and 5.6 and the respective true divergences as a function of k.
Black • Correlation for $d = 1$, Blue • Correlation for $d = 100$. The shades of blue correspond to the intermediate values 2, 5, 10, and 20. Especially for low values of k, the correlations for the 3 lowest dimensions are higher than the ones for the 3 highest.

Chapter 6

Some remarks on kNN

6.1 Link with classical regularization functions

Let u, a two-dimensional grayscale image, be the solution to an ill-posed inverse problem. A common constraint on u is to require that the gradient norm image $|\nabla u|$ be of small norm. This is usually expressed as follows

$$\arg\min_{v} \int_{U} \varphi(|\nabla v(s)|)\, \mathrm{d}s \tag{6.1}$$

where U is the image domain and φ is a positive function respecting specific conditions [Cha+97]. Let $B_1(s)$ be the circle of radius 1 centered on s. $B_1(s)$ defines a spatial neighborhood of s. Assuming that $|\nabla u(s)| \neq 0$, let t be the point of $B_1(s)$ in the direction $|\nabla u(s)|$ from s. Then, the norm $|\nabla u(s)|$ can be approximated with $\rho_B(s) = |u(s) - u(t)|$. Therefore, a possible discretization of (6.1) is

$$\arg\min_{u} \sum_{s \in U} \varphi(\rho_B(s)) \tag{6.2}$$

where, for convenience, U has been reused to denote the set of samples in the image domain. If u is smooth enough, t is surely the farthest neighbor of s (in the neighborhood $B_1(s)$) in the feature space defined by gray levels u, and $\rho_B(s)$ is its distance to s.

In conclusion, the kNN entropy (5.8) contains mainly a sum of $\log$-distances to (k-th) nearest neighbors searched for in the feature space among all available samples while (6.2) is better described as a sum of φ-distances to farthest neighbors searched for in the feature space among spatially close samples. This difference of viewpoint is comparable to the contrast between classical filtering and nonlocal filtering [Bua+05b, Bou+07].

6.2 Distance between features

As previously seen, kNN estimators rely on distances ρ_k between feature samples. In Section 3.2, the features were enriched with geometric information. This raises the classical question of the relative weights of heterogeneous feature components. Since the $\mathcal{L}^2$ norm treats each component equally, one usually employs a weighted version instead or might be tempted to use other metrics tailored specifically for the features. For example, the Earth mover's distance [Rub+00] is adapted to comparing histograms. However, the infinitely many choices of distance lead to as many different kNN estimators. To get a hint about the consequences of using a specific metric, let us come back to the PDF estimation.

The principle of the kNN framework is to approximate the PDF value at some location t with a local sample density (4.3). Thus, this density is assumed constant within the ball of radius $\rho_k(t)$ [Fuk&Hos75, Fuk90]. (In discrete terms, all the samples within this ball are equiprobable.) Yet, this might not be true. Some locations within the ball might not even be valid features. Then, it is indeed necessary to find the distance definition that matches the distribution of features.

The actual distribution of features such as image patches has been studied [Lee+03, Sri+03, Car+08]. Suppose that, according to such a study, a metric $\mathcal{M}$ is designed for a particular feature definition. Following the idea leading to (4.3), the PDF at t becomes

$$f_U(t) = \frac{k}{V_{k,d}(t)\ |U|} \tag{6.3}$$

where $V_{k,d}(t)$ is the volume of the ball $B_{k,d}(t)$ centered on t with a radius of $\rho_k(t)$ in the metric $\mathcal{M}$

$$V_{k,d}(t) = \int_{B_{k,d}(t)} \sqrt{|\det \mathcal{M}(r)|}\ \mathrm{d}r\ . \tag{6.4}$$

Let φ be the bijection defined by

$$\begin{array}{rccl} \varphi\ :\ B_d(t) & \longrightarrow & B_{k,d}(t) \\ s & \longmapsto & r = \rho_k(t)(s-t)+t \end{array} \tag{6.5}$$

where $B_d(t)$ is the ball of radius 1 centered on t. The Jacobian of φ is equal to $\rho_k^d(t)$. Therefore,

$$V_{k,d}(t) \ = \int_{B_d(t)} \sqrt{|\det \mathcal{M}(\varphi(s))|}\ \rho_k^d(t)\ \mathrm{d}s \tag{6.6}$$

$$= \rho_k^d(t) \int_{B_d(t)} \sqrt{|\det \mathcal{M}(\rho_k(t)(s-t)+t)|}\ \mathrm{d}s \tag{6.7}$$

$$:= \rho_k^d(t)\ v_d(t)\ . \tag{6.8}$$

It must be checked that $v_d(t)$ is indeed not a constant function. Assume that $v_d(t)$ is in fact independent of t, equal to $\tilde{v}_d$, regardless of the sample distribution. Let t_1 and t_2 be two locations in the feature space such that their respective sets of k nearest neighbors (used to determine $\rho_k(t_1)$ and $\rho_k(t_2)$, respectively) are disjoint. According to the previous assumption, $v_d(t_1) = v_d(t_2) = \tilde{v}_d$. Let us disturb the neighborhood set of t_1 (*e.g.*, by moving the farthest neighbor or by adding a sample closer to t_1 than $\rho_k(t_1)$) in such a way that $\rho_k(t_1)$ changes whereas the neighborhood set of t_2 remains unchanged. This new sample distribution is plausible. While $v_d(t_2)$ will still be equal to $\tilde{v}_d$, the integral $v_d(t_1)$ will, in general, not remain constant. This is in contradiction with the assumption of constancy of $v_d(t)$.

Therefore, using a metric adapted to the features (or to their distribution) not only straightforwardly modifies the kNN entropy estimation (5.8) through different values of the ρ_k's but also through local terms $v_d(t)$.[#1] By analogy with the development leading to (5.8), the constant involving v_d should be replaced with a term involving $v_d(\cdot)$

$$H_{\mathrm{kNN}}(\mathcal{M}, U) = \log\left[\left(\Pi_{s\in U} v_d(s)\right)^{1/|U|}(|U|-1)\right] - \psi(k) + \frac{d}{|U|}\sum_{s\in U}\log\rho_k(U,s)\ . \quad (6.9)$$

Note, however, that the $v_d(s)$'s may cancel out in some cases such as the Kullback-Leibler divergence estimator proposed in Section 5.2.3. As a result, expression (5.11) is valid for any metric $\mathcal{M}$ used to compute ρ_k.

[#1] For the Euclidean metric, $\det \mathcal{M} = 1$ and, as expected, $v_d(t)$ is then equal to v_d, the volume of the unit ball.

Part III

SOME IMAGE & VIDEO PROCESSING TASKS

Chapter 7

Segmentation

Context

In image or video segmentation, example-based similarity can be defined by comparing a local description to the global measure representing synthetically all the local descriptions within a region. For example, the local description can be the gray level of a pixel, the corresponding global measure being the mean gray level of a set of pixels. Then segmentation consists in computing the set of pixels maximizing this self-similarity. In practice, it is often more natural to write the problem as the minimization of a self-dissimilarity. Usually, this task has no closed form and is therefore solved iteratively from an initial guess. Because we are dealing with object boundaries, active contours represent a method of choice.

This chapter focuses on dissimilarities written as an integral on a domain of a function which can depend on this domain but deals with their minimization using active contours. The derivative with respect to the domain of such a dissimilarity, the so-called shape derivative, is a function of a velocity field applied to the domain boundary. For a given, non-optimal domain, a velocity such that the shape derivative is negative indicates a way to deform the domain in order to decrease its self-dissimilarity. In the continuous framework, assigning to the velocity the opposite of the gradient associated with the $\mathcal{L}^2$ inner product is a common practice. Nevertheless, it can be noted that the negativity of the shape derivative is not preserved, in general, when discretizing this velocity. Although this phenomenon is unlikely to occur in practice if the discretization is fine enough, its study led us to suggest an alternative approach relying on predefined velocities. It offers a way to impose constraints on the segmentation.

7.1 Shape derivative

7.1.1 Notations

From now on, a dissimilarity will be referred to as an energy.

In the following, D is a subset of $\mathbb{R}^2$ and the image to segment, f, is a function from D to $\mathbb{R}^m$. Domain Ω is an open set of D. Γ is the oriented boundary $\partial\Omega$ of Ω, and s is the arc-length parameterization of Γ.[#1] For convenience, the notation $a(s)$ refers to $a(\Gamma(s))$. Samples on Γ are denoted by Γ_i, $i \in [1, n]$. Arc-length s_i is such that $\Gamma(s_i) = \Gamma_i$. Then, the notations $a(s_i)$ and $a(\Gamma_i)$ are equivalent. Note that s_1 is equal to 0 and s_n is equal to $L - (s_{n+1} - s_n)$ where L is the length of Γ and $\Gamma(s_{n+1}) = \Gamma(s_1)$. The contour segment between Γ_i and Γ_{i+1} (*i.e.*, a line segment, a spline segment... depending on the contour representation) is denoted by γ_i.

Let U and V be two functions from Ω to $\mathbb{R}^2$ called velocities. The $\mathcal{L}^2$ inner product on the space of velocities restricted to Γ is defined as

$$\langle U \ , \ V \rangle = \int_\Gamma U(s) \cdot V(s) \, \mathrm{d}s \tag{7.1}$$

where $\cdot$ is the dot product.

7.1.2 General and specific expressions

Some details about active contours and the shape derivative are provided in Appendices A, B.1, B.2, and C. To fix the ideas, a typical context is reminded below.

Let us consider the energy

$$E(\Gamma) = \int_\Omega \phi(\Gamma, x) \, \mathrm{d}x + \int_\Gamma \varphi(s) \, \mathrm{d}s, \tag{7.2}$$

a function of a contour Γ. Since the set of simple closed curves is not a vector space, the derivative of (7.2) with respect to Γ cannot be expressed in the usual way. Let $\Omega(\tau)$, $\tau \geq 0$, be a family of domains such that $\Omega(\tau = 0) = \Omega$. When τ increases, $\Gamma(\tau)$ can be considered as a deforming interface in a medium characterized by ϕ and φ. Hence, some results in continuum mechanics [Hau&Cho93] can be applied to determine the derivative of (7.2) with respect to τ at τ equal to zero [Deb+01]. The study of such energies and their variations was further developed in the framework of shape optimization [Sch92, Sok&Zol92, Del&Zol01, Hin&Rin03, Jeh+03]. In this context, the following expression is known as the shape derivative of (7.2)

$$\begin{aligned} \mathrm{d}E(\Gamma, V) = & \int_\Omega \left. \frac{\partial \phi(\Gamma(\tau), x)}{\partial \tau} \right|_{\tau=0} \mathrm{d}x \\ & - \int_\Gamma \left(\phi(\Gamma, s) - \frac{\partial \varphi(s)}{\partial N} + \varphi(s) \, \kappa(s) \right) N(s) \cdot V(s) \, \mathrm{d}s \end{aligned} \tag{7.3}$$

[#1] Ω is assumed to be such that Γ is a smooth boundary without self-intersection. In case Ω is composed of several connected components, the problem can be divided into subproblems dealing each with a given component. Note however that the issue of change of topology is not covered here.

where V is by definition the restriction to Γ of a velocity field defined on Ω, N is the inward unit normal of Γ, and κ is the curvature of Γ.

Under weak assumptions, the shape derivative of the domain integral in (7.2) has an equivalent expression in the form of a boundary integral [Sok&Zol92, Sol&Ove05]. As is clear from Section 7.2.1, such an expression is convenient in the active contour framework since it allows to easily deduce an evolution equation.

In the following, it is assumed that the shape derivative of (7.2) has been rewritten into a boundary integral

$$\mathrm{d}E(\Gamma, V) = -\int_{\Gamma} \Psi(\Gamma, s)\; N(s) \cdot V(s)\; \mathrm{d}s = \langle -\Psi\; N\;,\; V \rangle, \tag{7.4}$$

either because one of the two conditions (C.2) or (C.19) applies (see Appendix C), or as the result of another development.

7.2 From continuous to discrete formulation

The usual, or direct, approach when having recourse to the shape derivative in the active contour framework corresponds to choosing the gradient associated with the $\mathcal{L}^2$ inner product as the descent direction (among the velocities that ensure the negativity of the shape derivative) and to discretizing it. Due to discretization, this direct approach implies an error possibly responsible for the loss of the negativity condition. In contrast, a constrained approach relying on predefined velocities can guarantee that the negativity condition still holds after discretization.

7.2.1 Direct approach

Negativity of the shape derivative. The shape derivative (7.4) is a function of a velocity field V. Since the energy (7.2) must be minimized, it is necessary to choose V such that (7.4) is negative. Interpreting (7.4) as a $\mathcal{L}^2$ inner product on the space of velocities, the velocity

$$G = -\Psi\; N \tag{7.5}$$

can be identified with the gradient associated with this inner product. It is called the shape gradient of (7.2) [Sok&Zol92, Del&Zol01, Hin&Rin03, Cha+05]. Then, taking a steepest descent approach, it seems natural to choose [Deb+01, Jeh+03]

$$V(s) = -G(\Gamma, s) \tag{7.6}$$

in the following active contour evolution equation

$$\frac{\partial \Gamma}{\partial \tau} = V(\tau)\;. \tag{7.7}$$

Minimization in the continuous framework. The implementation of the evolution equation (7.7) can be based on a finite difference approximation of the derivative with respect to τ verifying the CourantFriedrichsLewy (CFL) condition. Instead, a line search strategy can be used [Hin&Rin03]

$$\begin{cases} \Gamma^0 \\ \Gamma^{+1} = \Gamma + \alpha\, V \end{cases} \tag{7.8}$$

where Γ^0 is an initial contour, superscript $^{+1}$ represents the next element of a sequence,[#2] α is a positive constant, and V is given by (7.6). The optimal value for α can be computed as follows

$$\alpha = \arg\min_{a \geq 0} E(\Gamma + a\, V)\ . \tag{7.9}$$

Discretization and induced velocity. In practice, the active contour Γ is sampled. For example, it can be represented by a polygon $\{\Gamma_i,\ i \in [1..n]\}$ without self-intersection [Ger&Ref96]. The corresponding discrete version of evolution equation (7.8) is

$$\begin{cases} \Gamma_i^0 \\ \Gamma_i^{+1} = \Gamma_i + \alpha\, V(\Gamma_i) \end{cases} \tag{7.10}$$

where $\{\Gamma_i^0\}$ is an initial polygon. Note that (7.10) does not make use of V along the edges γ_i of the polygon. Instead, it implicitly defines a velocity $\tilde{V}$, called induced velocity, which transforms the edges γ_i into the edges γ_i^{+1}. However, except at the polygon vertices, it is unlikely that such a transformation matches the velocity (7.6)[#3] (see Fig. 7.1a). As a consequence, the negativity of the shape derivative (7.4) at $\tilde{V}$ is not guaranteed. In other words, the discrete evolution equation (7.10) might not generate a minimizing sequence of contours.

As illustrated in Fig. 7.1b, this problem is not specific to the polygonal representation. It also arises if the contour is represented by a smooth curve since it is due to the sampling of V. Only the contour samples are translated correctly. The contour segments in-between the translated samples are defined *a posteriori* by the selected interpolation model (polygon, uniform cubic B-spline...).

However, as brought to our attention [Ano07], if the edge length is small enough, $\mathrm{d}E(\Gamma, \tilde{V})$ remains negative and, consequently, the evolution (7.10) is guaranteed to generate a minimizing sequence. (In practice, the discretization error decreases, as illustrated in Fig. 7.2 with a smooth contour representation.) A global edge length upper bound depending on maximal variations of Ψ over Γ can

[#2] The sequence $x(0) = x^0,\ x(n+1) = f(x(n))$ is denoted by $x^0,\ x^{+1} = f(x)$.

[#3] In particular, (7.6) does certainly not transform, in general, a polygon into another polygon.

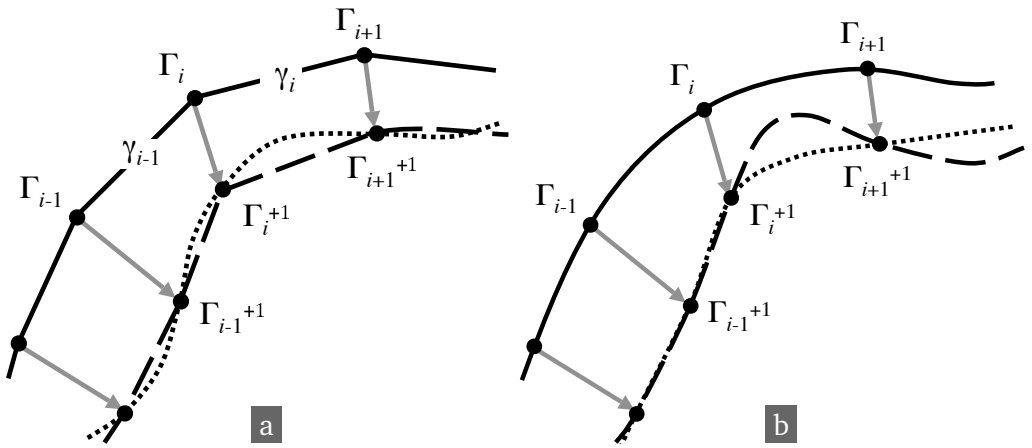

Figure 7.1 – Incorrect deformation due to the sampling of V. [a] *Disks:* polygon vertices; *Solid line:* polygon before deformation; *Dashed line:* polygon defined by the translated vertices; *Dotted line:* polygon deformed according to V. [b] *Disks:* curve samples; *Solid line:* curve before deformation; *Dashed line:* curve interpolating the translated samples; *Dotted line:* curve deformed according to V.

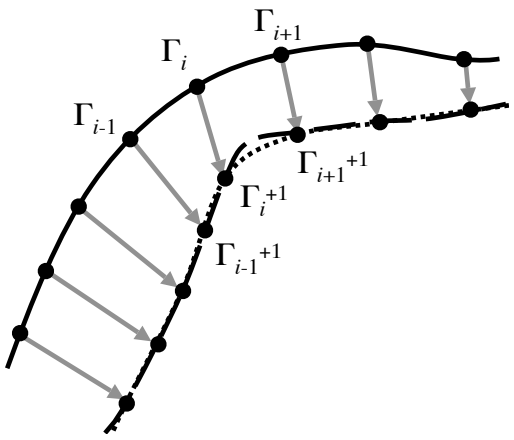

Figure 7.2 – The error between the correctly deformed curve (dotted line) and the wrongly estimated curve (dashed line) decreases when resolution gets higher (to be compared with Fig. 7.1b). *Disks:* curve samples; *Solid line:* curve before deformation; *Dashed line:* curve interpolating the (correctly) translated samples; *Dotted line:* curve correctly deformed.

be determined [Deb+07]. The edge length could also be adapted locally to be small in portions of Γ where Ψ varies a lot and larger where Ψ varies slowly [Tat&Lac02].

Finally, note that (7.10) might converge too early since the condition $\{V(\Gamma_i) = 0, i \in [1, n]\}$ is less restrictive than $\{V(s) = 0, s\}$. But again, the smaller the edge length, the less critical. Nevertheless, instead of a condition on the edge length, one can wonder if there is a way to choose V such that $\tilde{V}$ is equal to V.

7.2.2 Constrained approach

Negativity of the shape gradient. In order to guarantee that, after discretization, the induced velocity $\tilde{V}$ matches the original velocity V, the domain transformations can be restricted, beforehand in the continuous framework, to a linear combination of a set of predefined transformations [Deb+07][#4]

$$T(\tau) = \sum_i \beta_i T_i(\tau), \ \beta_i \in \mathbb{R} \ . \tag{7.11}$$

The differentiation of (7.11) with respect to τ leads to

$$V = \sum_i \beta_i V_i \tag{7.12}$$

where V_i is a so-called predefined velocity. In this context, an appropriate velocity V is defined by a choice of the β_i's that satisfies the negativity of the shape derivative (7.4).

The shape derivative can be rewritten as

$$\begin{aligned} \mathrm{d}E(\Gamma, V) &= \mathrm{d}E(\Gamma, \sum_i \beta_i V_i) && (7.13)\\ &= \sum_i \beta_i \ \mathrm{d}E(\Gamma, V_i) && (7.14)\\ &= \beta \cdot \mathrm{d}E_{\mathrm{pre}}(\Gamma) && (7.15) \end{aligned}$$

where $\cdot$ is the dot product, β is the vector of components β_i, and $\mathrm{d}E_{\mathrm{pre}}(\Gamma)$ is the vector of components $\mathrm{d}E(\Gamma, V_i)$. Taking a steepest descent approach, β should be such that $\mathrm{d}E(\Gamma, V)$ is as negative as possible. The Cauchy-Schwarz inequality implies that

$$|\mathrm{d}E(\Gamma, V)| = |\beta \cdot \mathrm{d}E_{\mathrm{pre}}(\Gamma)| \leq |\beta| \ |\mathrm{d}E_{\mathrm{pre}}(\Gamma)| \tag{7.16}$$

with equality when β and $\mathrm{d}E_{\mathrm{pre}}(\Gamma)$ are linearly dependent. Therefore, the β_i's should be set as follows

$$\beta_i = -\gamma \ \mathrm{d}E(\Gamma, V_i) \tag{7.17}$$

where γ is a positive constant.

[#4] For definitions and notations, see Appendix B.2.

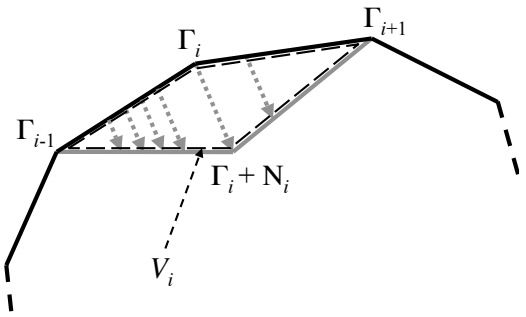

Figure 7.3 – A possible choice for the predefined velocity V_i.

Basic examples of predefined velocities. According to the remarks of Section 7.2.1, it can be deduced that the predefined velocities must be consistent with the contour representation: the representation must be preserved when the contour is deformed by a linear combination of the V_i's. Moreover, ideally, the V_i's should allow the generation of any velocity V by linear combination – at least, if there is no *a priori* knowledge about the optimal contour. There is no such basis of velocities. However, the V_i's must generate a reasonable variety of velocities. If the contour is represented by a polygon, the following definitions can be considered.[#5]

(i) At vertex Γ_i, a pseudo-normal $N(\Gamma_i)$ is defined [Lob&Vie95, Del&Mon01] and V_i is the velocity collinear to $N(\Gamma_i)$ at Γ_i and transforming $(\Gamma_{i-1}, \Gamma_i, \Gamma_{i+1})$ into $(\Gamma_{i-1}, (\Gamma_i + N(\Gamma_i)), \Gamma_{i+1})$ (see Fig. 7.3). In other words, V_i is a vector field with support $[\Gamma_{i-1}, \Gamma_{i+1}]$, linear from zero (at Γ_{i-1}) to $N(\Gamma_i)$ (at Γ_i), and linear again back to zero – at Γ_{i+1}. This definition involves no *a priori* knowledge.

(ii) The previous definition can be modified to introduce some *a priori* knowledge. For example, in tracking, the approximate motion of the object of interest might be known. In particular, a joint segmentation and motion computation method [Cre&Soa03] certainly requires computation of the motion for a given, fixed segmentation (and vice-versa). Therefore, if m_i is the estimated motion of Γ_i, then V_i can be defined similarly to *(i)* by replacing $N(\Gamma_i)$ with m_i or $m_i/|m_i|$.

These definitions will be referred to as polygonal predefined (PoPD) velocities.

[#5] There validity will be checked later on.

Minimization in the continuous framework and discretization. The minimizing sequence (7.8) of the direct approach is replaced with

$$\left\{\begin{array}{l}\Gamma^0\\ \Gamma^{+1} = \Gamma + \delta\, V\\ \qquad = \Gamma + \delta\ \sum_i \beta_i V_i\\ \qquad = \Gamma - \delta\gamma\ \sum_i \mathrm{d}E(\Gamma, V_i)\ V_i\\ \qquad = \Gamma - \alpha\ \sum_i \mathrm{d}E(\Gamma, V_i)\ V_i\end{array}\right. \tag{7.18}$$

where the optimal value for α can be computed as follows

$$\alpha = \arg\min_{a\geq 0}\ E\Big(\Gamma - a\ \sum_i \mathrm{d}E(\Gamma, V_i)\ V_i\Big)\ . \tag{7.19}$$

With a polygonal representation, the discretization of (7.18) leads to the following evolution equation

$$\left\{\begin{array}{l}\Gamma_i^0\\ \Gamma_i^{+1} = \Gamma_i - \alpha\ \sum_j \mathrm{d}E(\Gamma, V_j)\ V_j(\Gamma_i)\end{array}\right. . \tag{7.20}$$

For the PoPD velocities, the sum over the predefined velocities reduces to a single term since at Γ_i, only V_i is different from zero

$$\left\{\begin{array}{l}\Gamma_i^0\\ \Gamma_i^{+1} = \Gamma_i - \alpha\ \mathrm{d}E(\Gamma, V_i)\ V_i(\Gamma_i)\end{array}\right. . \tag{7.21}$$

Coherence between continuous and discrete evolutions. In order to establish that (7.18) and (7.21) lead to identical evolutions, it suffices to show that the deformations $\tilde{V}$ induced by (7.21) on contour segments γ_i are equal to the deformations V of the continuous evolution (7.18). This would also prove that the negativity of the shape derivative is preserved after discretization.

For example with the polygonal representation, let x be a point on the edge γ_i (see Fig. 7.4). There exists $t \in [0, 1]$ such that

$$x = \gamma_i(t) = (1 - t)\ \Gamma_i + t\ \Gamma_{i+1}\ . \tag{7.22}$$

Let V_i be a PoPD velocity – either of the two definitions. The velocity $V(t)$ at $\gamma_i(t)$ is the combination of two velocities

$$V(t) = \beta_i\ V_i(t) + \beta_{i+1}\ V_{i+1}(t)\ . \tag{7.23}$$

Therefore, in the continuous framework, the point $\gamma_i(t)$ is translated to

$$\gamma_i(t) + V(t) = (1 - t)\ \Gamma_i + \beta_i\ V_i(t) + t\ \Gamma_{i+1} + \beta_{i+1}\ V_{i+1}(t)\ . \tag{7.24}$$

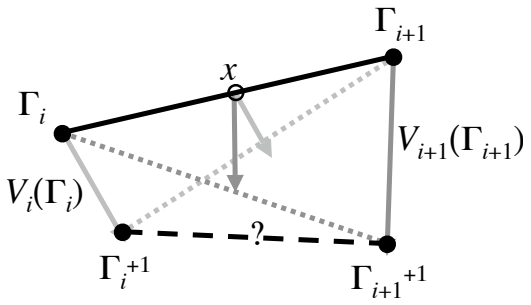

Figure 7.4 – Transformation of point x on edge γ_i.

The predefined velocities are such that

$$\begin{cases} V_i(t) = (1-t)\, V_i(\Gamma_i) \\ V_{i+1}(t) = t\, V_{i+1}(\Gamma_{i+1}) \end{cases} . \tag{7.25}$$

Combining (7.24) and (7.25) leads to

$$\gamma_i(t) + V(t) = (1-t)\,(\Gamma_i + \beta_i\, V_i(\Gamma_i)) + t\,(\Gamma_{i+1} + \beta_{i+1}\, V_{i+1}(\Gamma_{i+1})) \tag{7.26}$$

$$= (1-t)\,\Gamma_i^{+1} + t\,\Gamma_{i+1}^{+1}, \tag{7.27}$$

which means that the translation of the point $\gamma_i(t)$ in the continuous framework (left-hand side) is a point on the line segment $[\Gamma_i^{+1}, \Gamma_{i+1}^{+1}]$ obtained by joining the *discrete* translations of Γ_i and Γ_{i+1} (right-hand side). Moreover, when $\gamma_i(t)$ describes the edge γ_i, then the translation of $\gamma_i(t)$ describes this whole line segment. As a conclusion, the continuous equation (7.18) and the discrete equation (7.21) lead to identical evolutions.

The previous development is pretty straightforward and serves as an illustration. Other contour representations can be considered. For example, if the contour is represented by a uniform cubic B-spline [Uns+93, Bri+00, Jac+01, Pre+05], then one can think of basing the predefined velocities on the so-called blending function. It can be shown [Deb+07] that such a spline representation associated with appropriate predefined velocities also guarantees that the discrete evolution matches the continuous evolution (7.18).

7.3 Some remarks about predefined velocities

7.3.1 Normalization

The weight of a predefined velocity V_i in (7.18) is equal to $-\alpha\, \mathrm{d}E(\Gamma, V_i)$. By multiplying V_i by a constant, it can be made artificially preponderant in the evolution process. As a consequence, it seems appropriate to normalize the predefined veloci-

ties (unless otherwise imposed by some *a priori*). Actually, it suffices to impose that they all have the same norm. As a matter of fact, the PoPD velocities are such that

$$|V_i|^2 = \langle V_i, V_i \rangle \ = \frac{2l}{3} \tag{7.28}$$

if the edge length is constant and equal to l.

7.3.2 Interpretation as a projection

Individual projections and redundancy. The shape derivative (7.4) evaluated at V_i is equal to

$$\mathrm{d}E(\Gamma, V_i) := \langle -\Psi \ N \ , \ V_i \rangle \tag{7.29}$$

$$:= \langle G \ , \ V_i \rangle \tag{7.30}$$

$$= \ -\frac{1}{\gamma} \ \beta_i \tag{7.31}$$

where the last equality comes form (7.17). Therefore, the weights β_i can be interpreted as the projections of the opposite of the gradient G onto the V_i's (up to a multiplicative constant γ) or, otherwise stated, as the coordinates of $-G$ with respect to the set of directions formed by the V_i's. This raises the question of the orthogonality of the set of the V_i's.

One can check that the PoPD velocities do not form an orthogonal set relative to the $\mathcal{L}^2$ inner product

$$\langle V_i, V_j \rangle = \begin{cases} \frac{2l}{3} & \text{if } i = j \\ \frac{l}{6} \ N(\Gamma_i) \cdot N(\Gamma_j) & \text{if } |i - j| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{7.32}$$

where, as a reminder, $N(\Gamma_i) \cdot N(\Gamma_j)$ is equal to the cosine of the angle between $N(\Gamma_i)$ and $N(\Gamma_j)$. As a consequence, these predefined velocities are redundant.

Suppose that G is equal to V_i. The velocity V being constrained to a linear combination of the V_i's, it should logically have the form $-\alpha \ V_i$, $\alpha > 0$ in this case. If the predefined velocities form an orthogonal set, V is indeed proportional to $-V_i$

$$V = \sum_j \beta_j \ V_j \tag{7.33}$$

$$= \ -\gamma \sum_j \langle G \ , \ V_j \rangle \ V_j \tag{7.34}$$

$$= \ -\gamma \sum_j \langle V_i \ , \ V_j \rangle \ V_j \tag{7.35}$$

$$= \ -\gamma |V_i|^2 \ V_i, \tag{7.36}$$

as expected. Instead, the weights of the PoPD velocities V_{i-1}, V_i, and V_{i+1} are equal to (up to a multiplicative constant) $-N(\Gamma_{i-1}) \cdot N(\Gamma_i)$, -4, and $-N(\Gamma_i) \cdot N(\Gamma_{i+1})$, respectively. All three predefined velocities get weights different from zero (in general) with a higher weight for V_i in absolute value. If the dot products are positive (*i.e.*, if the (pseudo) curvature is not too high in absolute value), this can be interpreted as a smoothing of the response obtained in the case of an orthogonal set of predefined velocities, which could, as suggested in [Cha+05], increase the spatial coherence of the active contour evolution. Unfortunately, if the dot products are negative, V_{i-1} and V_{i+1} get positive weights, leading to an evolution where Γ_i is translated in a way that decreases the energy while the other two vertices are translated in a way that (although slightly) might tend to increase the energy.

As brought to our attention [Ano07], the redundancy could be suppressed by generating mutually orthogonal predefined velocities $V_i^\perp$ using the Schmidt orthonormalization process, or else G could be projected onto the space of linear combinations of the predefined velocities (instead of being projected individually onto each predefined velocity) as studied below.

Alternative linear combination. Instead of computing the weights β_i as individual projections (7.31), they could be computed such that V is the $\mathcal{L}^2$ projection of $-G$ onto the space of linear combinations of the V_i's

$$\beta = \arg\min_B \left| -G - \sum_i B_i \, V_i \right|^2 \tag{7.37}$$

$$= \arg\min_B \left| \sum_i B_i \, V_i \right|^2 + 2 \sum_i B_i \, \langle G, V_i \rangle \tag{7.38}$$

$$= \arg\min_B \sum_i B_i^2 \, |V_i|^2 + 2 \sum_{i<j} B_i B_j \, \langle V_i, V_j \rangle + 2 \sum_i B_i \, \langle G, V_i \rangle \; . \tag{7.39}$$

If the V_i's form an orthonormal set, then (7.39) is equivalent to

$$\beta = \arg\min_B |B|^2 + 2 \, B \cdot \eta \tag{7.40}$$

where η is the vector of components $\langle G, V_i \rangle$. For a given norm of β, the Cauchy-Schwarz inequality leads to the result based on individual projections $\beta_i \propto -\eta_i$, as expected.

For the PoPD velocities, if G is equal to V_i, then the projection of $-G$ is necessarily such that β_j is equal to zero if $j \notin \{i-1, i, i+1\}$.[#6] Let us fix the norm of β to one. Using (7.32) and the fact that $\beta_{i-1}^2 + \beta_i^2 + \beta_{i+1}^2$ is equal to

[#6] Otherwise, the support of V_j having no intersection with the support of V_i, the projection error (7.37) could be decreased by setting $\beta_j = 0$.

one, the projection error (7.37) can be developed as follows

$$\left| -V_i - \sum_i B_i\, V_i \right|^2 = \left| \beta_{i-1} V_{i-1} + (\beta_i + 1) V_i + \beta_{i+1} V_{i+1} \right|^2 \tag{7.41}$$

$$= \alpha(\beta_i + 1) \Big[4 + N(\Gamma_i) \cdot \Big(\beta_{i-1} N(\Gamma_{i-1}) + \beta_{i+1} N(\Gamma_{i+1}) \Big) \Big] \tag{7.42}$$

where α is a positive constant. The projection error is equal to zero if and only if $\beta_i = -1$, which implies that $\beta_{i-1} = \beta_{i+1} = 0$. Thus, the velocity V is equal to $-V_i$. Therefore, adopting this alternative projection approach allows riddance of redundancy.

7.3.3 Link with parametric approaches

Description. Another way to avoid the discretization flaw mentioned in Section 7.2.1 is to parameterize the active contour and rewrite the energy (7.2) as a function of these parameters [Jac+04, Mar05, Una+05]. Thus restricted to the set of domains whose boundary can be described by such parameters, the minimization of (7.2) becomes a classical problem in $\mathbb{R}^n$ where n is the number of parameters.

If ϕ in (7.2) does not depend on Γ, then the partial derivatives of (7.2) with respect to each parameter can be obtained by a calculus of variations [Jac+04]. However, it can be more complex if ϕ does depend on Γ, unless noticing that the partial derivatives can be expressed as shape derivatives [Mar05, Deb+06]. Let Γ be a curve described by a set of parameters $p = \{p_i,\ i \in [1..n]\}$, *e.g.*, a spline. The energy (7.2) can be rewritten as a function of p

$$E(p) = \int_{\Omega(p)} \phi(\Gamma(p), x)\ \mathrm{d}x + \int_{\Gamma(p)} \varphi(s)\ \mathrm{d}s\ . \tag{7.43}$$

It can be shown that the gradient of (7.43) is equal to

$$\nabla E(p) = \sum_i \mathrm{d}E \left(\Gamma, \frac{\partial \Gamma}{\partial p_i} \right)\ e_i \tag{7.44}$$

where $\partial\Gamma/\partial p_i$ is a so-called admissible velocity (see Fig. 7.5) and e_i is the i^{th} element of the canonical basis of $\mathbb{R}^n$ [Mar05, Deb+06]. The energy (7.43) can be minimized using the following procedure

$$\begin{cases} p^0 \\ p^{+1} = p - \alpha\ \nabla E(p) \end{cases} \tag{7.45}$$

where the optimal value for α can be computed as follows

$$\alpha = \arg\min_{a \geq 0}\ E(p - a\ \nabla E(p))\ . \tag{7.46}$$

Intuitively, it seems that the constrained approach (7.18)/(7.20) can be related to the parametric approach (7.45). The circumstances in which they are equivalent are detailed below.

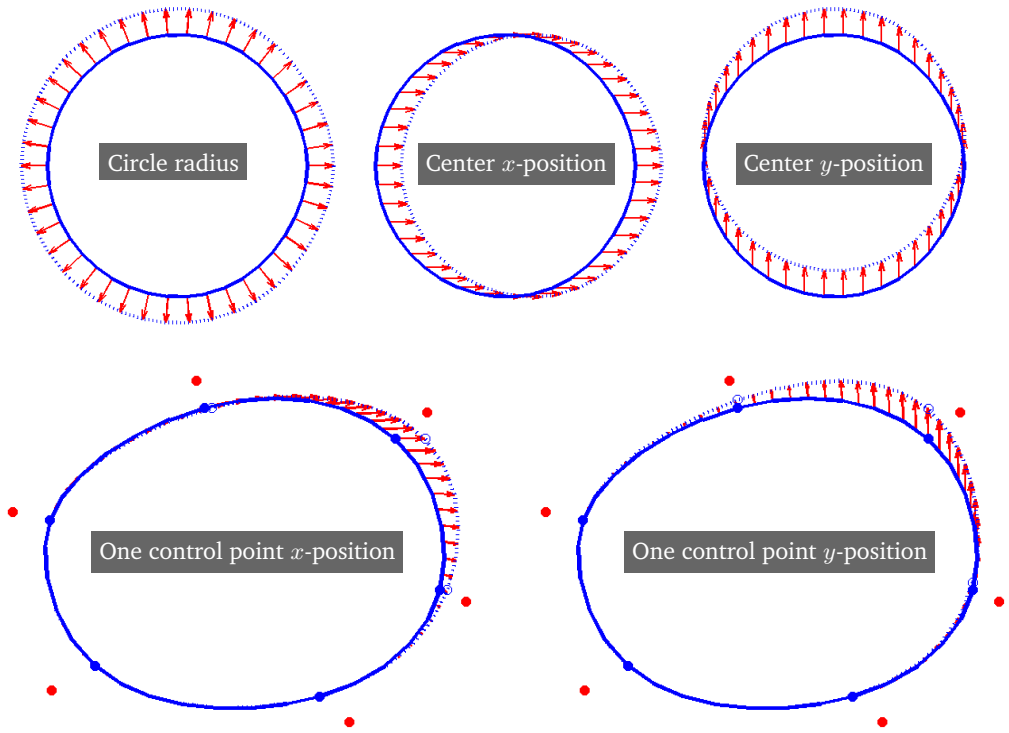

Figure 7.5 – Examples of admissible velocities represented discretely with red arrows. *(Upper row)* If the active contour is restricted to a circle, then the 3 parameters are the circle radius, and the horizontal and vertical position of its center, each of which giving rise to an admissible velocity. *(Lower row)* If the active contour is restricted to a spline with 6 control points (the red dots), then each control point gives rise to 2 admissible velocities, one for each coordinate. The 2 depicted velocities correspond to the upper-right control point.

Establishing a link. If the contour is represented by a polygon with n edges, then the parameters p involved in the parametric approach are simply the coordinates (a_i, b_i) of each vertex Γ_i

$$\begin{aligned} p &= \{p_i,\ i \in [1..2n]\} \\ &= \{\Gamma_i,\ i \in [1..n]\} \\ &= \{a_1, b_1, a_2, b_2 \dots a_n, b_n\}\ . \end{aligned} \tag{7.47}$$

Therefore, there are $2n$ admissible velocities such that

$$\begin{cases} \dfrac{\partial \Gamma}{\partial a_j}(\Gamma_i) = \delta_{ij}\ e_1 \\ \dfrac{\partial \Gamma}{\partial b_j}(\Gamma_i) = \delta_{ij}\ e_2 \end{cases}, \ i \in [1, n] \tag{7.48}$$

where δ_{ij} is equal to 1 if $i = j$ and 0 otherwise, and (e_1, e_2) is the canonical basis of $\mathbb{R}^2$. Writing the procedure (7.45) in terms of the vertices leads to

$$\Gamma_i^{+1} = \Gamma_i - \alpha \left[\mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial a_i}\right) \; e_1 + \mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial b_i}\right) \; e_2 \right] \tag{7.49}$$

$$= \Gamma_i - \alpha \left[\mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial a_i}\right) \; \frac{\partial \Gamma}{\partial a_i}(\Gamma_i) + \mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial b_i}\right) \; \frac{\partial \Gamma}{\partial b_i}(\Gamma_i) \right] \tag{7.50}$$

$$= \Gamma_i - \alpha \left[\sum_j \mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial a_j}\right) \; \frac{\partial \Gamma}{\partial a_j}(\Gamma_i) + \sum_j \mathrm{d}E\left(\Gamma, \frac{\partial \Gamma}{\partial b_j}\right) \; \frac{\partial \Gamma}{\partial b_j}(\Gamma_i) \right] \tag{7.51}$$

which corresponds to the discrete, constrained evolution (7.20) for the V_j's being the admissible velocities. Noting that $\partial\Gamma/\partial a_i$ and $\partial\Gamma/\partial b_i$ fit definition *(ii)* of the PoPD velocities and according to the equivalence between discrete and continuous evolutions shown in this case in Section 7.2.2, it can be deduced that the equivalence between (7.20) and (7.45) holds in the continuous framework. In conclusion, with the polygonal representation, if the predefined velocities are chosen equal to the admissible velocities, then the constrained approach is equivalent to the parametric approach.

A similar development with the control points of uniform cubic B-splines leads to the same conclusion for this smoother contour representation [Deb+07].

About redundancy. The admissible velocities are not necessarily mutually orthogonal. In particular, for the polygonal representation, they are not. Yet, these velocities arise in the computation of the gradient of the energy (7.43), a function from $\mathbb{R}^n$ to $\mathbb{R}$. In this context, the notion of gradient is classical. This might make think that redundancy (see Section 7.3.2) is not always an undesirable property.

7.4 Illustrative experiment

7.4.1 Direct vs. constrained approach

The different ways of using the shape derivative discussed above are summarized as basic segmentation algorithms in Tabs. 7.1, 7.2, and 7.3 for the polygonal representation.[#7] These algorithms should be considered as versions designed for testing. In particular, the contour resolution is fixed, which is not necessarily optimal. The constant $\alpha_{\min}$ is homogeneous to a number (possibly not an integer) of pixels. It is related to the achievable accuracy of the segmentation: the lower $\alpha_{\min}$, the better the accuracy.

Experimentally, the gain in accuracy of the constrained approaches (Tabs. 7.2 and 7.3) over the direct approach is not decisive. At low contour resolution (l

[#7] The parametric approach was considered for theoretical comparison only.

1. Choose an initial polygon $\Gamma = \{\Gamma_i, i \in [1..n]\}$ with an edge length equal to a given resolution l
2. Choose a threshold $\alpha_{\min}$
3. Compute Ψ (a function of Γ)
4. Compute the velocity V_i at Γ_i

$$V_i = -G(\Gamma_i) = \Psi(\Gamma_i)\ N(\Gamma_i) \tag{a}$$

5. Update Γ according to

$$\Gamma_i^{+1} = \Gamma_i + \alpha\ V_i \tag{b}$$

   where α is computed as $\arg\min_{a \geq 0}\ E(\Gamma + a\ V)$
6. If needed, resample Γ to approximately maintain a resolution of l
7. If α was less than $\alpha_{\min}$, then the algorithm is supposed to have converged; Otherwise go back to step 3.

Table 7.1 – Minimization of an energy E with the direct approach for the polygonal representation.

1.-3. See Tab. 7.1

4. Compute the PoPD velocities U_i according to definition *(i)*
5. Compute the shape derivatives $\mathrm{d}E(\Gamma, U_i)$
6. Compute the velocity V_i at Γ_i

$$V_i = -\mathrm{d}E(\Gamma, U_i)\ U_i(\Gamma_i) \tag{a}$$

7. See step 5 and after of Tab. 7.1.

Table 7.2 – Minimization of an energy E with the constrained approach/"individual projections" (see Section 7.3.2) for the polygonal representation.

1.-3. See Tab. 7.1

4. Compute the PoPD velocities U_i according to definition *(i)*

5. Compute the gradient $G = -\Psi\ N$

6. Compute the velocity V_i at Γ_i

$$V_i = \sum_j \beta_j\ U_j(\Gamma_i) \tag{a}$$

where $\beta = \arg\min_B \left| -G - \sum_i B_i\ U_i \right|^2$

7. See step 5 and after of Tab. 7.1.

Table 7.3 – Minimization of an energy E with the constrained approach/"projection on the set of linear combinations" (see Section 7.3.2) for the polygonal representation.

large), the segmentation qualities, measured in how low the energy E at convergence is, would rank in the following increasing order: algorithm of Tab. 7.1, algorithm of Tab. 7.2, and algorithm of Tab. 7.3. However, this potential advantage of the constrained approaches vanishes at high resolution since the error due to the discretization of the velocity in the direct approach gets negligible (see Fig. 7.2) [Deb+07]. Even in terms of computational cost, the benefit of working at low resolution with the constrained methods compared to working at high resolution with the direct method is far from obvious since steps 5 in Tab. 7.2 and 6 in Tab. 7.3 are rather demanding.

Yet, the constrained approaches have two interests: *(i)* for general-purpose predefined velocities such as the PoPD velocities – definition *(i)*, they provide coherence between the theory developed in the continuous framework and its discrete implementation, and *(ii)* they bring flexibility to the evolution process with the possibility of selecting application-driven predefined velocities such as the PoPD velocities – definition *(ii)*, as illustrated in Section 7.4.2.

7.4.2 An example of tracking constraint

An application where the possibility to introduce *a priori* knowledge in the evolution process can be useful is tracking. Indeed, a usual procedure is to use the segmentation of the object of interest in a frame as the initialization to segment the next frame. Then, the required contour deformation is clearly correlated to the

motion of the object.[#8] Consequently, motion-based predefined velocities (PoPD velocities - definition *(ii)*) appear appropriate. The standard test sequence "Football" was chosen to check the validity of this approach, the goal being to segment a player on several consecutive frames.

The choice of the energy is independent of the choice of the predefined velocities. Nevertheless, it is natural in tracking to use an energy which involves motion [Cre&Soa03]. Here, considering the complexity of the motion of the object of interest and the slight motion blur, it seemed more judicious to confine the use of motion to the predefined velocities and to select a motion-free energy able to account for the color variability of the object [Aub+03]

$$E(\Gamma) = \int_{\mathbb{R}^2} D\big(h(\Gamma, a), h_{\text{prev}}(a)\big)\, \mathrm{d}a \tag{7.52}$$

[#9]where h is a smooth, normalized version of the color histogram in Ω of the frame f_t to be segmented

$$h(\Gamma, a) = \frac{1}{|\Omega|} \int_{\Omega} g(f_t(x) - a)\, \mathrm{d}x \tag{7.53}$$

where $|\Omega|$ is a measure of Ω and g is a smoothing kernel, *e.g.*, a 2-dimensional Gaussian. Similarly, h_{prev} is a smooth, normalized version of the color histogram of the segmentation in the previously segmented frame f_{t-1}. Note that a and $f_t(x)$ should belong to $\mathbb{R}^3$. However, to limit the computation load, only the two most significant color components out of the three were considered. The function D is a positive function from $\mathbb{R}^2$ to $\mathbb{R}$ defined as $D(x, y) = (x - y)^2$. A maximal area constraint [Roy+06] was added to the energy (7.52) since its sensitivity decreases in the inner neighborhood of the correct segmentation

$$E_A(\Gamma) = -\delta \int_{\Omega} \mathrm{d}x \tag{7.54}$$

where δ is a weighting parameter. The shape derivative of the sum of (7.52)

[#8]Ideally, the motion could even transform the segmentation in a frame directly into the segmentation in the next frame. However, because the sequence is a two-dimensional projection of a three-dimensional scene and because the available motion is usually an apparent motion, this is not the case in practice.

[#9]Note that the energy (7.52) has not the same form as the domain integral in (7.2). As a matter of fact, integration over a domain appears only indirectly through $h(\Gamma, a)$ – see (7.53). Consequently, applying the results on shape derivative does not amount to simply identifying terms between this specific energy and the general expressions of the framework. On the other hand, it is not more complicated than combining classical rules of differentiation with these general expressions. However, the development leading to (7.55) being quite long, it is not presented here.

and (7.54) is

$$\begin{aligned} \mathrm{d}E(\Gamma, V) = -\int_{\Gamma} \Bigg[& \frac{1}{|\Omega|} \Big(g \star D'\big(h(\Gamma), h_{\mathrm{ref}}\big) \Big)\big(f_t(s)\big) \\ & - \frac{1}{|\Omega|^2} \int_{\mathbb{R}^2} h(\Gamma, a)\; D'\big(h(\Gamma, a), h_{\mathrm{ref}}(a)\big)\; \mathrm{d}a \\ & - \delta \Bigg] N(s) \cdot V(s)\; \mathrm{d}s \end{aligned} \tag{7.55}$$

where $\star$ is the convolution operator and D' is the partial derivative of D with respect to its first variable.

Two segmentations were computed, first by following strictly the algorithm described in Tab. 7.2 (to serve as a segmentation of reference relying on general-purpose, normal-based predefined velocities), second by replacing the step 4 with the PoPD velocities following definition *(ii)* $(m_i/|m_i|)$where the local motion at the vertices was estimated by a sub-optimal block matching technique [Zhu&Ma00] with a $1/4$-pixel precision. Here, a block was defined as the intersection between a 21×21-pixel square centered on a vertex and the mask of the current active contour. This prevented the pixels considered to be outside the object from interfering with the motion estimation. This procedure is given for illustrative purposes only. In a real-world tracking application, motion estimation should certainly be more sophisticated [Isa&Bla98, Aru+02, Rob&Mil03]. The resolution l was chosen equal to 10 pixels (to be compared to a size of frame of 352×288 pixels). The results are presented in Fig. 7.6. The normal-based segmentation is globally similar to the motion-based segmentation except for a small region wrongly included above the player's helmet. These segmentations correspond to two local minima, one of which being more relevant. In a way, the motion-based evolution *took the shortest path* toward the object of interest and converged toward a more satisfying minimizer. As a matter of fact, the convergence was reached after 10 iterations while it took 14 iterations for the normal-based evolution. An intuition of the behavior of the normal-based evolution is given by the normals shown in Fig. 7.6 compared to the local motions. When a normal has a direction close to the direction or opposite direction of the local motion,[#10] then both evolution processes behave similarly. However, when these directions are close to be orthogonal, then the normal-based process could have a tendency to evolve toward a less relevant local minimum and/or to require more iterations to converge. Indeed, the local motion is likely to be a better direction to take in a tracking application.

Other constrained evolutions can be designed by defining specific predefined velocities. For example, axial symmetry can be enforced [Deb+07].

[#10] If the direction of the normal is opposite to the direction of the local motion, the shape derivative takes equal values in absolute value for both but with opposite signs, making no difference as far as evolution is concerned.

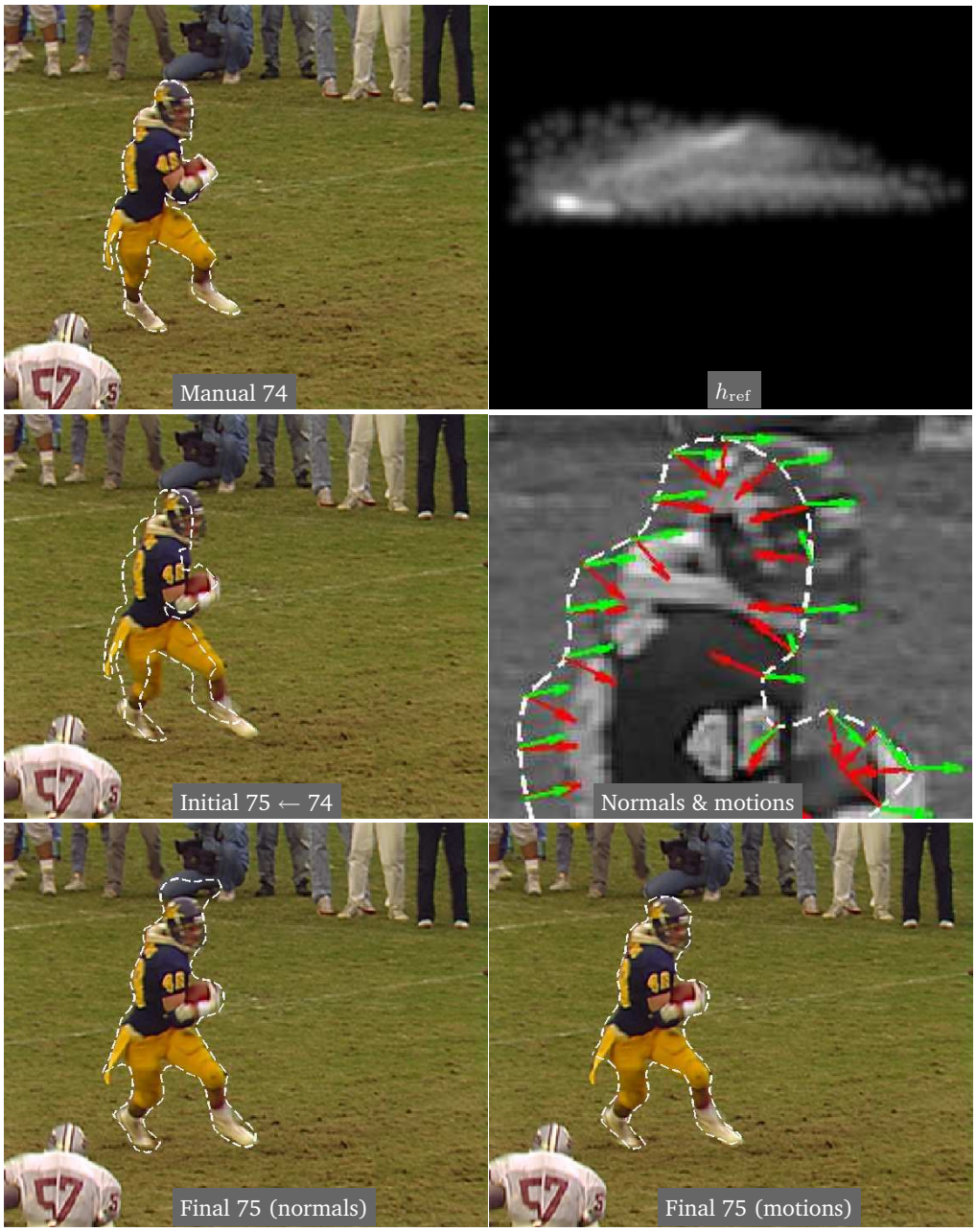

Figure 7.6 – Tracking of a player on frame 75 of the standard test sequence "Football" with the constrained approach. *Top left:* manual segmentation of frame 74 (used as the initial contour for the segmentation of frame 75); *Top right:* histogram h_{prev} of the manually segmented region (for display purposes, upper and lower zero-valued regions were cut out); *Middle left:* initial contour on frame 75 (copied from frame 74); *Middle right:* a close-up of the normals (red arrows) and the local motions (green arrows) computed at the vertices of the polygon at the first iteration; *Bottom left:* segmentation of frame 75 using normal-based predefined velocities; *Bottom right:* segmentation of frame 75 using motion-based predefined velocities.

7.5 Summary

The shape derivative is a convenient framework for deriving the evolution equation of an active contour from the energy to be minimized. Usually, the contour velocity is taken equal to the opposite of the energy gradient associated with the $\mathcal{L}^2$ inner product and then discretized. It induces an error responsible for a mismatch between the continuous evolution resulting from the theory and the discrete evolution implemented in practice. Although this has virtually no consequences if the contour is discretized finely enough, the constrained approach proposed to avoid this problem also gives more flexibility to the active contour process by allowing to introduce some *a priori* knowledge. This possibility of guiding the optimization procedure offers a way to compensate for the imperfection of the similarity (or dissimilarity) measure, as illustrated in Section 7.4.2 for tracking.

Chapter 8

Denoising

Context

As an alternative to pixel-based filtering, some denoising methods manipulate image patches. Indeed, it has been shown that there exist correlations among the patches forming natural images. The nonlocal means algorithm (NL-means) and the UINTA algorithm proved to be very efficient. However, these methods can be considered global since the filtering is performed identically everywhere in the image.

In this context, a locally adaptive denoising approach could represent a step forward (compared to other patch-level methods) similar to the one made in pixel-level denoising by edge-preserving filtering as opposed to isotropic filtering. The problem of minimizing the conditional entropy of a pixel color knowing its neighboring pixels can be revisited to this end. To begin with, a theoretically-founded motivation can be provided. Then, conveniently estimating the conditional entropy in the k nearest neighbor (kNN) framework offers the possibility to develop a locally adaptive kNN filtering method, thus adapting the smoothing to the nature of the regions. Moreover, with this approach, knowledge of the noise level is not required.

8.1 Patch-level processing

Justified by some studies on the distribution of patches forming natural images [Hua&Mum99, Lee+03, Sri+03, Car+08], patch-based processing methods have been proposed, *e.g.*, for image and video denoising [Bua+05a, Bua+05b, Awa&Whi06, Ker&Bou06, Ber+07, Bou+07, Dab+07], texture synthesis [Efr&Leu99, Efr&Fre01], and inpainting [Ber+03, Cri+04]. Indeed, these studies showed that there exist correlations among patches of natural images. As a consequence, the probability is high that patches similar to a given image patch are encountered in the image itself, offering the opportunity to recover unaltered or missing informa-

tion, should said given patch be degraded. As a matter of fact, the nonlocal means algorithm (NL-means) [Bua+05a, Bua+05b] and UINTA [Awa&Whi06] proved to be successful in image denoising.

In this context, *(i)* the problem of minimizing the conditional entropy of a pixel color knowing its neighboring pixels can be revisited, first by providing a proof of adequacy of this energy for image denoising. Then, a direct relation between the energy derivative and the mean shift can be established. As a consequence, it is possible to provide a variational interpretation of NL-means [Bua+05a, Bua+05b], thus linking an iterative algorithm such as UINTA [Awa&Whi06] to filtering methods. *(ii)* The aforementioned energy derivative can be approximated in the k nearest neighbor (kNN) framework. From a practical point of view, this allows to adapt to the local sample density and to reduce the effect of the curse of dimensionality when dealing with data of high dimension as it is inherently the case with patches. From a methodological point of view, it gives the opportunity to introduce local adaptability in the denoising process. For patch-based denoising, this improvement is of the same order as was, for pixel-based denoising, edge-preserving filtering [Per&Mal90, Cha+97] over isotropic filtering.

8.2 Neighborhood constrained denoising

8.2.1 Entropy-based energy

The inverse problem of image restoration can be formulated as a minimization problem. As mentioned in Section 8.1, natural images exhibit correlations among the patches which compose them. This correlation should be accounted for in deriving a restoration procedure.

Let h be the conditional entropy of patches, *i.e.*, the uncertainty on the color of a pixel when its neighborhood is known. Let X be a random variable modeling the color of the pixels of an image. Let D be the set of pixels of the image domain and let C be a structure of neighborhood of a pixel in D. The random vector $Y = \{X(t), t \in C\}$ represents the set of colors of the neighbors of a pixel. The random vector $Z = (X, Y)$ denotes the corresponding patch. A denoised version of a noisy image can be recovered by minimizing the entropy of X conditional on Y

$$X^* = \arg\min_X h(X|Y) \tag{8.1}$$

$$= \arg\min_X \int_y h(X|Y = y)\ f_y(y)\ \mathrm{d}y \tag{8.2}$$

$$= \arg\min_X E_Y[h(X|Y)] \tag{8.3}$$

where f_y is the probability density function (PDF) of Y. This PDF is unknown since it refers to noiseless neighborhoods. However, a sample $\tilde{y}_s$ of noisy neighborhood

can be extracted from the noisy observation at each pixel s of D. Therefore, the problem (8.3) is replaced with

$$X^* = \arg\min_X \frac{1}{|D|} \sum_{s \in D} h(X|Y = \tilde{y}_s), \tag{8.4}$$

which is, up to the fact that y_s is a noisy version of a neighborhood, an approximation of it. Actually,

$$\frac{1}{|D|} \sum_{s \in D} h(X|Y = \tilde{y}_s) \simeq h(X|\tilde{Y}) \tag{8.5}$$

where $\tilde{Y}$ denotes the random vector of noisy neighborhoods, so that the problem that will be (approximately) solved is rather

$$X^* = \arg\min_X h(X|\tilde{Y}) \ . \tag{8.6}$$

First of all, let us check that $h(X|Y)$ is, in theory, a suitable energy for denoising. Then, it will be verified that the practical alternative $h(X|\tilde{Y})$ is also valid.

8.2.2 Proof of adequacy

The conditional entropy of patches represents the uncertainty on the color X of a pixel when its neighborhood Y is known. Due to the spatial correlation between a pixel and its neighborhood, this conditional uncertainty is generally small in average. When adding noise to the image, some of the information carried by the neighborhood is lost, so that the uncertainty of a pixel knowing its neighborhood tends to be higher in average. This is formally stated by the following proposition.

Proposition 8.1. *Let X be a random variable and Y a random vector representing its neighborhood. Let $\tilde{X}$ be the sum of X and a white noise[#1] independent of X. Similarly, let $\tilde{Y}$ be a noisy neighborhood vector. Then,*

$$h(\tilde{X}|\tilde{Y}) \geq h(X|Y) \ . \tag{8.7}$$

Proof. By definition,

$$h(\tilde{X}|\tilde{Y}) = H(\tilde{X}) - I(\tilde{X};\tilde{Y}) \tag{8.8}$$

and

$$h(X|Y) = H(X) - I(X;Y) \tag{8.9}$$

[#1] The samples are assumed to be statistically independent.

where H denotes entropy and I denotes mutual information. The entropy $H(\tilde{X})$ is greater than $H(X)$ since the sum of two independent random variables increases the entropy [Cov&Tho91]. Then, it is sufficient to prove that $I(X;Y) \geq I(\tilde{X};\tilde{Y})$.

Let N and M be such that $\tilde{X} = X + N$ and $\tilde{Y} = Y + M$, both N and M being independent of X and Y. Then,

$$\begin{aligned} P(\tilde{Y}|X,Y) &= P(Y+M|X,Y) && (8.10)\\ &= P(M|X,Y) && (8.11)\\ &= P(M|Y) && (8.12)\\ &= P(Y+M|Y)\ . && (8.13) \end{aligned}$$

Therefore, $X \to Y \to \tilde{Y}$ forms a Markov chain. Thus, according to the data processing inequality [Cov&Tho91], we have

$$I(X;Y) \geq I(X;\tilde{Y}) = I(\tilde{Y};X)\ . \tag{8.14}$$

We can also write

$$\begin{aligned} P(\tilde{X}|X,\tilde{Y}) &= P(X+N|X,\tilde{Y}) && (8.15)\\ &= P(N|X,\tilde{Y}) && (8.16)\\ &= P(N|X) && (8.17)\\ &= P(X+N|X)\ . && (8.18) \end{aligned}$$

Therefore, $\tilde{Y} \to X \to \tilde{X}$ also forms a Markov chain. It can be concluded that

$$I(\tilde{Y};X) \geq I(\tilde{Y};\tilde{X}). \tag{8.19}$$

Finally, by combining (8.14) and (8.19), we have

$$I(X;Y) \geq I(\tilde{X};\tilde{Y})\ . \tag{8.20}$$

□

Proposition 8.1 supports the intuition that the minimization of the conditional entropy is an appropriate denoising approach. However, in practice, X must be recovered while the noiseless neighborhood Y is also unknown. $\tilde{Y}$ can be inferred from realizations of the observation $\tilde{X}$, though. Hence, an inequality involving $h(X|\tilde{Y})$ would better justify an algorithm based on conditional entropy.

Proposition 8.2. *The conditions are the same as in Prop. 8.1. Then,*

$$h(\tilde{X}|\tilde{Y}) \geq h(X|\tilde{Y})\ . \tag{8.21}$$

Proof. Inequality (8.19) can be developed as follows

$$I(\tilde{Y};X) \geq I(\tilde{Y};\tilde{X}) \tag{8.22}$$
$$\Leftrightarrow H(X) - h(X|\tilde{Y}) \geq H(\tilde{X}) - h(\tilde{X}|\tilde{Y}) \tag{8.23}$$
$$\Leftrightarrow h(\tilde{X}|\tilde{Y}) \geq h(X|\tilde{Y}) + \underbrace{H(\tilde{X}) - H(X)}_{\text{Positif}} \tag{8.24}$$
$$\Leftrightarrow h(\tilde{X}|\tilde{Y}) \geq h(X|\tilde{Y})\ . \tag{8.25}$$

□

As a consequence, the random variable X associated with the noiseless image is also a minimizer of the conditional entropy when the noisy neighborhood $\tilde{Y}$ is known.

Figure 8.1 illustrates for several images and noise levels the behavior of the conditional entropy before and after denoising[#2] with respect to the lower bound $h(X|\tilde{Y})$. As expected, the conditional entropy $h(\tilde{X}|\tilde{Y})$ of the noisy image is greater than or equal to the conditional entropy $h(X^*|\tilde{Y})$ of the denoised image for all noise levels. However, the conditional entropy of the denoised image is occasionally lower than the theoretic lower bound. This is explained by the fact that rapidly varying textures are interpreted as noise and therefore partially degraded by the denoising process. It is clearly noticeable when no noise is added to the images "Aerial" and "Baboon": the denoising algorithm does not leave the image unchanged, causing the conditional entropy to fall below the lower bound. This is not a caveat specific to the proposed method but rather an inevitable behavior of denoising algorithms. Fortunately, such a behavior, when present with the proposed method, seems to become less pronounced at high noise levels.

8.2.3 Energy derivative

Classically, one can use a gradient descent procedure to solve the minimization problem (8.6). As a consequence, the derivative of the conditional entropy of the color of a pixel knowing its neighborhood must be determined.

Considering (8.5), it can be shown that the derivative of $h(X|\tilde{Y})$ with respect to the pixel color x_s can be approximated as follows (see Appendix D)

$$\frac{\partial h(X|\tilde{Y})}{\partial x_s}(x_s) \simeq -\frac{1}{|D|}\frac{\nabla f_z}{f_z}(z_s) \cdot \frac{\partial z_s}{\partial x_s} \tag{8.26}$$

where z_s is equal to $(x_s, \tilde{y}_s)$ with $\tilde{y}_s$ being the observed noisy neighborhood of the observed noisy pixel $\tilde{x}_s$. Thus, this derivative can be estimated by a mean shift term (see Section 2.2.4) in the high dimensional space of Z multiplied by a projection term.

[#2] The denoising method is described in the subsequent sections. However, some results are already used here without further details for illustration.

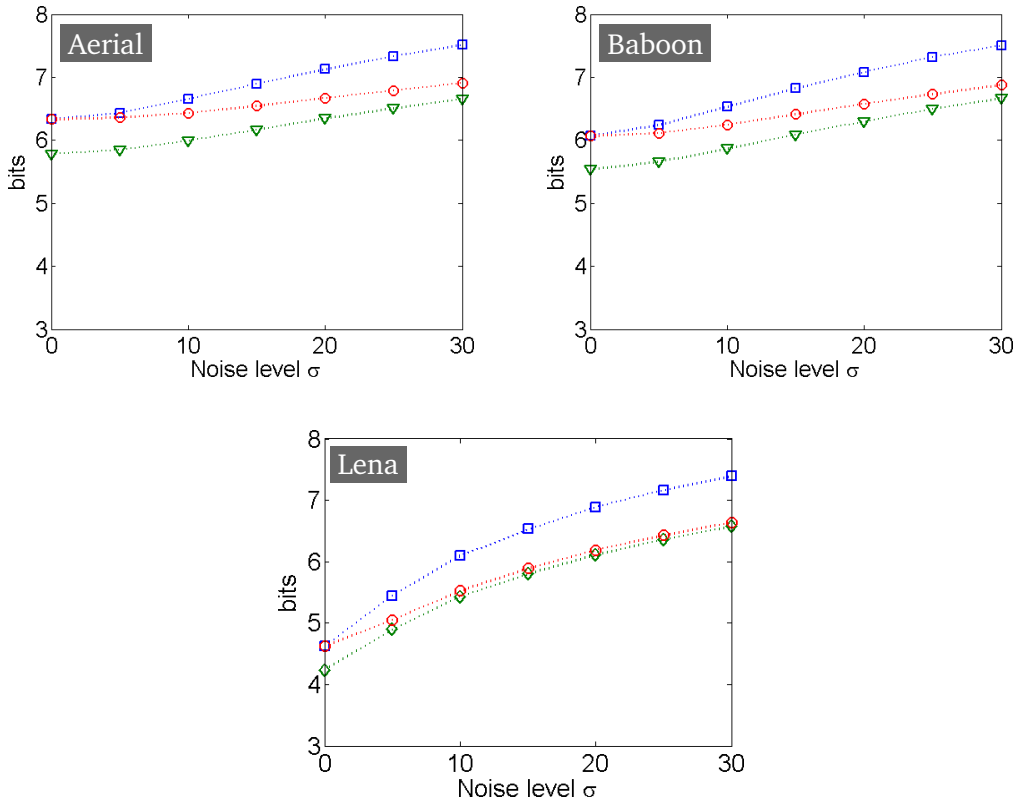

Figure 8.1 – Behavior of the conditional entropy before and after denoising. □ — Conditional entropy $h(\tilde{X}|\tilde{Y})$ of the noisy image, ▽ — Conditional entropy $h(X^*|\tilde{Y})$ after denoising with the proposed method, ○ — The lower bound $h(X|\tilde{Y})$ (see the comments in the body of text concerning the apparent contradiction). The images "Aerial", "Baboon", and "Lena" can be seen in following figures.

8.3 Toward locally adaptive kNN

The purpose is to develop a locally adaptive kNN-based approach. To this end, part of the reasoning will be made in the global case and then further extended to involve adaptability.

8.3.1 Global approach using kNN

Description. Let $\{z\}$ denote a set of patch samples of $\mathbb{R}^d$ drawn according to the PDF f_z. Let z be a point of $\mathbb{R}^d$. In the kNN framework, the mean shift approximation

at z is given by [Fuk&Hos75]

$$\frac{\nabla f_z}{f_z}(z) \simeq \frac{d+2}{\rho_k^2} \left(\frac{1}{k} \sum_{\substack{z_{t_i} \in \{z\} \\ |z_{t_i} - z| \leq \rho_k}} z_{t_i} - z \right) \tag{8.27}$$

where ρ_k is a short notation for the distance from z to its k-th nearest neighbor in $\{z\}$. In the expression (8.27), all the neighbors z_{t_i}, $i \in [1..k]$, are equally weighted by $1/k$. In classical pixel-based filtering, the counterpart is a spatial filter of size $\sqrt{k} \times \sqrt{k}$ with all the coefficients equal to $1/k$. Of course, smoothly decaying filters are usually preferred to this rectangular function. Such filters assign lower weights to pixels spatially faraway from the filter center. It seems natural to modify (8.27) in a similar manner, the spatial pixel distance being replaced with the distance between patches

$$\frac{\nabla f_z}{f_z}(z) \simeq \frac{d+2}{\rho_k^2} \left(\frac{1}{\sum_{j=1}^k w_{t_i}} \sum_{\substack{z_{t_i} \in \{z\} \\ |z_{t_i} - z| \leq \rho_k}} w_{t_i} z_{t_i} - z \right) . \tag{8.28}$$

The following weights have been proposed in NL-means [Bua+05a, Bua+05b]

$$w_{t_i} = \exp\left(-\frac{|z_{t_i} - z|^2}{\alpha} \right) = \exp\left(-\frac{\rho_i^2}{\alpha} \right) \tag{8.29}$$

where α is a positive constant chosen according to the standard deviation σ of the noise. When the noise level is not or imprecisely known, the algorithm performances are not optimal. Thanks to the kNN point of view, a reasonable estimation of α can be proposed – see below.

kNN noise estimation. As was pointed out [Bua+05a, Bua+05b], the Euclidean distance maintains, in expectation, the same order of similarity between patches before and after addition of noise. Formally, it can be shown that

$$E\left[|\tilde{Z}_{t_i} - \tilde{Z}_{t_k}|^2 \right] = |Z_{t_i} - Z_{t_k}|^2 + 2\sigma^2 \tag{8.30}$$

where the norm $|z|$ must be understood as $(z^\mathsf{T}\ m\ z)^{0.5}$, m being a weighting kernel whose coefficients sum to 1, and σ is the standard deviation of the noise. If we assume that, for any given patch z_{t_i}, there exist at least k patches similar enough to z_{t_i} in the image, then $|z_{t_i} - z_k|^2$ is negligible. Therefore, the left-hand side

of (8.30) represents a good estimation of $2\sigma^2$. This left-hand side term can then be approximated in the kNN framework with $\bar{\rho}_k^2$, the average value of ρ_k^2 for all the patches of the image. Then,

$$\sigma^2 \approx \frac{\bar{\rho}_k^2}{2} \,. \tag{8.31}$$

The estimation (8.31) seems accurate enough and the proposed denoising method appears to be stable with respect to the parameter k: for several noise levels and 3 noise level estimations (the actual standard deviation, $\sqrt{(\bar{\rho}_1^2)}$, and (8.31)), Fig. 8.2 shows the root mean squared error (RMSE) and the structural similarity index (SSIM) of image "Lena" after denoising as a function of k. Even though (8.31) seems reliable, it must be kept in mind that this is certainly an overestimation since $|Z_{t_i} - Z_{t_k}|^2$ has been neglected. Therefore, the proposed expression for α is

$$\alpha = \beta \, \bar{\rho}_k^2 \tag{8.32}$$

where β can be tuned within the interval $[0, 1/2]$ for optimal performances.

The weights w_{t_i} are a function of the distance between z_{t_i} and z. They correspond to the sampling of a predefined, continuous, univariate, real-valued weighting function w at abscissa $\rho_i = |z_{t_i} - z|, i \in [1..k]$. As long as this function does not explicitly depend on z or z_{t_i}, the denoising procedure remains global – see Section 8.4.3. One might want to introduce local adaptability to better preserve certain image structures.

8.3.2 Adaptively Weighted kNN (AWkNN)

Let us analyze the behavior of (8.29). Assume z belongs to the sample set $\{z\}$. If the k nearest neighbors of z are not very similar to it, then the distances $|z_{t_i} - z|^2$ are large and the corresponding w_{t_i}'s are low, except for z itself. Then, the weighted patch average is very close to z and the derivative (8.28) is almost equal to zero, which is supposed to mean that convergence has been reached. This situation could mainly arise for two reasons: *(i)* the image does not contain enough samples of the texture represented by z or *(ii)* the noise level is significant. In other words, the noise might be high and yet denoising will not occur.[#3] This sounds counter intuitive.

On the other hand, if the k nearest neighbors of z are very similar to it, then the distances $|z_{t_i} - z|^2$ are small and the corresponding w_{t_i}'s are large and close to the weight of z itself. Therefore, the k neighbors will all have an influence comparable to the one of z in the weighted patch average. This situation could mainly arise when most of the noise was gotten rid of. In short, noise is likely to be negligible

[#3] Naturally, if case *(i)* is true, then denoising would cause a degradation of the image but, as already mentioned, this is inevitable.

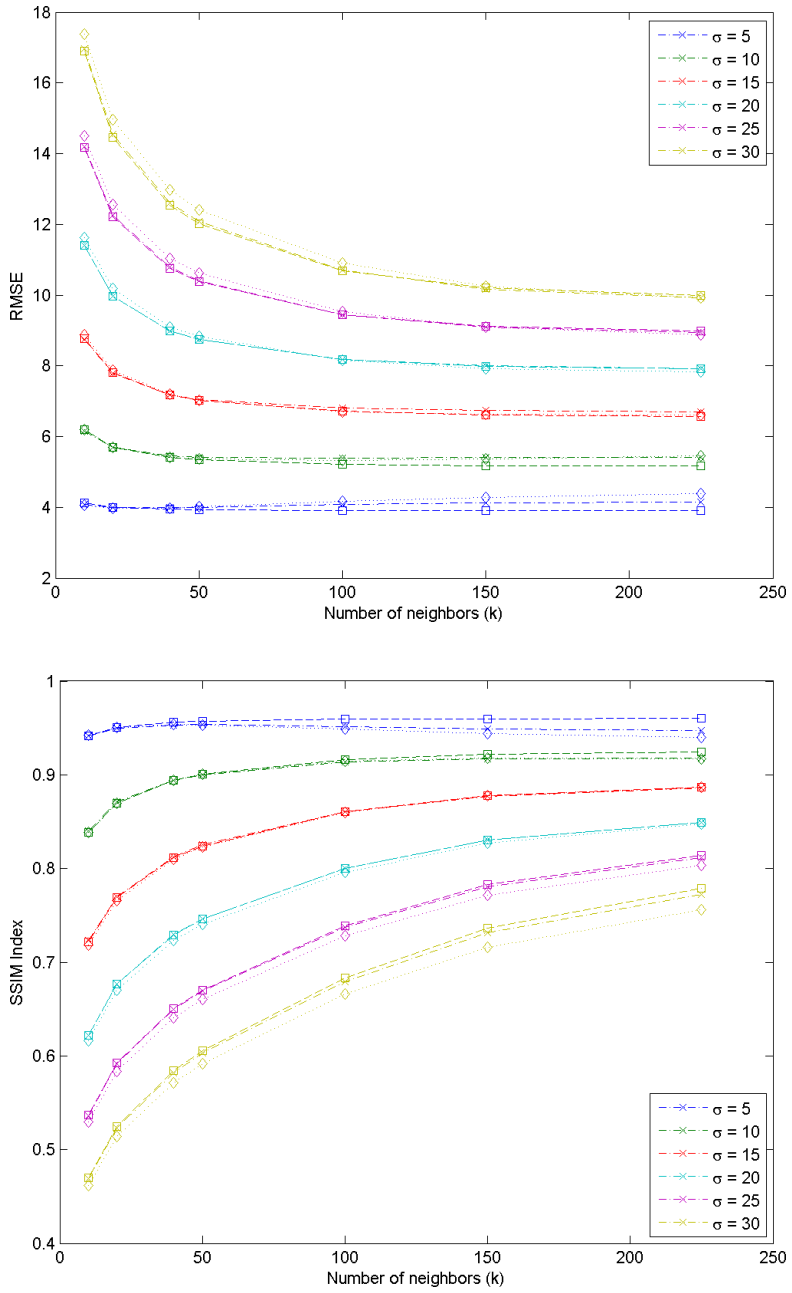

Figure 8.2 – RMSE and SSIM index of image "Lena" after denoising as a function of k for several noise levels. The estimated noise level $\hat{\sigma}$ was taken equal to either the actual value σ *(curves with square markers –□–)*, $\sqrt{(\bar{\rho}_1^2)}$ *(curves with cross markers –×–)*, or $\sqrt{(\bar{\rho}_k^2)}/\sqrt{2}$ *(curves with diamond markers –◇–)*.

and yet denoising will occur. This sounds unnecessary and could possibly induce a slight oversmoothing.

The idea is then to enforce the opposite behavior by involving ρ_k^2 in the weights instead of $\bar{\rho}_k^2$, replacing w_{t_i} in (8.28) with

$$a_{t_i} = \exp\left(-\frac{\rho_i^2}{\beta\ \rho_k^2}\right)\ . \tag{8.33}$$

While the corresponding continuous weighting function a is, just like w, evaluated at $\rho_i, i \in [1..k]$, it depends on ρ_k which itself depends (implicitly) on z whereas $\bar{\rho}_k^2$ involved in the expression of w is a pre-computed constant. Hence, (8.33) clearly brings local adaptability. Let us check precisely how. The weight a_{t_i} is maximum for z and minimum equal to $\exp(-1/\beta)$ for the k-th nearest neighbor of z, whether it is close to or faraway from z in terms of patch distance. Therefore, whenever the noise level is high, the following happens: ρ_k^2 is large, the weighting function a has a large bandwidth, and several neighbors will get involved in the denoising process with significant weights. Conversely, if the noise is negligible, then ρ_k^2 is small, the weighting function a has a reduced bandwidth, and only those neighbors very close to z will get significantly involved in the averaging, avoiding unnecessary smoothing.

8.4 Denoising method

8.4.1 Synthesis of the previous developments

The proposed denoising method relies on minimizing an energy defined as the conditional entropy h of the color of pixels knowing their neighborhood. The derivative (8.26) combined with (8.28) was determined to solve this problem with a gradient descent. The weights w_{t_i} in (8.28) were replaced with (8.33) to introduce local adaptability. Note that for comparison purposes, the global weights (8.29) will also be used in some experiments.

The main computations in the implementation of the method are searches for k nearest neighbors in the space of patches. As this is costly, these searches can be restricted to a search area centered around the patch of interest. Table 8.1 presents the steps of the denoising algorithm.

Let us make some remarks about this code. First, the norm in $|z_s - z_t|$ should be understood in a broad sense. And indeed, we did not use the $\mathcal{L}^2$ norm but a weighted version $|z_s - z_t| = [(z_s - z_t)^\mathsf{T}\ m\ (z_s - z_t)]^{0.5}$ where a^T denotes the transpose of a and m is a kernel employed to give relatively less importance to the pixels close to the edges of a patch.

Second, a discrete gradient descent normally writes as

$$x_s^* \leftarrow x_s^* + \mathrm{d}\tau\ (-\nabla_s) \tag{8.34}$$

1. Let $\tilde{x}$ be the noisy image
2. Initialization: $x^* \leftarrow \tilde{x}$
3. Temporary copy: $x \leftarrow x^*$
4. For each pixel s of D
   - Let z_s be the patch of radius r formed by
     the pixel color x_s^* and
     the neighborhood $\tilde{y}_s = \{\tilde{x}_t, t \in C_s\}$
     where C_s is the neighborhood of radius r of the pixel s
   - Let $A(s)$ be the search area of radius w centered at s
   - For each pixel $t \in A(s)$
   $$\rho(s,t) \leftarrow |z_s - z_t| \tag{a}$$
   - Select the k nearest patches z_t, *i.e.*, the t_i's such that
   $$0 = \rho(s, t_1 = s) \leq \rho(s, t_2) \leq \cdots \leq \rho(s, t_k) \tag{b}$$
   - For each patch $z_{t_i}, i \in [1..k]$
   $$a_{t_i} \leftarrow \exp\left(-\frac{\rho^2(s,t_i)}{\beta\ \rho^2(s,t_k)}\right) \tag{c}$$
   - Perform the following update
   $$x_s^* \leftarrow \frac{1}{\sum_{i=1}^k a_{t_i}} \sum_{i=1}^k a_{t_i}\ x_{t_i} \tag{d}$$
5. If x^* did not change significantly during this procedure, then the algorithm is supposed to have converged and x^* is the denoised image.
   Otherwise, go back to step 3.

Table 8.1 – Pseudocode of the proposed denoising algorithm.

where, here, the derivative ∇ is such that

$$\nabla_s = -\frac{1}{|D|}\frac{d+2}{\rho_k^2}\left(\frac{1}{\sum_{i=1}^k a_{t_i}}\sum_{i=1}^k a_{t_i} z_{t_i} - z_s\right)\cdot v \tag{8.35}$$

$$= -\frac{1}{|D|}\frac{d+2}{\rho_k^2}\left(\frac{1}{\sum_{i=1}^k a_{t_i}}\sum_{i=1}^k a_{t_i} x^*_{t_i} - x^*_s\right) \tag{8.36}$$

where v is a vector of projection whose components are all zeros expect the component corresponding to the pixel x^*_s which has a value of 1. It can be deduced that Tab. 8.1-(d) corresponds to choosing

$$\mathrm{d}\tau = \rho_k^2 |D| \;/\; (d+2)\;. \tag{8.37}$$

Third, note that Tab. 8.1-(d) explicitly writes as a filtering process. This was expected given the established relationship between NL-means and neighborhood/bilateral filtering [Tom&Man98, Bua+05b]. Equivalently, one can recognize the expression of a barycenter, again in accordance with the known interpretation of bilateral filtering as a (*restricted*) mean shift procedure [Bar&Com04].

Finally, it has been noted that the weight a_1, being always equal to one, gives too much influence to $x_{t_1=s}$ relatively to the other neighbors [Bua+05a, Bua+05b]. Therefore, the common, *heuristical* solution has been adopted. It consists in replacing a_{t_1} with $\max_{i\in[2..k]} a_{t_i} = a_{t_2}$ after the computations Tab. 8.1-(c).

8.4.2 Some remarks about NL-means and UINTA

Variational interpretation of NL-means. In (8.33), let us replace $\beta\,\rho_k^2$ with σ^2 where σ is the standard deviation of the noise. This modification inhibits local adaptability. In Tab. 8.1-(d), let us replace the weighted sum over the k nearest neighbors with the weighted sum over all the patches of the search area $A(s)$. Finally, in Tab. 8.1, instead of performing iterations until convergence, stop the algorithm at the end of the first iteration. The resulting algorithm is an implementation of NL-means [Bua+05a, Bua+05b]. Since the proposed method minimizes the conditional entropy h of the color of a pixel knowing its neighborhood, this observation provides a statistically founded, variational interpretation to NL-means as one step toward the minimization of h without local adaptability.

A deterministic, variational interpretation of NL-means as one step of a fixed-point iteration has been shown, up to a slight modification of the kernel, in the scope of optimization of nonlocal functionals [Kin+05]. Following a similar approach, an analogy can also be made between NL-means and an iteration of the Jacobi method [Gil&Osh07]. Additionally, a connection between neighborhood filters and diffusion Partial Differential Equations (PDEs) was established [Bua+06].

Highlight on some differences with UINTA. The proposed method has the same starting point as UINTA [Awa&Whi06]: the variational formulation (8.1). From there, mainly three differences distinguish the two developments and the resulting methods. First, the kNN framework was preferred here to the classical Parzen windowing in order to automatically adapt to the local sample density. Second, our approach introduces local adaptability into patch-based denoising.

The third significant difference, which has crucial consequences, is the interpretation of the energy derivative. In [Awa&Whi06], similarly to (8.26), it involves a patch-based derivative and a projection term onto a pixel. In implementing the gradient descent, UINTA alternately updates the pixels x and the neighborhoods y. Actually, the problem (8.1) would ideally be solved analytically as stated: find the values of the pixels which minimize the conditional entropy h knowing their (fixed) neighborhoods. For this reason, an iterative scheme should only update the pixels according to the projected derivative,[#4] working with mixed noisy/denoised patches z where the pixels are taken in the current estimation x^* and the neighborhoods are kept equal to the original noisy ones extracted in the noisy image $\tilde{x}$ – see Tab. 8.1. This has two consequences. First, since the proposed algorithm maintains a fixed reference, it converges in a few iterations instead of drifting away from the observation. Second, if one lets the UINTA algorithm iterate, geometrical artifacts appear in the denoised image. The conditional entropy decreases when more and more similar patches are encountered. Intuitively, one feels that updating the neighborhoods will lead to the creation of repeated patterns which exaggerate some structures of the original image. This phenomenon has been observed [Bro&Cre07].

8.4.3 Introducing local adaptability into feature-based denoising

A simplistic chronology of denoising approaches could be: *(i)* global pixel-level denoising, *(ii)* locally adaptive pixel-level denoising, *(iii)* global patch-level denoising, and *(iv)* locally adaptive patch-level denoising.

(i) By pixel-level, we refer to classical filtering where the weight of a pixel only depends on its distance to the pixel of interest. Such a filter can be, *e.g.*, a Gaussian whose argument is a real number representing a spatial distance between two pixels. By global, we mean that the Gaussian has a fixed bandwidth. Therefore, anywhere in the image, the filtering is the same.

(ii) By locally adaptive pixel-level denoising, we refer to edge-preserving methods [Per&Mal90, Cha+97]. Sticking with the Gaussian example, its bandwidth is made variable and dependent of the local color information. Typically, if local color variability is high, the bandwidth is reduced to avoid damaging this abrupt variation, or edge.

(iii) By patch-level, we refer to the present context with methods such as NL-means and UINTA. The argument of the Gaussian is still a real number but it now

[#4] Projection from the space of Z onto the space of X.

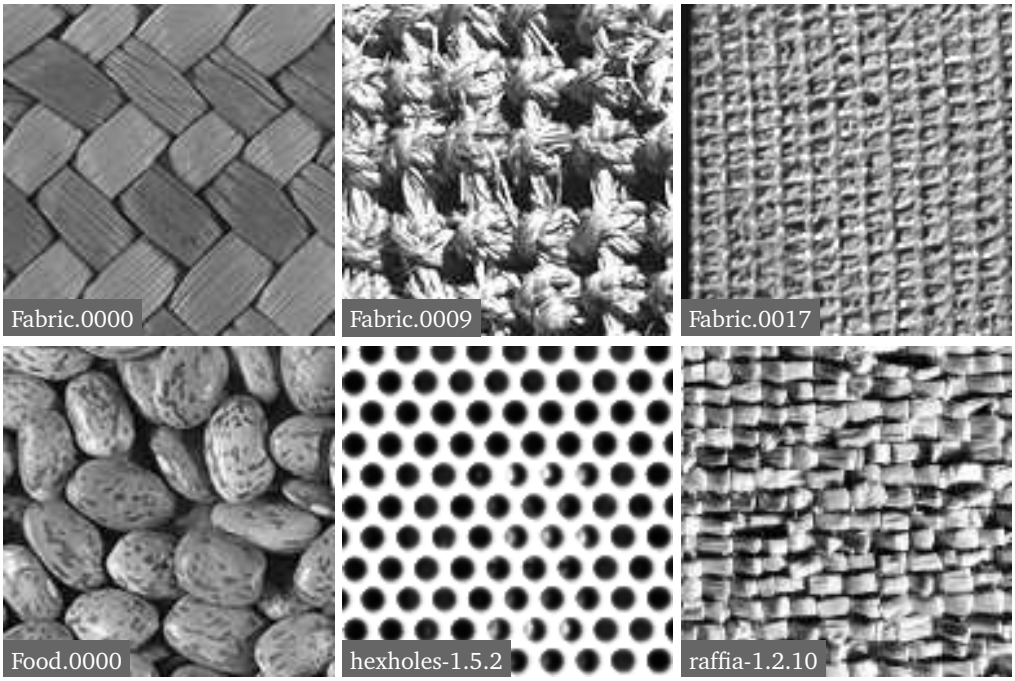

Figure 8.3 – Six of the 40 textures used for denoising tests.

represents a color distance between two patches, wherever they are in the image. Yet, as long as the Gaussian bandwidth is fixed, the method is considered global.

(iv) The proposed method introduces local adaptability by letting the Gaussian bandwidth depend on the variability among the k nearest patches. This follows the same philosophy as *(ii)* except that the considered patches are visual neighbors as opposed to spatial neighbors. Of course, patch-based methods can be thought of as a particular case of feature-based methods. What is valid for patches and the Euclidean distance should also be valid for other features and their associated metric.

8.5 Illustrative experiments

The authors of NL-means [Bua+05a, Bua+05b] made public an implementation of their algorithm. This implementation takes grayscale images as input. Therefore, in the following (brief) comparison, only grayscale images will be presented. A result of inpainting of a color image, whose algorithm relies on the same point of view as the present one, will be shown in Section 10.1.

The performances of NL-means and the proposed method, referred to as AW-kNN, were compared on a set of 40 textures part of a larger database publicly available [MIT02]. Six of them are shown in Fig. 8.3. Before processing, the images

	NL-means				AWkNN					
Texture	w	r	h	*RMSE*	*RMSE*	w	r	k	β	*Its.*
Fabric.0000	4	5	0.8	**10.26**	11.43	4	4	20	0.3	2
Fabric.0004	7	2	0.8	**12.08**	14.61	10	4	20	0.3	2
Fabric.0007	15	2	0.8	**12.32**	12.93	15	5	20	0.3	2
Fabric.0009	15	2	0.8	19.99	**19.95**	4	2	20	0.3	1
Fabric.0015	15	2	0.8	**13.40**	13.48	15	2	40	0.3	2
Fabric.0017	15	2	0.8	**17.26**	17.89	15	2	20	0.3	2
Fabric.0018	15	2	0.8	**19.06**	19.62	15	2	20	0.3	2
Food.0000	4	4	0.8	**14.05**	14.43	15	2	20	0.3	2
Food.0005	15	2	0.8	**17.04**	17.08	15	2	20	0.3	2
Leaves.0003	4	5	1	**14.15**	14.61	15	2	20	0.3	2
Leaves.0012	15	2	1	**15.62**	15.98	15	2	20	0.3	2
Leaves.0013	15	5	0.8	10.01	**9.75**	15	5	20	0.3	2
Metal.0000	15	2	0.8	20.92	**19.95**	4	2	20	0.3	1
Metal.0002	15	2	0.8	**19.61**	19.95	4	2	20	0.3	1
Misc.0000	15	2	1	**12.14**	13.37	15	2	20	0.3	2
Misc.0002	4	5	0.8	**9.79**	10.09	4	5	20	0.3	2
Stone.0005	15	2	0.6	**14.76**	15.20	15	2	40	0.3	2
Water.0001	7	5	1	**6.20**	6.29	15	5	60	0.3	3
Water.0006	4	5	0.8	7.83	**7.76**	15	3	40	0.3	3

Table 8.2 – Comparison of NL-means and AWkNN on a set of textures. The parameter w is the radius of the search window A, r is the radius of the patches z, h is expressed in fractions of the standard deviation of the noise σ (the recommended value for h is σ [Bua+05a, Bua+05b]), and $Its.$ is the optimal number of iterations of AWkNN – Table continued in Tab. 8.3.

were cropped to 64×64. A Gaussian noise with a mean equal to zero and a standard deviation σ equal to 20 was added. For each image, Tabs. 8.2 and 8.3 present the best result of NL-means and AWkNN in terms of RMSE among the experiments performed for all the combinations of parameters (see Tab. 8.2 for the notations) taken in the sets:

- $w \in \{4, 7, 10, 15\}$,
- $r \in \{2, 3, 4, 5\}$,
- $h/\sigma \in \{0.6, 0.8, 1.0, 1.2\}$,
- $k \in \{20, 40, 60, 80\}$, and
- $\beta \in \{0.3, 0.4, 0.5, 0.6\}$.

It should be noted that AWkNN usually reaches the lowest RMSE before convergence. The textures are probably degraded when iterating too much. The iteration

	NL-means				AWkNN					
Texture	w	r	h	*RMSE*	*RMSE*	w	r	k	β	*Its.*
beachsand-1.2.7	15	2	0.8	20.71	**19.95**	4	2	20	0.3	1
calfleath-1.1.6	15	2	0.8	14.64	**14.59**	15	2	40	0.3	2
calfleath-1.2.6	15	2	1	**19.83**	19.95	4	2	20	0.3	1
grass-1.1.1	15	2	0.8	**17.99**	18.82	15	2	20	0.3	2
grass-1.5.7	15	2	0.8	11.89	**11.75**	4	5	20	0.3	2
gravel-1.5.5	4	5	1	**8.09**	9.09	4	5	40	0.3	2
hexholes-1.5.2	15	2	1.2	**9.16**	9.91	15	2	40	0.3	2
image38	15	2	0.8	**12.42**	12.67	15	2	40	0.4	2
pigskin-1.1.11	7	2	0.8	**11.57**	11.61	4	5	20	0.3	2
pigskin-1.2.11	15	2	0.8	20.88	**19.95**	4	2	20	0.3	1
plasticbubs-1.1.13	4	5	0.8	**11.18**	11.69	7	5	20	0.3	2
raffia-1.1.10	15	5	0.6	11.45	11.45	15	5	20	0.3	2
raffia-1.2.10	15	2	1	**16.65**	17.76	15	2	20	0.3	2
roughwall-1.5.3	4	5	0.8	**10.22**	10.85	4	5	20	0.3	2
sand-1.5.4	15	3	0.6	**12.36**	13.38	15	5	20	0.3	2
water-1.1.8	15	5	0.6	8.99	**8.69**	15	5	20	0.3	3
water-1.2.8	15	2	1	**17.91**	18.70	15	2	20	0.3	2
woodgrain-1.1.9	15	5	0.6	9.38	**9.33**	10	5	40	0.3	2
woodgrain-1.2.9	15	4	0.8	**15.07**	15.59	15	2	20	0.3	2
woolencloth-1.1.5	15	4	0.6	**12.23**	12.24	15	3	80	0.3	2
woolencloth-1.2.5	15	2	0.8	21.24	**19.95**	4	2	20	0.3	1
RMSE: $\mu_r \pm \sigma_r$				14 ± 4.2	14.3 ± 4.0					

Table 8.3 – Continuation of Tab. 8.2. The last row shows the mean μ_r and standard deviation σ_r of the RMSEs for each method.

numbers in Tabs. 8.2 and 8.3 reflect this condition of optimality rather than the convergence.

In light of these results, it appears that both methods performed equally well on these texture images. Therefore, as is, the local adaptability does not bring much. Some alternatives to (8.33) should be envisioned in order to improve AWkNN.

Another series of experiments was conducted with the images "Aerial", "Baboon", and "Lena". The experimental setup was identical to the one for the textures. The results are presented in Figs. 8.4, 8.5, and 8.6, and some quantitative measures are given in Tab. 8.4. Although, NL-means achieved lower optimal RMSEs with all three images, the difference with AWkNN is truly significant on "Lena" only. On average over all the parameter combinations, AWkNN performed better on "Aerial" and "Baboon". This tendency, that could already be guessed with the experiments on textures, indicates that AWkNN might be a bit more stable with

Figure 8.4 – Denoising of the image "Aerial". The result produced by AWkNN seems more natural.

respect to the choice of the parameters. Naturally, a deeper study is necessary before possibly drawing any conclusion. Regarding the parameters, AWkNN has two more than NL-means: β and the optimal number of iterations. However, in practice, it appears that both admit generic values for the type of images that were tested, namely, $\beta = 0.3$ and *Its.* $= 2$. Finally, when fixing w, r, β, and *Its.* to the values corresponding to the optimal result for a given image, the influence of k beyond a certain value is relatively small in terms of RMSE – the figures were not included in the document.

Apparently, NL-means produces slightly over-smoothed images while AWkNN does not fully get rid of the noise. This is particularly noticeable on "Lena", which contains many homogeneous or slowly varying areas – see Fig. 8.6. In these

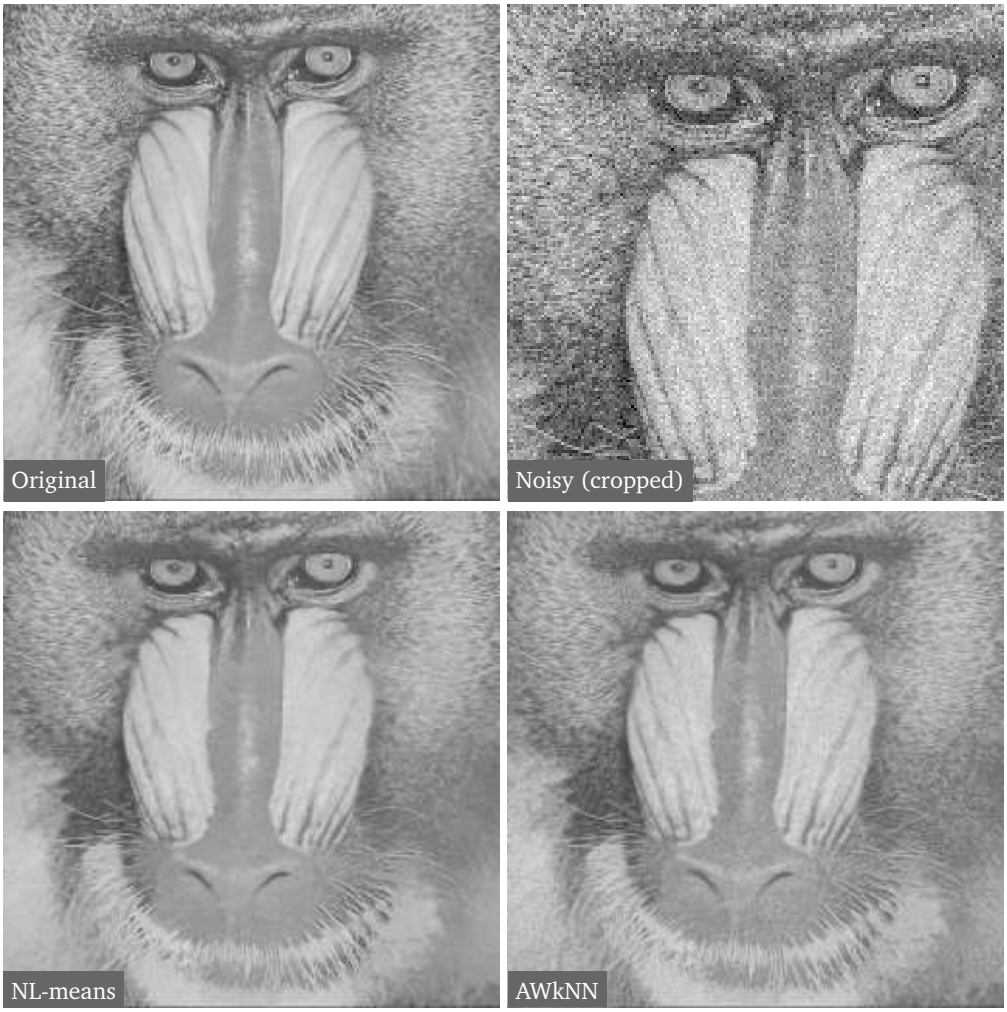

Figure 8.5 – Denoising of the image “Baboon”.

regions, NL-means performs very well since oversmoothing has little consequences. However, the hat feathers are better preserved by AWkNN. The other two images are more textured. In such a situation, the denoising results of AWkNN seem somewhat more natural. It is especially visible on “Aerial” – see Fig. 8.4. This encourages to investigate whether alternative ways to define local adaptability could still allow the preservation of textures while better recovering homogeneous areas.

Figure 8.6 – Denoising of the image “Lena”. The homogeneous areas are well recovered by NL-means whereas AWkNN better preserves the textures.

	NL-means				AWkNN					
	w	r	h	*RMSE*	*RMSE*	w	r	k	β	*Its.*
Aerial	10	2	0.8	**13.331**	**13.575**	15	4	80	0.4	2
	15	2	0.8	**13.332**	**13.575**	15	5	60	0.4	2
					13.575	15	5	80	0.4	2
RMSE: $\mu_r \pm \sigma_r$	14.63 $\pm$ 1.09				14.49 $\pm$ 1.13					
Baboon	10	2	0.8	**12.648**	**13.139**	10	5	80	0.4	2
RMSE: $\mu_r \pm \sigma_r$	13.93 $\pm$ 1.11				13.89 $\pm$ 0.72					
Lena	4	5	0.8	**7.564**	**8.681**	7	5	40	0.4	3
	7	5	0.8	**7.564**						
RMSE: $\mu_r \pm \sigma_r$	8.41 $\pm$ 0.65				9.36 $\pm$ 0.47					

Table 8.4 – Quantitative denoising results for the images "Aerial", "Baboon", and "Lena" – see Tabs. 8.2 and 8.3 for the notations.

8.6 Summary

The proposed development (cascading conditional entropy, mean shift and a kNN approximation) provides a new variational interpretation of NL-means. It also offers the possibility to adapt the denoising process locally. However, further investigation is needed to better exploit this opportunity.

In practice, it is quite usual for denoising methods to have a parameter (such as a threshold) whose optimal value depends on the noise level. It is the case of NL-means. When the noise level can be guessed, it performs very well. However, the sensitivity of the method with respect to this parameter might not be ideally low. The proposed method seems to behave well regarding the tuning of its parameter k. When k exceeds a certain value, the quality of denoising does not change significantly as a function of k – at least in terms of RMSE.

Referring to the proposed method as an operator on an image, one can note that this operator is not idempotent. Applying it again and again will produce images less and less noisy but also, eventually, degraded. In such an iterative scenario, the denoising operator should therefore be balanced with a data fidelity term,[#5] preferably following the same information-theoretic inspiration.

[#5] As also suggested in [Kin+05].

Chapter 9

Tracking

Context

The goal of region-of-interest (ROI) tracking in a video sequence is to determine in successive frames the region which best matches, in terms of a similarity measure, a ROI defined in a reference frame. Some tracking methods define similarity measures which combine several visual features into a probability density function (PDF) representation, thus building a descriptive model of the ROI. This approach implies dealing with PDFs with domains of definition of high dimension. To overcome this obstacle, a standard solution is to assume independence between the different features in order to bring out low-dimension marginal laws and/or to make some parametric assumptions on the PDFs at the cost of generality.

Alternatively, these assumptions can be discarded by having recourse to the Kullback-Leibler divergence expressed in the k nearest neighbor (kNN) framework. In consequence, the divergence is estimated directly from the high-dimensional samples (*i.e.*, without explicit estimation of the underlying PDFs), adapting to the local sample density. As an application, we defined 5, 7, and 13-dimensional feature vectors gathering color information (including pixel-based, gradient-based and patch-based) and spatial layout. The proposed procedure performs tracking allowing for translation and scaling of the ROI.

9.1 Methodological choices

9.1.1 A statistical approach

Two classical similarity measures are the sum of squared differences (SSD) and the sum of absolute differences (SAD) between the reference region-of-interest (ROI) and a candidate region in a target frame. They impose a strict geometric constraint since the underlying residual is computed with a deterministic pixel-to-pixel correspondence between the reference ROI and the candidate region. Therefore, they are not adapted to complex motions. Moreover, they implicitly

correspond to parametric assumptions on the residual probability density function (PDF) (respectively, Laplacian and Gaussian for the two examples above), which might not help in efficiently accounting for outliers due, for example, to partial occlusions.

An alternative is to adopt a statistical point of view by building a PDF from the ROI and using it as a reference 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, and therefore for complex motions. The PDFs can be defined in a radiometric space [Com+00, Per+02], either in grayscale or color. To improve the tracking accuracy, later developments tend to show that more information is required. Different features were integrated into the ROI PDF model, *e.g.*, employing spatial derivative filters [Bro+03, Bro+04, Low04], Gabor or wavelet filters [Par&Der02], and temporal filters [Bro+03, Bug&Per07]. A review of segmentation methods based on this framework was carried out in [Cre+07].

While this increase of description features improves accuracy, their combination leads to high-dimensional PDFs. There exist efficient [Sco92, Ihl03] and fast [Yan+03] 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 [Bug&Per07] so as to ensure that it will enclose enough samples. However, the resulting PDFs are over-smoothed. Another standard solution is to assume independence between the different features in order to bring out low-dimension marginal laws [Bro+03] and/or to make some parametric assumptions on the PDFs [Elg+03]. While these solutions may be satisfactory in some cases, they appear inappropriate for tracking – see Section 9.1.2.

9.1.2 High-dimensional feature space

The combination of color and geometry proved to be efficient for tracking. 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 [Com+00, Per+02]. 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 ROI pixels [Elg+03]. In this case, 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 conditionally on the ROI, follows a uniform distribution regardless of the ROI and, therefore, brings no information. While considering color and geometry jointly, simplification can still be achieved

by approximating the PDFs with parametric laws [Elg+03]. Nevertheless, fully data-driven, nonparametric PDF estimation was advantageously applied to segmentation [Aub+03, Kim+05, Her+06]. This approach will be followed.

In this context, the color and geometry feature space will be extended with the gradient of the luminance and patches of the luminance [Bol+07, Bol+09]. The former was motivated by the fact that involving the gradient, besides being efficient for keypoint matching [Low04, Mor&Yu09], has proved to increase accuracy in another motion-related task, namely optical flow computation [Bro+04]. The latter was motivated by works such as [Lee+03, Bua+05a, Car+08]. Finally, the k nearest neighbor (kNN) framework will help handling the components of these high-dimensional feature vectors jointly and to work in a locally adaptive manner in the feature space, thus avoiding under or oversmoothing in processing the data set [Bol+07, Bol+09].

The following development applies to feature spaces of arbitrary dimension. In practice, though, the experiments presented in Sections 9.5 and 9.6 involve features of dimension 5, 7, or 13.

9.1.3 Similarity measure

The kNN PDF estimators were proposed a while ago [Fix&Hod51, Lof&Que65]. Yet, they did not receive much attention since they are biased [Ter&Sco92, Sai02]. Recently though, corrective terms have been derived to cancel the bias, leading to consistent kNN-based estimators of statistical measures such as entropy [Koz&Leo87, Gor+05, Leo+08]. Moreover, even if the kNN PDF estimator is only adapted to high dimensions [Ter&Sco92], the kNN entropy estimator appears to be accurate in both low and high dimensions – see Section 5.4.2.

On these grounds, the Kullback-Leibler divergence between high-dimensional PDFs will be suggested as a similarity measure for tracking. Although this measure has already been proposed for this application [Elg+03, Fre&Zha04], here the divergence will be expressed nonparametrically, with no assumptions, and directly from the samples, *i.e.*, without explicit estimation of the underlying PDFs. This divergence estimator being well-adapted to high dimensions, it can be used with extended radiometric/geometric feature spaces [Bol+07, Bol+09].

9.1.4 Notations

A statistical measure $\mathcal{M}$ function of a PDF f_U (*e.g.*, entropy) might appear as $\mathcal{M}(f_U)$ or $\mathcal{M}(U)$, where U is a set of samples drawn from f_U. This notation may be used whether it refers to the definition of the measure or a sample-based estimation of it.

9.2 Similarity measure between ROIs

9.2.1 Definition and motivation

Let I_{ref} be the reference frame in which the ROI domain Ω is (user-)defined and let I_{tgt} be the target frame in which the region which best matches this reference ROI (in terms of a given similarity measure) is to be searched for. Assume that Ω is sampled on, *e.g.*, a Cartesian grid. At each grid node i, a feature vector of $\mathbb{R}^d$ describing the frame locally at i can be built. For convenience, the set of grid nodes will also be denoted by Ω. Given the statistical approach chosen in Sections 9.1.2 and 9.1.3, the region search mentioned above amounts to finding the geometric transformation Φ such that

$$\Phi = \arg\min_{\varphi} \mathfrak{D}_{\text{KL}}(f_{T_\varphi}, f_R) \tag{9.1}$$

where $\mathfrak{D}_{\text{KL}}$ is the Kullback-Leibler divergence (or information gain), and f_R, respectively f_T, is the PDF of the reference feature samples $R = \{R(i), i \in \Omega\}$ in I_{ref}, respectively the PDF of the target feature samples $T_\varphi = \{T_\varphi(i), i \in \Omega\}$ in I_{tgt}. Whenever appropriate, U or V will be used as a generic notation for either R or T_φ.

The choice of a non-symmetric measure and, further, the choice of the order of the arguments in (9.1) deserve to be motivated. Various works proposed symmetric versions of the Kullback-Leibler divergence, *e.g.*, J-divergence and Jensen-Shannon divergence [Lin91]. Nevertheless, for tracking, $\mathfrak{D}_{\text{KL}}(f_{T_\varphi}, f_R)$ seems to be appropriate. Indeed, f_{T_φ} can never be identical to f_R due to noise, 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 minimization enforces a relevant behavior in trying to "align" the main modes of the PDFs – see Section 3.1. In a way, it follows the same philosophy as the Bhattacharya coefficient,[#1] a measure widely used for tracking since a mean shift-based implementation has been proposed [Com+00].

9.2.2 Estimation in the kNN framework

The estimation of $\mathfrak{D}_{\text{KL}}(f_{T_\varphi}, f_R)$ in the kNN framework directly from the sample sets R and T_φ is described in Section 5.2.3. It is reminded here

$$\mathfrak{D}_{\text{KL}}(f_{T_\varphi}, f_R) = \log \frac{|R|}{|T_\varphi|-1} + \frac{d}{|T_\varphi|} \sum_{s \in T_\varphi} \log \frac{\rho_k(R, s)}{\rho_k(T_\varphi, s)} \ . \tag{9.2}$$

[#1] The Bhattacharya coefficient is only sensitive to mismatches within the intersection of the supports of the PDFs while the (supposedly) secondary modes of the ROIs are located outside this intersection.

9.3 Feature space: handling geometry and radiometry

As noted earlier, the feature vectors combine radiometry and geometry. Radiometry allows to check if the reference ROI and a candidate 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. Sections 9.3.1, 9.3.2, and 9.3.3 describe three levels of strictness. It is assumed that R and T_φ contain radiometric information only.

9.3.1 Geometry-free similarity measures

Classically, the similarity measure between the reference ROI and a candidate region 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 coefficient [Com+00, Per+02]

$$\mathfrak{D}_{\mathrm{BHA}}(f_{T_\varphi}, f_R) = \int_{\mathbb{R}^d} \sqrt{f_R(t)\ f_{T_\varphi}(t)}\ \mathrm{d}t \tag{9.3}$$

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 9.5.

Not accounting for the knowledge of where a given color is present allows more flexibility regarding the geometric transformation φ between the reference ROI and a candidate 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.

9.3.2 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 reference 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 [Bla&Ana96] 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

$$\mathfrak{D}_{\mathrm{SSD}}(T_\varphi, R) = \sum_{i\in\Omega}(T_\varphi(i) - R(i))^2 \tag{9.4}$$

and

$$\mathfrak{D}_{\mathrm{SAD}}(T_\varphi, R) = \sum_{i\in\Omega}\phi(T_\varphi(i) - R(i)) \tag{9.5}$$

where ϕ can be either the absolute value or a smooth approximation of it, *e.g.*, $\phi(x) = \sqrt{x^2+\epsilon^2} - \epsilon$ [Wei&Sch01].

9.3.3 Similarity measures with soft geometry

The geometric constraint can be soften, *e.g.*, by cascading a strict geometry approach and a radiometric approach [Bab+07] or, as proposed here, by adding geometry to the PDF-based approach presented in Section 9.3.1, *i.e.*, by defining a joint radiometric/geometric PDF [Elg+03, Bol+07]. 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 $\{(U(i), i), i \in \Omega\}$. In general, i can be any couple of independent spatial coordinates. For the ROI tracking application presented here, *normalized* Cartesian coordinates (x, y) were chosen, meaning that $(x, y) = (0, 0)$ at the center of the bounding boxes of the reference ROI or candidate regions and that $\max(\max(|x|), \max(|y|)) = 1$ among the points of the bounding boxes. Because geometry and radiometry are not comparable data, it might be useful or even necessary to weight one relatively to the other. It was decided to multiply the normalized coordinates by a spatial weight δ, resulting in $\max(\max(|x|), \max(|y|))$ being equal to δ. As is clear, the extended feature space is a subset of $\mathbb{R}^5$.

9.3.4 Enrichment of the radiometric features

As mentioned earlier, the proposed kNN framework is valid for any feature space dimension d. In Section 9.5, it will be clear that color and geometry as combined in Section 9.3.3 can provide enough information even in challenging situations. Yet, if this accounts for the correlation between a color and its location, it does not explicitly account for the correlation between the colors of neighboring pixels. It could be done by involving, *e.g.*, the color gradient or image patches [Lee+03, Car+08] – see Section 9.6. The influence of the chosen feature space, involving geometry and radiometry in several ways, is illustrated in Fig. 9.1. The following radiometric features will be tested

- $U(i) = I(i)$;
- $U(i) = \big(I(i), \gamma\nabla I_Y(i)\big)$;

Figure 9.1 – Influence of the feature space. The pixels in green are the 500 nearest neighbors of the pixel marked with the white cross. The feature space is defined as [a] geometry only, [b] color only, [c] color and geometry, [d] color, geometry, and gradient, and [e] 3×3-patch and geometry. Note that when the gradient is added to color and geometry, most of the neighboring pixels are located on edges with a gradient norm and orientation similar to the ones of the pixel of reference (besides having a similar color and not being to far away in the image plane).

- and $U(i) = \big(\text{Patch}_{3\times3}(I_Y)(i), I_U(i), I_V(i)\big)$;

where I_Y is the luminance component of I, (I_U, I_V) are the chrominance components, $\text{Patch}_{3\times3}(I_Y)(i)$ is a 3×3-patch of I_Y centered at i, and γ is a constant.

9.4 Tracking method

9.4.1 The main steps

The tracking is performed by minimizing the kNN Kullback-Leibler divergence (9.2) with respect to φ, or actually a set of parameters defining φ. The chosen motion model includes translation and scaling

$$\varphi(i) = i + M(i)\, p \tag{9.6}$$

$$= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x & 1 & 0 \\ y & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha - 1 \\ u \\ v \end{bmatrix} \tag{9.7}$$

where α is the scaling factor and (u, v) is the translation. The steps of the tracking algorithm are presented in Tab. 9.1. Starting from the frame $I_{\text{ref}} = I_1$, the tracking result over T consecutive frames is represented by the set $\{\varphi(I_{\text{tgt}})\} = \{\varphi(I_2), \varphi(I_3) \dots \varphi(I_T)\}$.

9.4.2 Series of minimizations

The minimization of (9.2) with respect to $\varphi = (\alpha, u, v)$ can be performed by a series of minimizations in (u, v) at α fixed. 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 the gradient (9.8) to zero. For computational considerations, they can instead be performed using a suboptimal search procedure such as the diamond search [Zhu&Ma00], thus following the approach for block matching of standard video coders. Naturally, more sophisticated search techniques such as particle filters [Per+02],[#2] also known as sequential Monte Carlo methods, can be used.

9.4.3 Mean shift-based gradient descent

The estimator (9.2) being defined in the kNN framework, it is not differentiable. Nonetheless, falling back on the Parzen formulation of the PDFs and the mean shift approximation, the derivative of the Kullback-Leibler divergence can be determined and then evaluated in the kNN framework. The corresponding development is presented in Appendix E and leads to

$$\begin{aligned}\nabla_\varphi \mathfrak{D}_{\text{KL}}(T_\varphi, R) = & -\frac{1}{k\,|T_\varphi|} \sum_{s \in T_\varphi} \mathcal{D}_s(T_\varphi) \left(\frac{d+2}{\rho_k^2(R,s)} \sum_{t \in W_{\rho_k(R,s)}} (t-s) \right. \\ & \left. - \frac{d+2}{\rho_k^2(T_\varphi,s)} \sum_{t \in W_{\rho_k(T_\varphi,s)}} (t-s) - \sum_{\substack{t \in T_\varphi \\ |t-s| = \rho_k(T_\varphi,t)}} \frac{t-s}{\rho_k(T_\varphi,t)} \right) \end{aligned} \tag{9.8}$$

where $\mathcal{D}_s(T_\varphi)$ is a $3 \times d$-matrix involving frame gradients and $W_{\rho_k(\cdot,s)}$ is a window of radius $\rho_k(\cdot, s)$ centered at sample s. As a consequence, the ROI tracking could be solved by gradient descent in the space of the parameters (α, u, v) using the kNN framework. 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 (especially much larger or much smaller). Therefore, a procedure based on a series of minimizations might be preferable – see Section 9.4.2.

[#2] These methods are particularly efficient in case of total occlusion of the object of interest on several frames.

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 a 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 (see Section 9.4.2):
      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 9.4.3) 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) $\varphi(I_{\text{tgt}}) \leftarrow \varphi = (\alpha, u, v) \leftarrow (\alpha\ \tilde{\beta}, u + \tilde{m}, v + \tilde{n})$

Table 9.1 – Pseudocode of the proposed tracking algorithm.

9.5 Some experimental comparisons

The test sequences of this section were selected for the specific conditions they exhibit, in particular, partial occlusions, variations of luminance, noise, and complex motion – non-frontoparallel motion with rotation and motion blur.

9.5.1 Setup

The proposed kNN-based algorithm presented in Section 9.4 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:

- kNN-KL – a geometry-free version of the proposed method;
- Pz-KL-G – a version of the proposed method where the kNN expression (9.2) of the divergence was replaced with an estimation based on Parzen windowing;[#3]
- SAD – an SAD version of the algorithm described in Tab. 9.1, *i.e.*, replacing the Kullback-Leibler divergence in step 4(b)i with the dissimilarity (9.5);
- Mean-Shift – a mean shift-based tracker whose implementation is publicly available [Col+05].

These comparisons 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, the tracking algorithm was kept simple, purposely setting aside improvements such as reference update and motion prediction. Moreover, for a fair comparison between the methods, the experimental setup of the above-mentioned Mean-Shift implementation was followed, namely, a rectangular reference ROI Ω (see Figs. 9.2, 9.4, and 9.8 for the dimensions) and a translation motion model φ[#4] with a pixel resolution. The chosen radiometric space was YUV because the standard test sequences used in these experiments are available in this color space.

For the kNN-based methods, the parameter k in (9.2) was chosen equal to 3, which satisfies the conditions mentioned after (5.8). The distance $\rho_k(U, s)$ to the k-th nearest neighbor of s in U was computed in the classical Euclidean sense and obtained using a publicly available toolbox [Got97].

The components of the feature vectors were normalized as follows: Y, U, and V were rescaled into the interval $[0, 1]$ and, as explained in Section 9.3.3, the coordinates (x, y) were rescaled into $[-1, 1]$, both in the reference ROI and the

[#3] This implementation is publicly available [Ihl03].
[#4] In other words, λ was set to $\{1\}$ in Tab. 9.1.

candidate regions, the origin being located at the center of the bounding box of the region. The spatial weighting δ was taken equal to 1.

The minimization in $\varphi = (u, v)$ was implemented using a suboptimal search procedure known as the diamond search [Zhu&Ma00]. Tracking was performed with I_{ref} being set to 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 area was empirically limited to ± 12 pixels horizontally and vertically around the position of the center of the ROI computed in frame I_{t-1}.

9.5.2 Partial occlusions

Sequence "Car" is part of the VIVID tracking testbed [Col+05]. It is composed of 640×480-frames. Tracking was performed on 150 consecutive frames – see Figs. 9.2 and 9.3. kNN-KL eventually lost the ROI and ended up tracking the second car which has colors similar to the reference ROI. This is probably due to the fact that it is based on radiometry only. Pz-KL-G also failed in tracking the car. Mean-Shift performed quite well although the tracking shifted upward when occlusion occurred in order to avoid including the green colors of the trees in the 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.

9.5.3 Variations of luminance

Sequence "Crew" is composed of 352×288-frames. Two faces were tracked on 80 consecutive frames – see Figs. 9.4 and 9.5. kNN-KL-G tracked the faces successfully. The other methods lost the ROIs sooner or later, apparently due to the variations of luminance. This is particularly obvious with *(i)* Mean-Shift whose brutal changes in tracking shift match the camera flashes in frames 16, 20, and 61 for the face on the left, and in frames 1, 7, and 61 for the face on the right, and *(ii)* kNN-KL whose tracking error seems to follow the curve of average intensity.

9.5.4 Noisy sequence

Sequence "Schnee" is composed of 768×576-frames. Two cars were tracked on 160 consecutive frames – see Figs. 9.6 and 9.7. This sequence can be considered noisy due to the snowflakes which fall rather densely. Despite this "Salt" noise, the two cars were accurately tracked by kNN-KL-G. The objects being small and rather homogeneous, their motion could be considered as a translation. Therefore the strict geometric constraint of SAD is not violated and it performed quite well. Clearly, Mean-Shift was disturbed by the noise. The other two methods (kNN-KL and Pz-KL-G) worked pretty well for one car. For the other car, Pz-KL-G performed

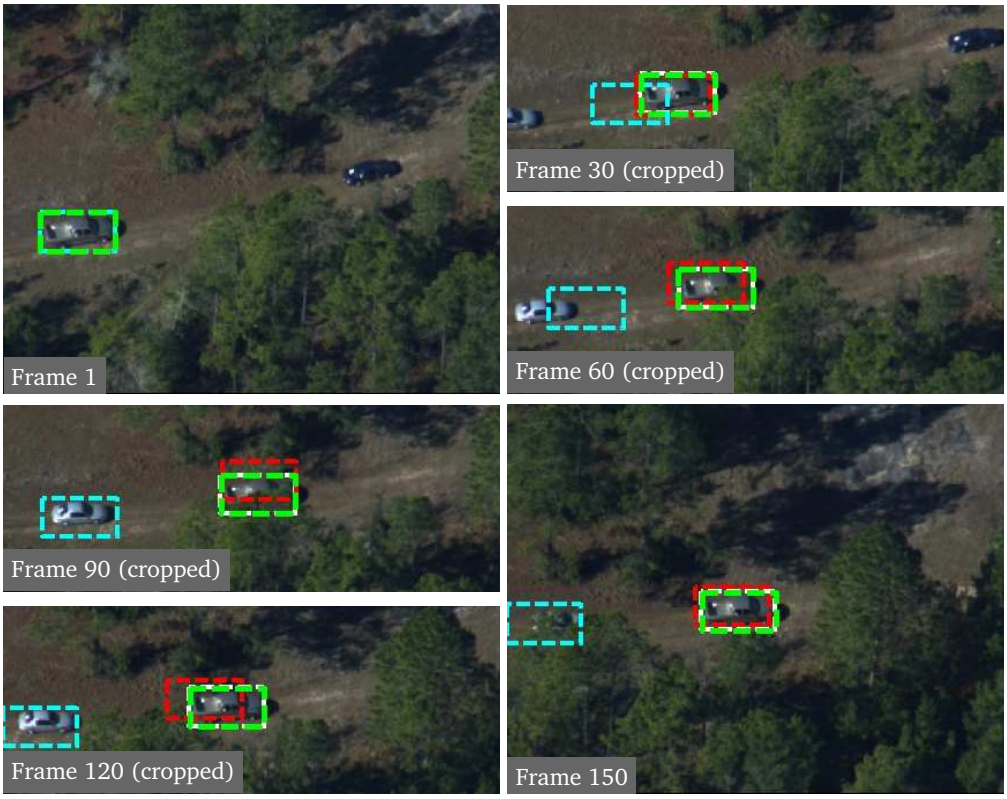

Figure 9.2 – Tracking on sequence "Car" (frame indices are relative to the reference frame). Green • kNN-KL-G (proposed method), Cyan • kNN-KL, Magenta • Pz-KL-G, Red • Mean-Shift, **Black** • SAD (white on the frames and black in the diagram). There are several frame drops at frame 38 (vertical dashed line in the diagram) and the tracked car is partially occulted by trees from frame 42 to frame 122 (gray area in the diagram). Ω: 95×47-rectangle – Figure continued in Fig. 9.3.

reasonably well for almost half of the sequence while kNN-KL, which does not account for geometry, failed immediately.

9.5.5 Complex motion

Sequence "Football" is composed of 352×288-frames. Tracking was performed on 20 consecutive frames – see Figs. 9.8 and 9.9. Note that part of the public has colors similar to colors that can be found in the reference 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

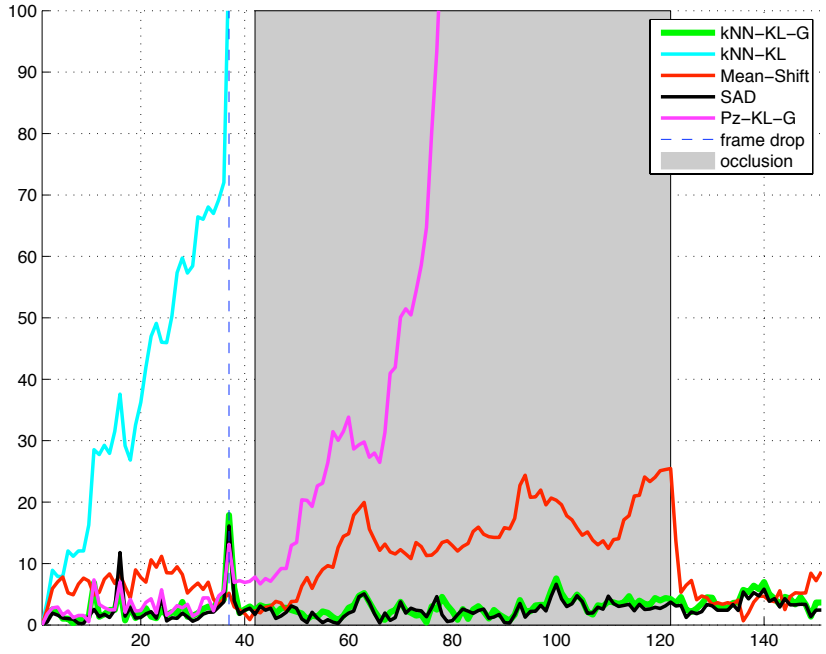

Figure 9.3 – Continuation of Fig. 9.2 – The diagram represents the shift (in percent of the ROI diagonal length in pixel) with respect to a manually defined tracking as a function of the frame index.

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

To support these conclusions, the dissimilarity between the reference ROI and candidate regions in frame 20 was computed as a function of the translation parameters for SAD, kNN-KL, Pz-KL-G, and kNN-KL-G – see Fig. 9.10. The SAD minimum is shifted as a result of the inappropriateness of the translation model between frame 1 and frame 20. kNN-KL has several local minima as there are several possible matches when accounting for radiometry only. By adding geometry, Pz-KL-G finds a unique minimum, although not at the right location. This is certainly due to the reduced accuracy of the Parzen-based estimator of the statistical measure in $\mathbb{R}^5$. Finally, kNN-KL-G has a minimum that matches the correct motion. Also note that the kNN-KL-G criterion seems strictly convex within a window around the minimum. It is not surprising since the Kullback-Leibler divergence is indeed strictly convex in its first argument. Let t_m be defined as $\alpha t_1 + (1-\alpha)t_2$ for some

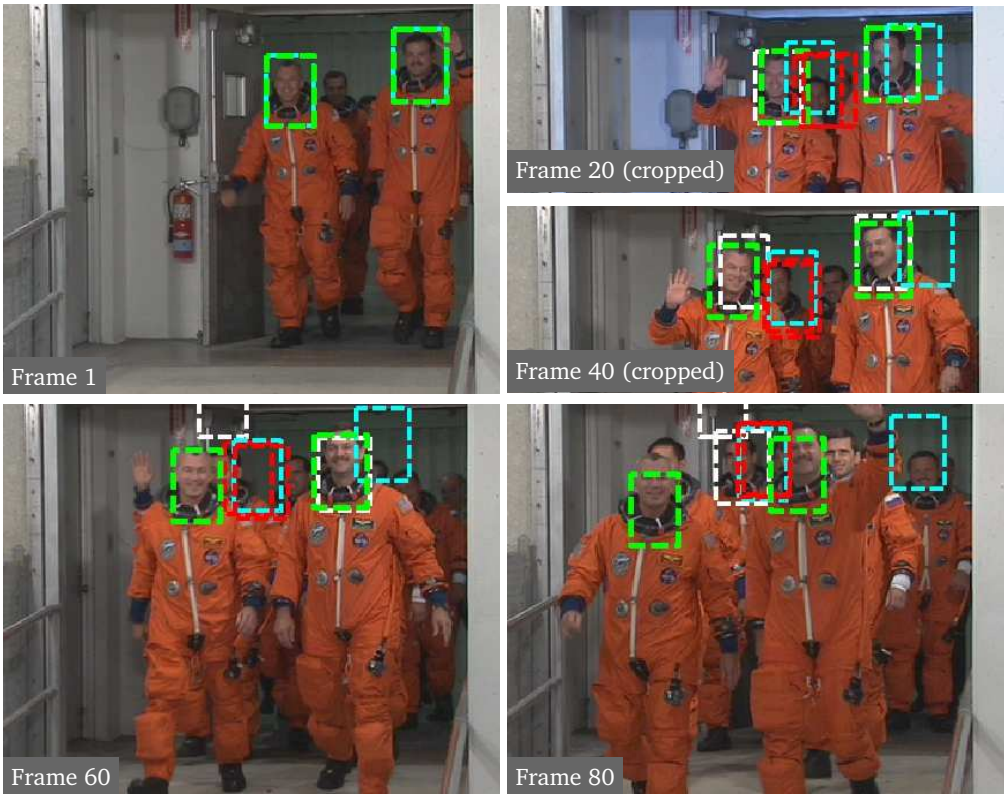

Figure 9.4 – Tracking on sequence "Crew" (frame indices are relative to the reference frame). Green • kNN-KL-G (proposed method), Cyan • kNN-KL, Magenta • Pz-KL-G, Red • Mean-Shift, **Black** • SAD (white on the frames and black in the diagram). There are two kinds of intensity changes in the sequence: 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). Ω: 33×52-rectangle – Figure continued in Fig. 9.5.

α in $[0,1]$. Then,

$$\begin{aligned}
\mathfrak{D}_{\mathrm{KL}}(t_m, r) &= \alpha \left[\int_{\mathbb{R}^d} t_1(x) \ \log t_m(x) \ \mathrm{d}x - \int_{\mathbb{R}^d} t_1(x) \ \log r(x) \ \mathrm{d}x \right] \\
&\quad + (1-\alpha) \left[\int_{\mathbb{R}^d} t_2(x) \ \log t_m(x) \ \mathrm{d}x - \int_{\mathbb{R}^d} t_2(x) \ \log r(x) \ \mathrm{d}x \right] \qquad (9.9) \\
&\leq \alpha \left[\int_{\mathbb{R}^d} t_1(x) \ \log t_1(x) \ \mathrm{d}x - \int_{\mathbb{R}^d} t_1(x) \ \log r(x) \ \mathrm{d}x \right] \\
&\quad + (1-\alpha) \left[\int_{\mathbb{R}^d} t_2(x) \ \log t_2(x) \ \mathrm{d}x - \int_{\mathbb{R}^d} t_2(x) \ \log r(x) \ \mathrm{d}x \right] \qquad (9.10) \\
&\leq \alpha \mathfrak{D}_{\mathrm{KL}}(t_1, r) + (1-\alpha) \mathfrak{D}_{\mathrm{KL}}(t_2, r) \qquad (9.11)
\end{aligned}$$

with equality if and only if $t_1 = t_m$ and $t_2 = t_m$, *i.e.*, $t_1 = t_2$. (The essential step in this development is due to the cross-entropy $H^{\times}(t, t_m)$ being larger than the

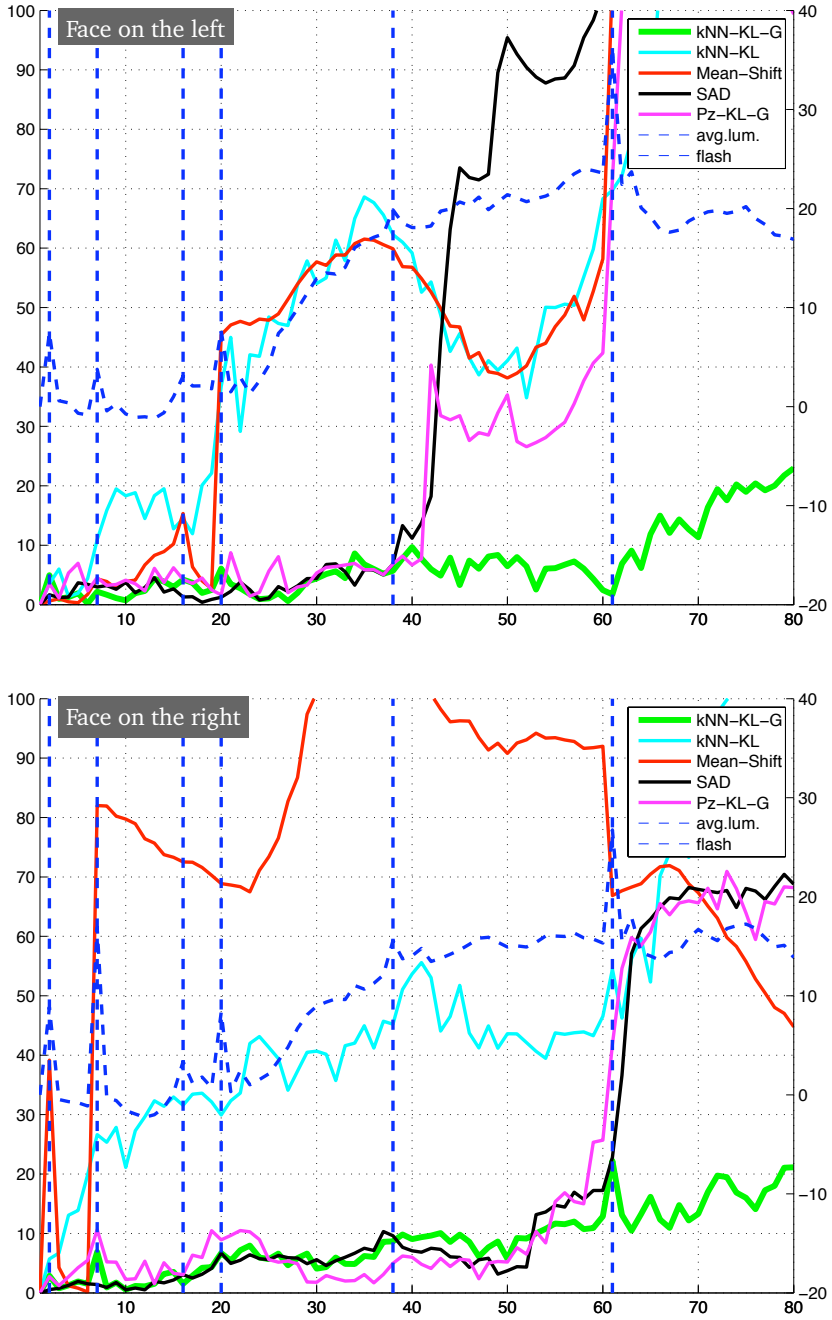

Figure 9.5 – Continuation of Fig. 9.4 – The diagrams represent the shift (in percent of the ROI diagonal length in pixel) with respect to manually defined trackings as a function of the frame index. The vertical axis on the right of each diagram corresponds to the blue dashed curves which represent the evolution of the average intensity (Y component) within the manually defined trackings. The average intensity in frame 1 is taken as a reference and the scale is in unit of intensity. Both the continuous intensity increase and the camera flashes are noticeable.

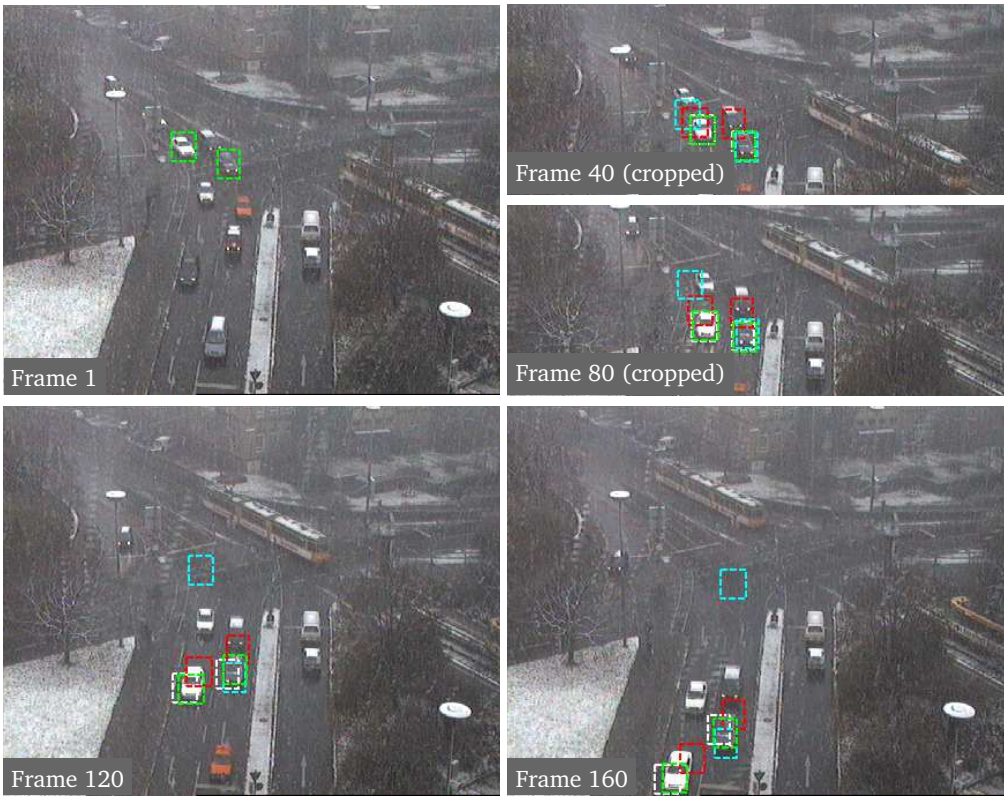

Figure 9.6 – Tracking on sequence "Schnee" (frame indices are relative to the reference frame). Green • kNN-KL-G (proposed method), Cyan • kNN-KL, Magenta • Pz-KL-G, Red • Mean-Shift, **Black** • SAD (white on the frames and black in the diagram). This sequence can be considered noisy due to the snowflakes. Ω: a 38×42-square for the car on the left and a 34×42-square for the car on the right – Figure continued in Fig. 9.7.

entropy of t.) This property of convexity is naturally interesting for the convergence of optimization algorithms – the diamond search in our case.

9.5.6 Summary

The previous comparisons can be coarsely summarized by selecting the two or three best and worst methods for each of the four sequences – see Tab. 9.2. The conclusions that could be made are:

- kNN-KL almost always failed. This is a known effect of not taking geometry into account (see Section 9.3.1);
- Mean-Shift failed in most of our tests (variation of illuminance and noise) but can also perform quite well;

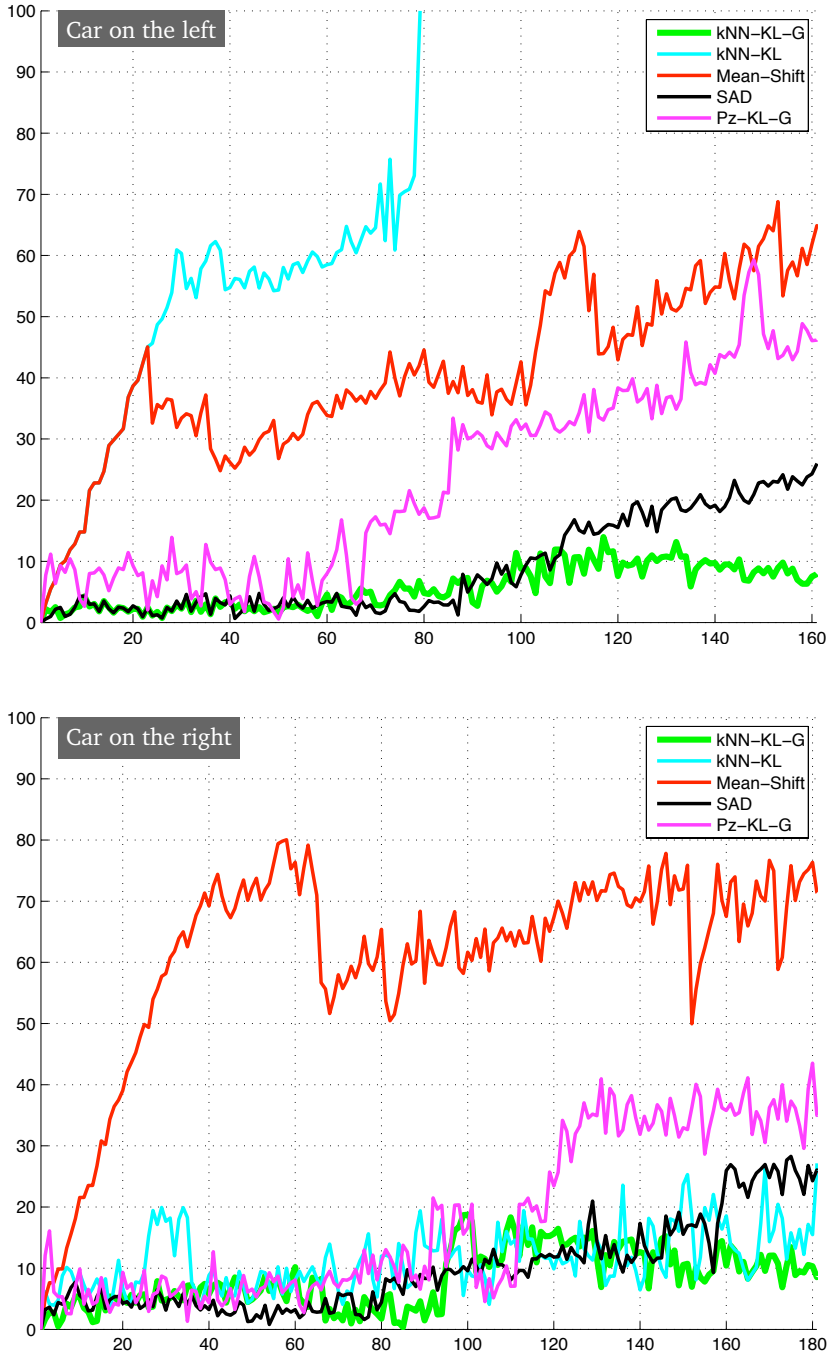

Figure 9.7 – Continuation of Fig. 9.6 – The diagrams represent the shift (in percent of the ROI diagonal) with respect to manually defined trackings as a function of the frame index.

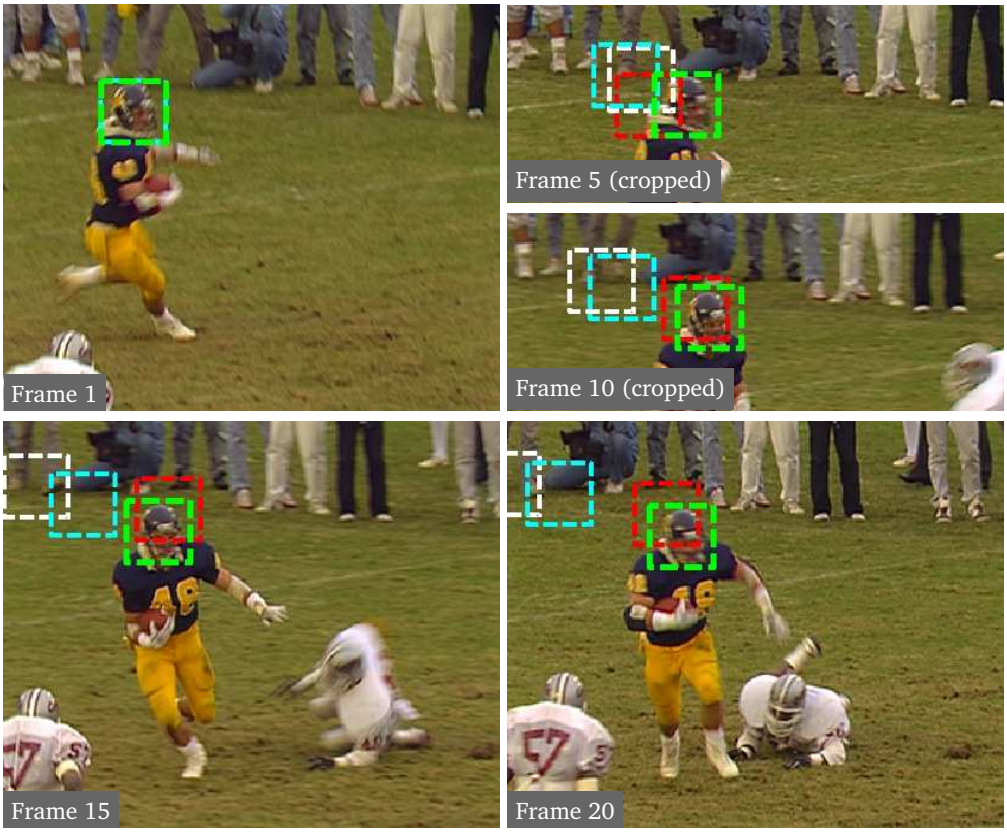

Figure 9.8 – Tracking on sequence "Football" (frame indices are relative to the reference frame). Green • kNN-KL-G (proposed method), Cyan • kNN-KL, Magenta • Pz-KL-G, Red • Mean-Shift, **Black** • SAD (white on the frames and black in the diagram). This sequence is characterized by a non-frontoparallel motion and a fast motion generating motion blur. Moreover, the motion has a rotational component responsible for the disappearance of some areas and the exposure of others. Ω: 43×43-square – Figure continued in Fig. 9.9.

- SAD might represent a computationally efficient alternative to kNN-KL-G if an average accuracy is considered satisfying for some task. Unfortunately, it can fail completely when the motion is complex (see Fig. 9.8) since it relies on a strict geometric constraint (see Section 9.3.2);
- The performance of Pz-KL-G ranges from reasonably good to terrible. It relies on the Parzen approach instead of the proposed kNN framework to estimate the chosen statistical measure, and therefore allows to illustrate the expected advantages of kNN (see Chapter 5);
- Finally, kNN-KL-G represents the best option in all cases.

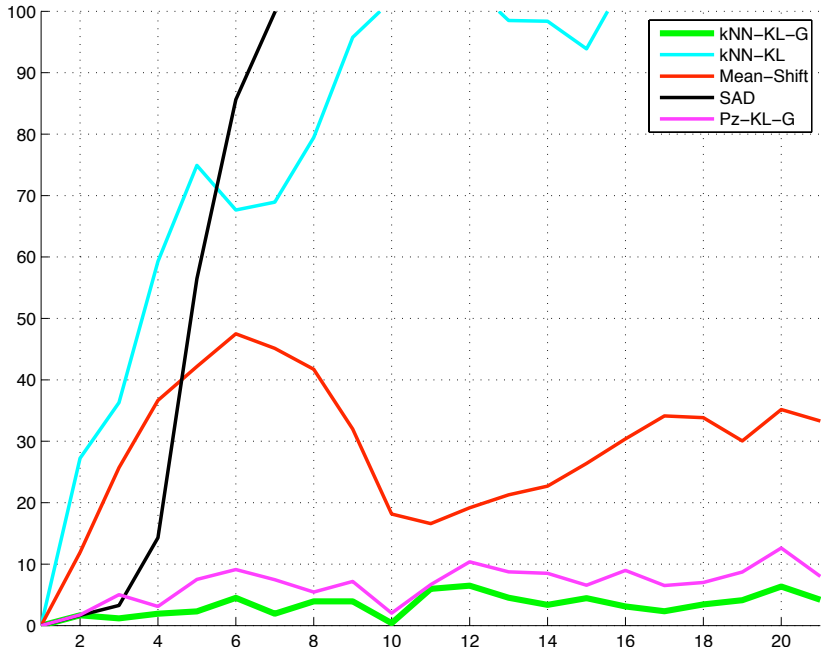

Figure 9.9 – Continuation of Fig. 9.8 – The diagram represents the shift (in percent of the ROI diagonal) with respect to a manually defined tracking as a function of the frame index.

Fig.	Best results	Worst results
9.2	kNN-KL-G, SAD & (to some extent) Mean-Shift	kNN-KL & Pz-KL-G
9.4a	kNN-KL-G	kNN-KL & Mean-Shift
9.4b	kNN-KL-G & (to some extent) SAD & Pz-KL-G	kNN-KL & Mean-Shift
9.6a	kNN-KL-G & SAD	kNN-KL & Mean-Shift
9.6b	kNN-KL-G, SAD, kNN-KL & (to some extent) Pz-KL-G	Mean-Shift
9.8	kNN-KL-G & Pz-KL-G	kNN-KL & SAD

Table 9.2 – Summary of the comparisons on the four sequences "Car", "Crew", "Schnee", and "Football". The "a" and "b" labels were added to distinguish between the ROIs when two simultaneous trackings were performed.

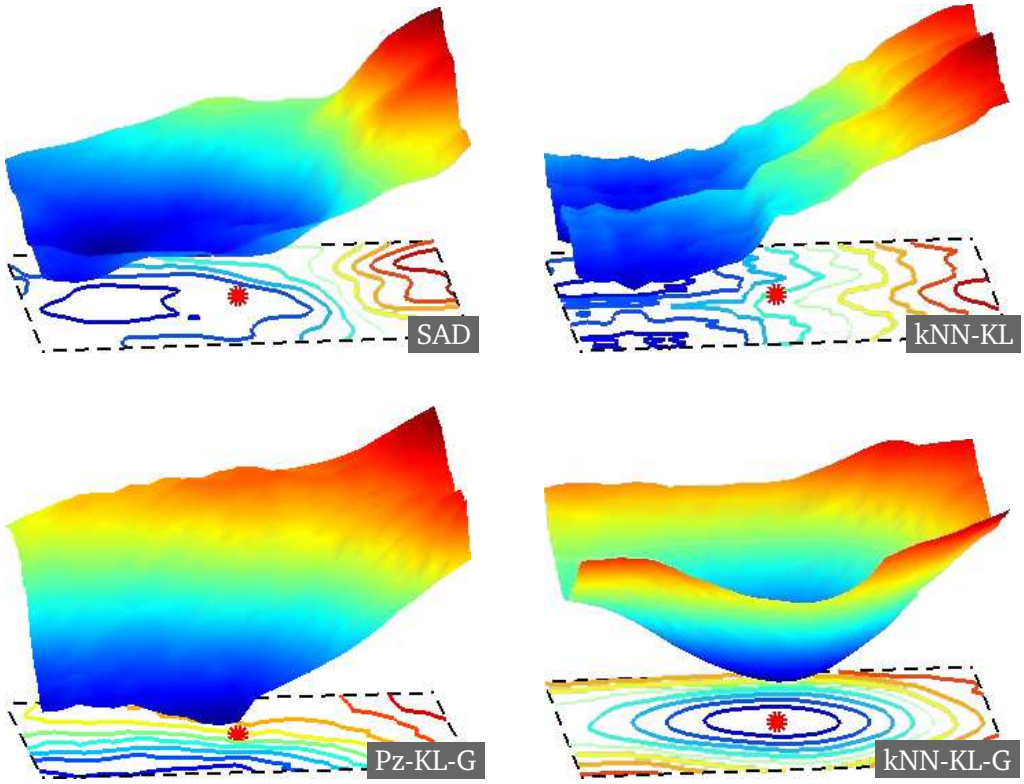

Figure 9.10 – Dissimilarity between the reference ROI of sequence "Football" and candidate regions in frame 20 as a function of horizontal and vertical translations. The dashed box is a 12×12-square (the same size as the search window). The red spot at its center represents the correct translation. The SAD and Pz-KL-G minima are shifted and kNN-KL has two local minima whereas the minimum of kNN-KL-G seems accurate.

9.5.7 Stability with respect to k

To evaluate the stability of kNN-KL-G with respect to the choice of the parameter k, tracking was performed on sequence "Football" with various values of k that comply 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 Tab. 9.3. In the i^{th} frame, the shift $s_i = (x_i, y_i)$ between the bounding box obtained for $k = 3$ and the bounding box obtained for another value of k was determined. The line "Avg norm" in Tab. 9.3 corresponds to the average of the shift norm over the 20 frames $(1/20)\sum_i |s_i|$. Clearly, as k gets further away from the chosen value of reference, the solution of the tracking has also a tendency to shift away from the solution of reference. This is not surprising and, looking at the numbers, this behavior is not excessive. The following lines of Tab. 9.3 correspond to the norm and the orientation of the sum of the shifts in the successive frames $\sum_i s_i$. These measures allow us to check

Value of k	3	10	20	43$=\sqrt{\lvert\Omega\rvert}$
Avg norm	*Ref.*	0.46	1.69	2.65
Sum norm	*Ref.*	0.60	1.80	1.30
Sum angle	*Ref.*	118	-68	117

Table 9.3 – Stability of kNN-KL-G with respect to k: average norm of the tracking shifts, norm of the sum of the shifts (both in percent of the ROI diagonal length in pixel), and orientation of the sum of the shifts (in degree) taking the result obtained with $k = 3$ as a reference.

whether the shifts correspond to a consistent bias. Apparently, this is not the case since the norm of the cumulated shifts does not increase consistently with k and the orientations vary. In conclusion, as k increases, the tracking shift tends to oscillate more and more around the solution for $k = 3$, but confined in an acceptable range and without any obvious coherence. Therefore, the method appears to be quite stable with respect to k.

9.6 Brief experimental study

9.6.1 Setup

In this section, the proposed method is compared with variants of itself. Consequently, there are no constraints on the experimental setup. Then, scaling was taken into account by choosing $\lambda = \{0.98, 0.99, 1, 1.01, 1.02\}$, and the gradient of the luminance and patches of the luminance were optionally used as additional radiometric features.

An extra sequence [Exc06] was used. It was available in the RGB space but, to remain consistent with Section 9.5, it was converted to the YUV space before processing. The components of the feature vectors were normalized as follows: Y, U, and V, were rescaled into the interval $[0, 1]$, the gradient of the luminance Y (whenever used) was computed using the filter $[-1, 9, -45, 0, 45, -9, 1]/60$ and rescaled using $\gamma = 10$, and, as a reminder, the local pixel coordinates (x, y) were rescaled into $[-1, 1]$. These coordinates were further modified by applying the spatial weighting δ for the target and the scaling $\alpha\delta$, $\alpha \in \lambda$ for the reference, meaning that (x_R, y_R) actually belongs to the interval $[-\alpha\delta, \alpha\delta]^2$ and (x_T, y_T) belongs to $[-\delta, \delta]^2$ (see Tab. 9.1).

The minimization in $\varphi = (\alpha, u, v)$ was either performed by a series of minimizations at α fixed (see Section 9.4.2) implemented using a suboptimal search procedure known as the diamond search [Zhu&Ma00], or by a gradient descent procedure: for stability, the gradient (9.8) was normalized such that the translation component has a norm equal to one and the scaling component is either 0.99, 1.00, or 1.01. The former minimization strategy will be referred to as “Discrete search” and the latter one as “Gradient search”.

The other aspects of the setup were identical to Section 9.5.1.

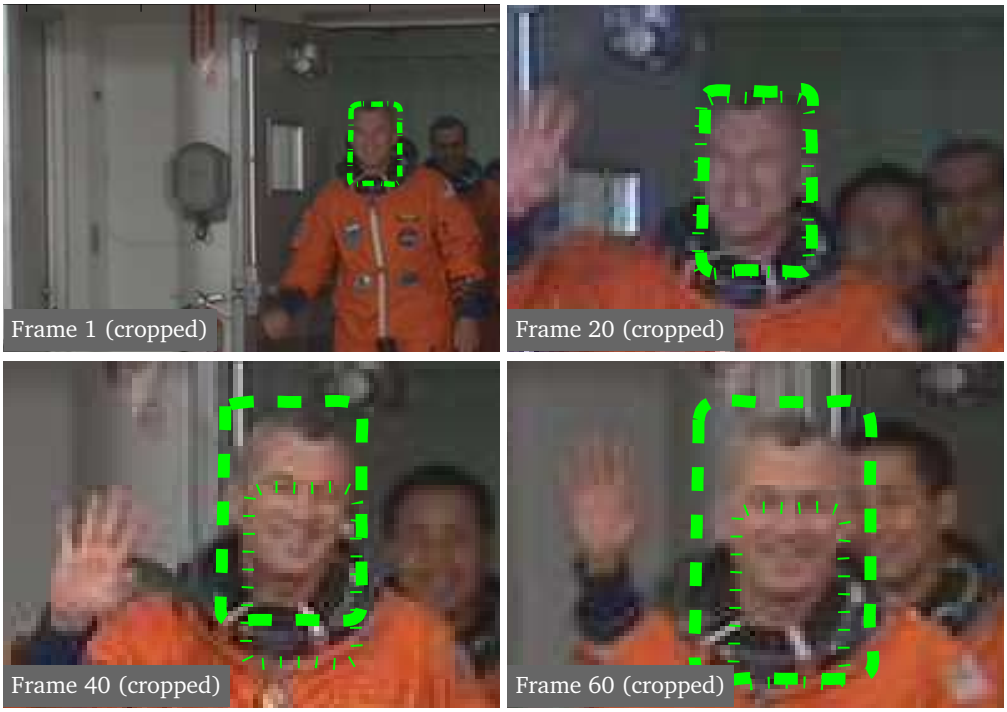

Figure 9.11 – Tracking on sequence "Crew" (frame indices are relative to the reference frame). $\delta = 1$: dotted line; $\delta = 0.6$: dashed line. Ω: 33×52-rectangle.

9.6.2 Influence of δ

A tracking was performed on 60 consecutive frames of sequence "Crew" using the discrete search and two values of δ: 0.6 and 1. The radiometric information was limited to color – see Fig. 9.11. As expected, the spatial weighting has an influence on the tracking quality. Nevertheless, it is not dramatic since it seems to play mostly on the duration the tracking can be considered accurate for rather than acting on the stability of the processing.

9.6.3 Discrete search vs. gradient search

Sequence "Poltergay" [Exc06] is composed of 720×576-frames. Tracking was performed on 100 consecutive frames using either the discrete search or the gradient search, with $\delta = 0.8$ and the feature space defined as (color, gradient of the luminance, geometry) – see Fig. 9.12 and 9.13. As expected, the discrete search performed better than the gradient search due to the presence of local minima which can mislead a gradient descent – the gradient search performed very decently, though. However, this is at the cost of a computational time 7 times higher.

Figure 9.12 – Tracking on sequence "Poltergay" (frame indices are relative to the reference frame). Discrete search: dashed line; gradient search: plain line. Ω: 63×101-rectangle – Figure continued in Fig. 9.13.

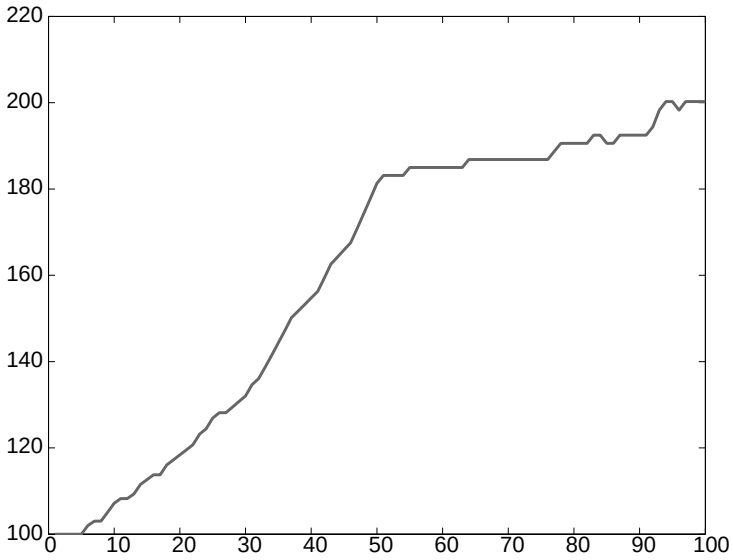

Figure 9.13 – Continuation of Fig. 9.12 – The diagram represents the scaling of the ROI (parameter α times 100) as a function of the frame index for the solution using the discrete search. To deduce the scaling in terms of area, the values must be divided by 100 and squared. At the highest point, the ROI is more than 4 times larger in area than Ω.

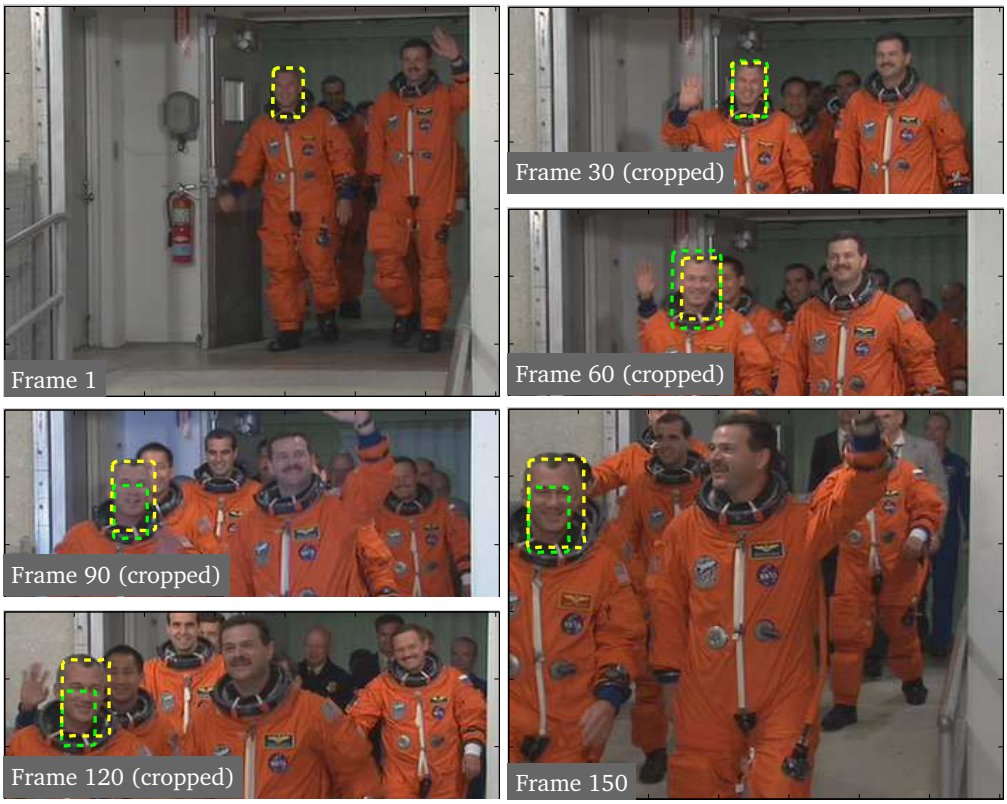

Figure 9.14 – Tracking on sequence "Crew" (frame indices are relative to the reference frame). Green • Without the gradient of the luminance, Yellow • With the gradient. Ω: 33×52-rectangle – Figure continued in Fig. 9.15.

9.6.4 Gradient as an additional radiometric feature

A tracking was performed on 150 consecutive frames of sequence "Crew" using the discrete search and $\delta = 0.6$. The feature space was either (color, geometry) or (color, gradient of the luminance, geometry) – see Figs. 9.14 and 9.15. Clearly, the addition of the gradient improved the tracking accuracy. As mentioned earlier, any other feature can be added without algorithm modifications. (It only add terms to the Euclidean distance computation between the feature vectors during the kNN search). Nevertheless, more features does not always imply better tracking performances – see Section 9.6.5.

9.6.5 Adjusting the feature space in the presence of noise

A tracking was performed on 100 consecutive frames of two degraded versions of sequence "Poltergay" using the discrete search, $\delta = 0.8$ and the feature space being either *(i)* (color, geometry), *(ii)* (color, gradient of the luminance, geometry), or

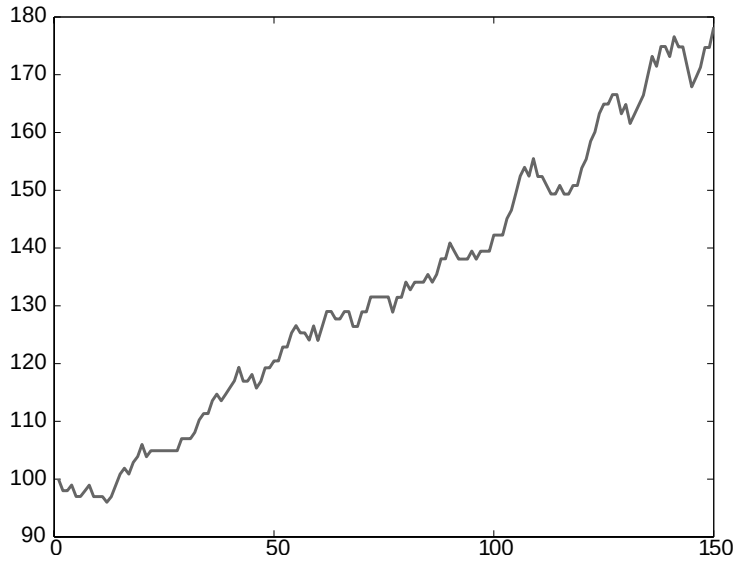

Figure 9.15 – Continuation of Fig. 9.14 – The diagram represents the scaling of the ROI (parameter α times 100) as a function of the frame index for the solution that used the gradient. To deduce the scaling in terms of area, the values must be divided by 100 and squared. At the highest point, the ROI is more than 3 times larger in area than Ω.

(iii) (patch 3×3 on Y, U, V, geometry).[#5] The first degraded version was obtained by compressing each frame at a very low rate using a JPEG2000 coder – see Fig. 9.16. The original frames in JPEG format are around 32 kB in size. The JPEG2000 compression rate was chosen such that the size went down to 4 kB. For the second degraded version, each color channel of each frame was corrupted by a Gaussian noise of mean zero and variance equal to 9 (see Fig. 9.16).

The proposed method being independent of the ROI shape, the rectangular shape used so far for Ω was replaced with an ellipse with a bounding box of 63×101 pixels.

Both feature spaces *(i)* and *(ii)* dealt very well with the JPEG2000 artifacts – see Fig. 9.17. Therefore, the feature space *(iii)* was not even considered. However, since the Gaussian noise largely corrupted the gradient of the frames, the object could not be tracked correctly using the feature space (color, gradient of the luminance, geometry). Although not satisfying, the (color, geometry) space produced a more acceptable tracking. (Actually, it is even quite accurate until frame 22 – not shown in Fig. 9.18.) It is only when considering patches (feature space *(iii)*) that the tracking became fully accurate – see Fig. 9.18. This robustness to noise when having recourse to patches is not surprising since this kind of information is used for denoising [Bua+05a, Bua+05b, Awa&Whi06, Ker&Bou06, Bou+07, Dab+07, Ang+08a].

[#5] Space of dimension 13.

Figure 9.16 – Detail of frame 1 of sequence "Poltergay" before and after degradation. The JPEG2000 compression rate was around 8 compared to the original JPEG frames. The Gaussian noise had a mean of zero and a variance of 9.

9.7 Summary

The proposed method can be characterized by such keywords as statistical, non-parametric, variable kernel bandwidth (kNN), joint radiometry 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 temporal 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

Figure 9.17 – Tracking on sequence "Poltergay" in the presence of JPEG2000 compression artifacts (frame indices are relative to the reference frame). Green • Without the gradient of the luminance (feature space *(i)*), Yellow • With the gradient (feature space *(ii)*). Ω: ellipse with a bounding box of 63×101 pixels.

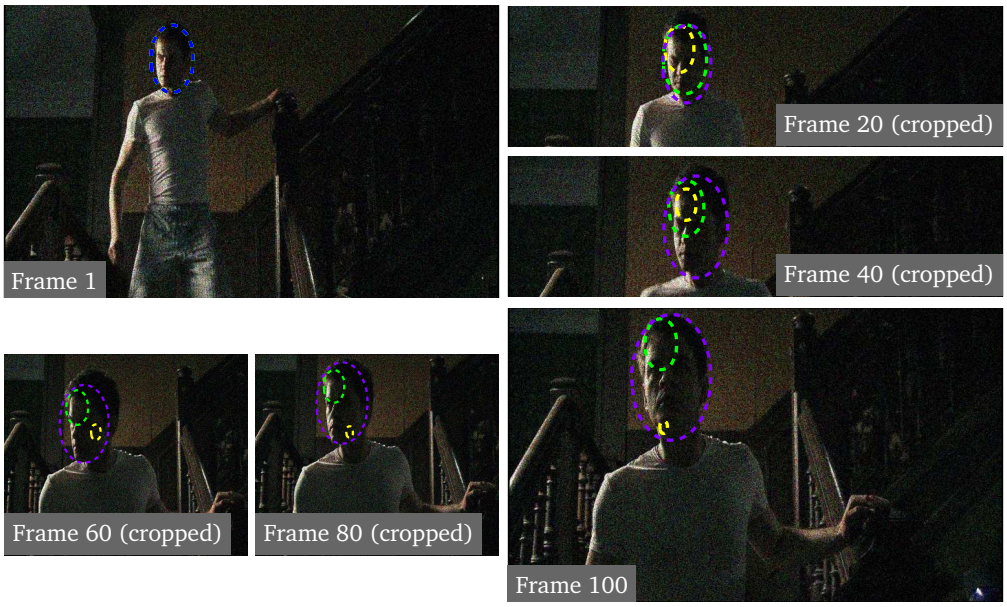

Figure 9.18 – Tracking on sequence "Poltergay" in the presence of Gaussian noise (frame indices are relative to the reference frame). Green • Without the gradient of the luminance (feature space *(i)*), Yellow • With the gradient (feature space *(ii)*), Violet • With the patches (feature space *(iii)*). Ω: ellipse with a bounding box of 63×101 pixels.

samples, is a weakness, as is clear from the experimental results. *(iv)* The mean shift-based tracker used in the comparisons [Com+00, Col+05] rely on another statistical measure: the Bhattacharya coefficient. Whether the differences observed in the experimental results presented here between this tracker and the proposed method depend on the measure itself or on the way geometry is involved[#6] is unclear.

To a certain extent, the proposed method seems to provide answers to the problems *(i)* to *(iii)*.

[#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 radiometry and geometry processing for kNN-KL-G.

Chapter 10

Other tasks

Context

Expressing image and video processing tasks as a problem of minimization of a similarity measure, and deriving a practical algorithm in the k nearest neighbor (kNN) framework is a rather general approach. The chapter on segmentation focused on a specific minimization aspect. Then, solutions to image denoising and region-of-interest (ROI) tracking were developed in the above-mentioned context. Other possible applications will be briefly described in this chapter.

10.1 Inpainting

The nature of the denoising method of Section 8 makes its modification for inpainting [Ber+00, Ber+03, Cri+04] easily conceivable. Indeed, it recovers the original color of a *damaged* pixel $\tilde{x}$ by looking for other pixels $\tilde{x}_{t_i}$ in the image with surrounding colors similar to the colors surrounding $\tilde{x}$. The recovering procedure relies on a weighted average involving $\tilde{x}$ and the $\tilde{x}_{t_i}$'s. If $\tilde{x}$ is so damaged that it is totally undependable, one can imagine to remove it from the weighted average. This is how denoising can be turned into inpainting. However, note that such a procedure can only inpaint reliably rather thin regions (*e.g.*, scratches) unless additional constraints are used [Bar+09].

Assume $\tilde{x}$ is a pixel of an area I to be inpainted. Let y be the colors in the neighborhood of $\tilde{x}$, excluding the pixels in I, if any. Let x_{t_i}, $i \in [1..k]$, be the k nearest neighbors of $\tilde{x}$ in the sense that the distances between y and the neighborhoods y_{t_i} of x_{t_i}[#1] are the k smallest ones among all the neighborhoods of the image. The missing color $\tilde{x}$ can then be replaced with a weighted average of the colors x_{t_i}. However, note that the purpose of this weighted average was to get rid of the noise added to the pixels $\tilde{x}$ and $\tilde{x}_{t_i}$. In inpainting, the pixels are noise-free. As a consequence, a weighted average could induce some blurring. It

[#1] See Tab. 10.1 concerning how the neighborhoods y_{t_i} are built.

1. Let I be the user-defined area to be inpainted
2. Initialization: $J \leftarrow I$
3. For each pixel s of J that has at least one immediate neighbor not in J
   - Let y_s be the neighborhood of radius r of the pixel s excluding the pixels belonging to J
   - Let $A(s)$ be the search area of radius w centered at s
   - For each pixel $t \in A(s) \backslash J$

     $$\rho(s,t) \leftarrow |y_s - y_t| \tag{a}$$

     ⚠ *The neighborhoods y_t are built excluding the pixels that would fall in J if x_t was translated to s. If y_t contains pixels in J other than the ones just mentioned, then it is discarded.*
   - Select the k nearest neighborhoods y_t, *i.e.*, the t_i's such that

     $$\rho(s,t_1) \leq \rho(s,t_2) \leq \cdots \leq \rho(s,t_k) \tag{b}$$

   - Perform the following update

     $$x_s \leftarrow x_{t_j} \tag{c}$$

     where j is chosen randomly in $[1..k]$ with equal probabilities
4. Remove from J all the pixels that were updated at (c). If J is not empty, then go back to step 3.

Table 10.1 – Pseudocode of the proposed inpainting algorithm.

will be replaced with a random pick among the k values x_{t_i} [Efr&Leu99, Pog08]. The algorithm is presented in Tab. 10.1.

Note that, like in denoising, the norm in Tab. 10.1-(a) is not necessarily $\mathcal{L}^2$. In practice, a radially decreasing weight was applied to the elements of y_s and y_t. An inpainting result is presented in Fig. 10.1.

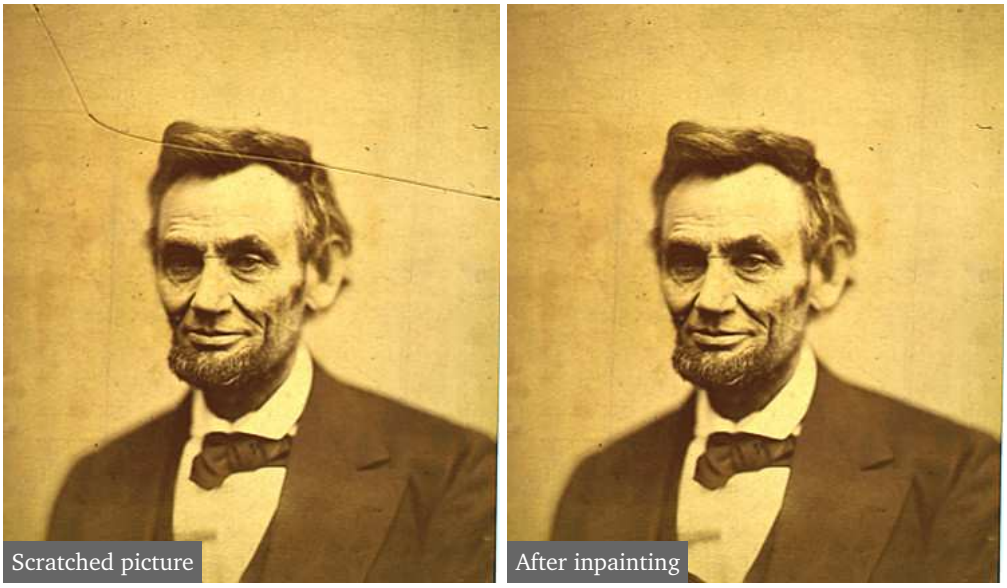

Figure 10.1 – Inpainting of image “Lincoln”.

10.2 Optical flow

10.2.1 Notations

Let f_t and f_{t+1} be two consecutive frames of a video sequence. Let D be the domain of these frames. The optical flow v is a vector field such that

$$f_t(s) = f_{t+1}(s + v_s) \tag{10.1}$$

for all s in D, a condition known as the brightness constancy. The flow v can be determined by solving the following problem

$$v = \arg\min_u \sum_{s \in D} \big(f_t(s) - f_{t+1}(s + u_s)\big)^2 \tag{10.2}$$

$$= \arg\min_u |r(u)|^2 \tag{10.3}$$

where r is the so-called residual and $|\cdot|$ is the $\mathcal{L}^2$ norm.

10.2.2 Apparent inappropriateness of entropy

The norm of the residual is equal to zero if and only if the residual is identically equal to zero while the Shannon entropy

$$H(r) = -\sum_x p_r(x) \ \log p_r(x), \tag{10.4}$$

p_r being the probability mass function (PMF) of r, is equal to zero whenever the residual is constant. Of course, the purpose is to get a residual as close to zero as possible. Therefore, apparently, the $\mathcal{L}^2$ constraint leads to the correct solution while the entropy constraint has a larger set of admissible solutions. Actually, this conclusion is valid only if getting a residual equal to zero is a sufficient condition to guarantee the optical flow problem to be solved. If the colors present in f_t are all unique and if the same colors also appear in f_{t+1}, then there is indeed a unique optical flow for which the brightness constancy is verified. However, as soon as some colors are repeated, several solutions exist. Hence, the $\mathcal{L}^2$ constraint is not sufficient. Finding the solution that is coherent with the reality requires regularization. Similarly, among the solutions with an entropy equal to zero, certainly only one is coherent with an appropriate regularization and corresponds to a residual equal to zero.

But is the $\mathcal{L}^2$ constraint even necessary? As is well-known, its fulfillment can only be fortuitous. For example, it is usually broken if some object parts in f_t are occulted in f_{t+1}, if some object parts not visible in f_t are exposed in f_{t+1}, or if the luminance changes locally or globally between f_t and f_{t+1}. This surely does not mean that defining the optical flow as the minimizer of the norm of the residual is not a correct approach. Many works testify to the contrary [Hor&Sch81, Luc&Kan81, Wei&Sch01, Bro+04, Bru+05] – to cite just a few. However, it supports attempts to propose alternatives. Minimizing the entropy of the residual is one of them.

As a final remark on entropy, note that, if for some reason the residual is expressly required to be as close to zero as possible, the entropy can be computed on a symmetrized version of the PMF/probability density function (PDF).

10.2.3 Advantages of entropy

In Section 10.2.2, it was reminded that the optical flow problem should be regularized. It was also argued that the entropy of the residual could then be employed. If a regularized solution relying on an entropy-based fidelity term happens to correspond to a residual constant but different from zero, it is reasonable to infer that the flow is correct but that a global change of luminance occurred. Indeed, the probability that the residual be constant and the optical flow be coherent with the regularization while not being close to the actual flow is very low, unless the regularization term is inadequate. Although this kind of robustness can be added to an $\mathcal{L}^2$ approach [Mol&Dub91, Neg&Yu93], it is interesting to have it by nature with the entropy.

The low sensitivity to outliers is another feature of entropy (see Section 2.1.2) which is necessary in optical flow. It will be illustrated on a synthetic example. Assume the frame f_t depicts a uniform disk on a textured background. From f_t to f_{t+1}, the disk undergoes a translation. Then, a Gaussian noise with a mean equal to zero and a variance equal to 5 is added to f_{t+1}. Figure 10.2 shows the evolution

of the $\mathcal{L}^1$ norm, the $\mathcal{L}^2$ norm, and the entropy of the residual as a function of the translation extent when the correct optical flow is known. As expected, the $\mathcal{L}^2$ constraint is too strong when there are outliers while the $\mathcal{L}^1$ constraint is much more tolerant. Still, the penalty for the maximum number of outliers is around for times higher than the penalty when there are none. In contrast, the increase of the entropy is around 12 %.

As a conclusion, it seems realistic to investigate the use of entropy for optical flow computation. For example, the entropy of the residual could serve as data fidelity and the entropy of the flow combined with some spatial constraint (see Tab. 2.1) could act as a regularization term.

10.3 Content-based indexing and retrieval

10.3.1 Context

The challenge in content-based indexing and retrieval is to summarize an image into discriminative pieces of information that can be looked for in other images. Given a query image, the images of a database can then be ordered decreasingly in terms of how well their summarized description matches the one of the query. There are at least three points of view to reduce the amount of data represented by the image pixels into a subset informative for the present task. *(i)* A global analysis accounting for the whole image but without much precision can be performed, leading, *e.g.*, to color histograms [Szu&Pic98] or histograms of edge direction [Vai+01]. Such approaches are mostly targeted to image categorization for generic classes such as indoor vs. outdoor. *(ii)* A local, detailed analysis of a few selected areas can be done. For example, salient points or salient regions can be extracted and associated with local features or segmented areas [Li&Wan03, Laz+06, Mez+06, Zha&Izq06]. By aggregating information about the spatial arrangement of features, a notion of global description is added [Laz+06, Mez+06] *(iii)* Finally, instead of attempting to find sparse, representative elements in the image domain, the key information can also be derived in a transformed domain chosen for its ability to concentrate prominent areas into a few points [Do&Vet02]. The transformation corresponds to a global analysis of the image. Then, it is followed by a detailed interpretation. Let us take this approach.

10.3.2 Sparse multiscale patches

The wavelet transform provides a sparse representation of images in the sense that it concentrates the informational content into a few coefficients of large amplitude while the rest of the coefficients are small. These large coefficients together with the dependencies that exist between them are characteristic of structures present in an image and can be exploited, *e.g.*, for image enhancement [Rom+01, Por+03]. In particular, patches, or neighborhoods, of wavelet coefficients can be used [Por+03].

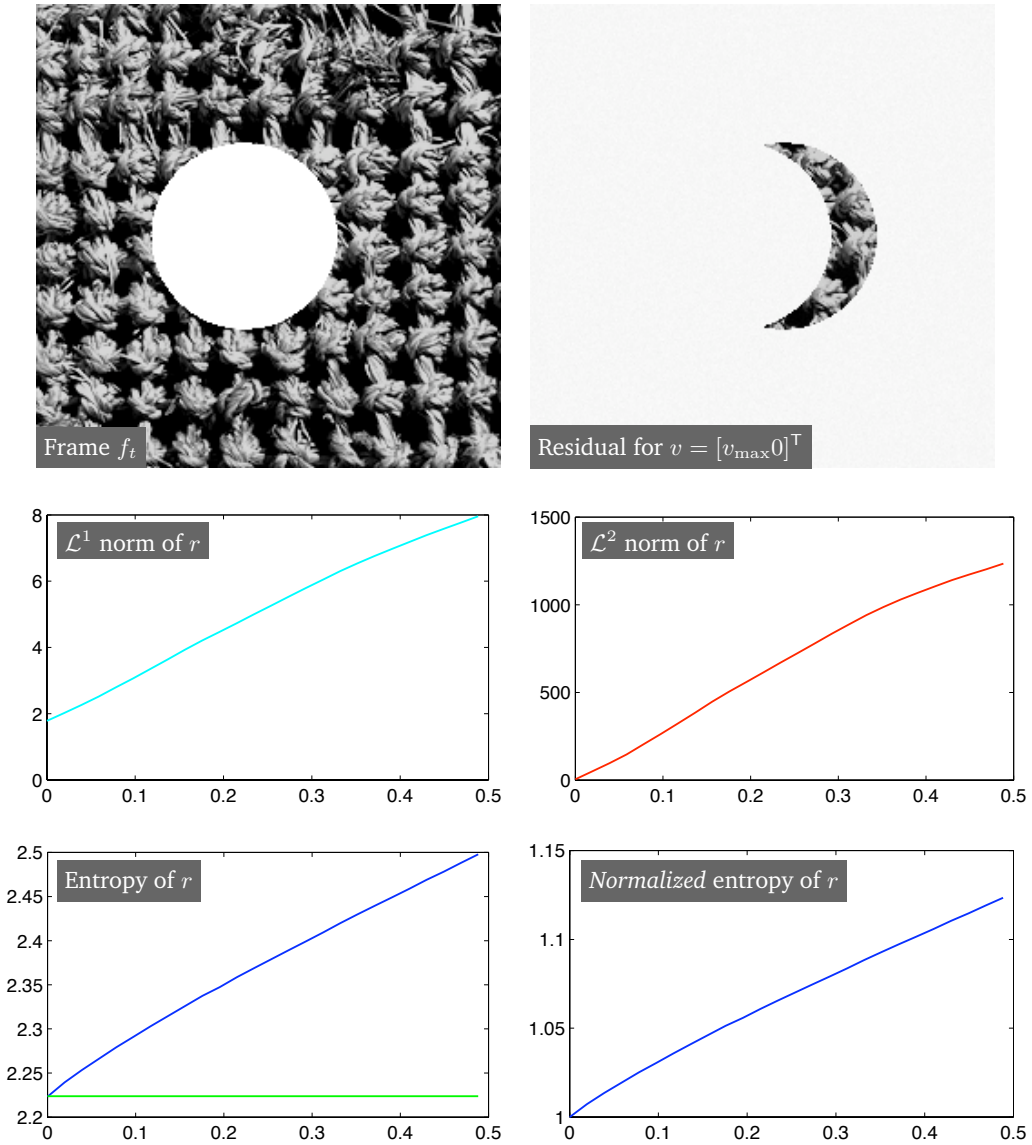

Figure 10.2 – $\mathcal{L}^1$ norm, $\mathcal{L}^2$ norm, and entropy of the residual r as a function translation. The disk was successively translated horizontally by values v ranging from 0 to $v_{\max}$ which is equal to half the radius of the disk. The background was fixed. The $\mathcal{L}^1$ and $\mathcal{L}^2$ norms were divided by the number of pixels of the frame. On the entropy diagram, the entropy of the noise added to f_{t+1} was plotted in green. The normalized entropy refers to the entropy of the residual divided by the entropy when the disk is not translated (*i.e.*, the entropy of the noise). The horizontal axes of the diagrams represent the translation divided by the radius of the disk.

Following this philosophy, neighboring coefficients of the Laplacian pyramid of an image [Bur&Ade83] can be grouped together to form multiscale patches

$$W_{\sigma,s} = (w_{\sigma,s}, w_{\sigma,s\pm e_1}, w_{\sigma,s\pm e_2}, w_{\sigma-1,s}) \tag{10.5}$$

where $w_{\sigma,s}$ is the coefficient at scale σ and intrascale location s, the scale $\sigma - 1$ is coarser than σ, and (e_1, e_1) is the canonical basis of $\mathbb{R}^2$. If dealing with color images, the patches $W^c_{\sigma,s}$, $c \in [1..3]$, are combined together, resulting in interscale, intrascale, interchannel patches of dimension 18. By selecting only significant patches,[#2] a sparse subset is obtained [Ant+08, Pir+08a, Pir+08b, Pir+08c]. Such a description can be completed with $3\times3\times3$-pixel patches[#3] z of the low-frequency approximation of the image at the coarser scale of the decomposition. Then, a selection of meaningful patches is made according to a strategy similar to the ones mentioned for the Laplacian coefficients. In conclusion, the image is summarized by the two sets of so-called Sparse Multiscale Patches (SMPs) $\{W_{\sigma,s}, \sigma, s\}$ and $\{z_t, t\}$.

The PDFs of such patches at each scale σ has proved to characterize spatial structures in images [Por+03, Pie+05].

10.3.3 Similarity measure

Let I_1 and I_2 be two images visually similar. For example, they may represent different views of the same scene, they may contain similar objects (potentially at different locations)... Despite their similarity, there is in general no strict or even loose geometric correspondence between the SMPs of each image. As a consequence, a notion of residual is unlikely to adequately assess whether I_1 and I_2 are similar. In contrast, accounting for the presence and frequency of occurrence in I_2 of informative local structures detected in I_1 appears to be reasonable.[#4] Hence, relying also on the last remark of Section 10.3.2, the similarity between two images is defined as the closeness between their respective SMP PDFs at corresponding scales. Actually, the Kullback-Leibler divergence, a measure of dissimilarity, is used at each scale. This measure already showed good performances in the context of image retrieval [Puz+99]. Formally, the similarity between I_1 and I_2 is expressed as follows

$$S(I_1, I_2) = \sum_{\sigma} \alpha_\sigma \, \mathfrak{D}_{\mathrm{KL}}(f_{1,\sigma}, f_{2,\sigma}) + \alpha \, \mathfrak{D}_{\mathrm{KL}}(g_1, g_2) \tag{10.6}$$

[#2] Different selection strategies are possible: retain only the patches whose central coefficient $w_{\sigma,s}$ is higher than a given threshold, retain only the patches whose $\mathcal{L}^2$ norm is higher than a given threshold, fix the threshold of one of the previous strategies so as to retain a given number of patches...

[#3] $3\times3\times3$ = patch width $\times$ patch height $\times$ number of color channels

[#4] On this aspect, the philosophy is similar to the idea behind the bags of words [Siv&Zis03]. However, no quantization or weighting (*e.g.*, relying on the frequency of occurrence of specific SMPs among the images of the database) is performed on the SMPs.

where the α_σ's and α are positive constants, $f_{i,\sigma}$ is the SMP PDF of the high-frequency details of I_i at scale σ, and g_i is the PDF of the low-frequency patches of I_i. As was seen in Section 5.2.3, the Kullback-Leibler divergence, and then S, can be estimated directly from the sets of samples $\{W_{\sigma,s}, \sigma, s\}$ and $\{z_t, t\}$ using the k nearest neighbor (kNN) framework [Ant+08, Pir+08a].

The same (dis)similarity measure can be used for image categorization [Pir+08c], which consists in labeling a query image according to a set of predefined classes or categories. This process is composed of two steps. First, training aims at determining one prototype per given image category. Each category is illustrated by a set of images. The set of all the images for every category is called a labeled training set. The prototype P_c of the category C can be chosen as the image of C which minimizes the sum of the dissimilarities to all the other images of C (very much like the medoid of a cluster)

$$P_c = \arg\min_{J \in C} \sum_{I \in C} S(I, J) \ . \tag{10.7}$$

Then, classification denotes the process of allowing the database to grow with new, unlabeled images by assigning to them the label of their closest prototype.

CONCLUSION

Chapter 11

Summary and final remarks

11.1 kNN-based variational approach

If the suggestions made in this document were to be turned into a recipe (among others) to attempt to solve a particular image or video processing task, it could look like this:

1. Identify key features of the image or video; For example,
   - the pixel colors,
   - local derivatives,
   - coefficients in a transformed domain,
   - groups of previously mentioned quantities,
   - ...
2. Search for a statistical/information-theoretic interpretation of the problem; In particular, the features should be assumed to follow a random law;
3. Express the solution as the optimum of an appropriate (dis)similarity measure/energy; The rest of the recipe should be applicable if the heart of this measure is composed of:
   - means or expected values,
   - variances,
   - probability density functions (PDFs),
   - entropies,
   - relative entropies or divergences,
   - mutual information,
   - and, more generally but speculatively, measures tightly linked to local densities of samples – see note below.

4. Consider the feature instances that can be extracted in the image or video as samples;
5. Determine a k nearest neighbor (kNN) estimate of the measure, *i.e.*, an approximation depending directly on the samples with adaptability to their local density in the feature space;
6. Determine the kNN derivative of the measure; Part of the corresponding development might be easier or valid only in the continuous framework;
    - If this step involves active contours, the notion of shape derivative should help,
    - If the measure involves logarithms of PDFs, the mean shift and its kNN version should be useful.
7. Perform a gradient descent.

Such a procedure skeleton is neither original nor unusual. What this document aimed at highlighting was the use of two concepts, the shape gradient (should the minimization be dependent on deforming interfaces) and the kNN framework, in order to try to systematize and to simplify the resolution of some image and video processing tasks.

Note that the general statement made about the kind of measures that can be dealt with in the kNN framework is purely speculative since no evidence was provided aside from differential entropy (whether joint, conditional, or cross-entropy) and the Kullback-Leibler divergence. Yet, the clear relationship between the notion of k nearest neighbors and the local density of samples leads to the intuition that a kNN approximation could be found for a measure which, at its lowest level of interpretation, depends on local densities.

11.2 A note on the metric of the feature space

As mentioned in Section 6.2, the distance between features is a topic of concern. It may seem appropriate to define this distance, once and for all, based on the nature of the features. Instead, a suitable definition could follow from the actual distribution of the samples or from a distribution learned beforehand. For example, in the kNN framework, the isotropic neighborhood can be replaced with an ellipsoid [Ter&Sco92]. To illustrate the idea that sample distribution can help define an adequate distance, a simple clustering experiment is proposed hereafter.

Let the features be represented by points of $\mathbb{R}^2$. Figure 11.1 shows 200 samples drawn according to two given distributions (corresponding to two classes) and three clusterings made using different metrics: $\mathcal{L}^2$-k-means, $\mathcal{L}^2$-k-medoids, and a local density-based k-means. For all clusterings, the same two seeds were used. They were conveniently placed within each actual cluster.

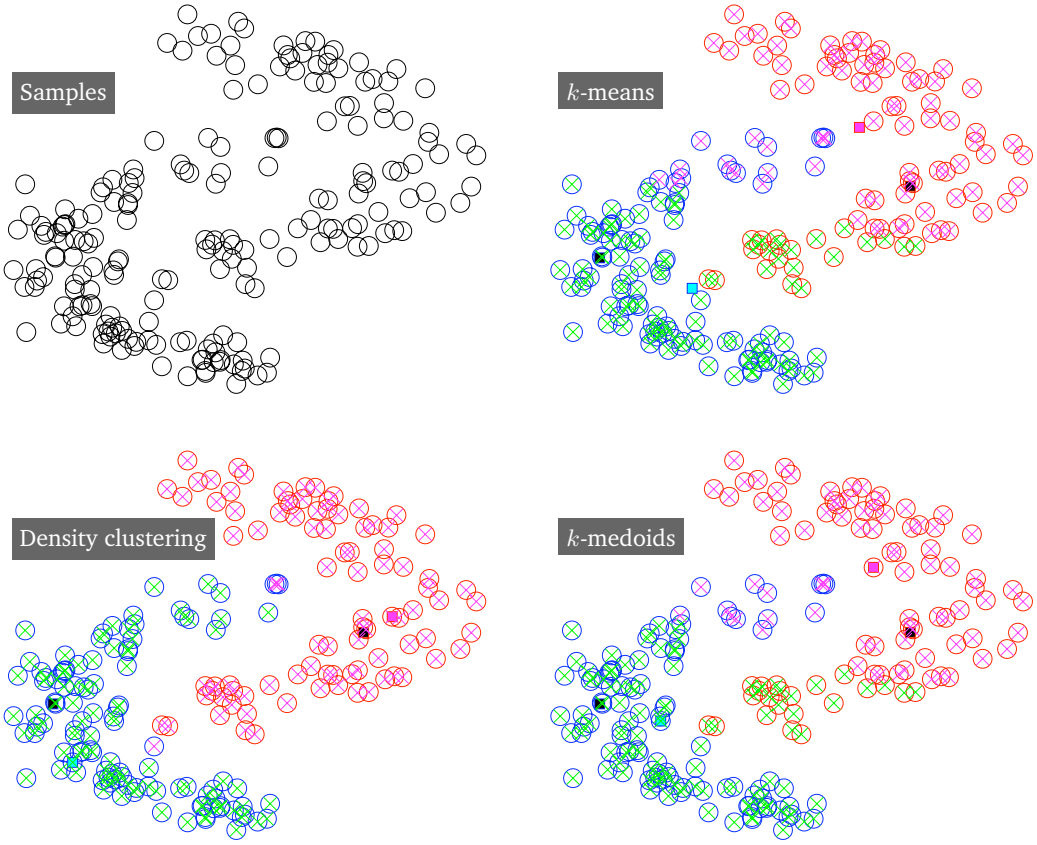

Figure 11.1 – Influence of the metric used for clustering. The purpose of the black&white image is to show that 2 clusters can be visually distinguished. In the other 3 images, red and blue circles represent the ground truth clustering, magenta and green crosses represent the corresponding computed clusters, the black squares represent the clustering seeds, and the magenta and the green squares represent the final cluster prototypes.

The algorithm of k-medoids is a variant of k-means. In k-means, the prototype of a cluster is its centroid. In the variant, the centroid is replaced with the medoid, *i.e.*, the cluster element which minimizes the sum of distances to all other elements of the cluster. Both versions were implemented using the $\mathcal{L}^2$ norm.

Finally, a density clustering is proposed. It relies on a metric defined locally using distances to nearest neighbors among all the samples, *i.e.*, considering the samples of all the classes without distinction. The idea was that if the straight path between two samples crosses a low density area, the samples probably belong to different clusters. Actually, the notion of straight path is adapted to convex clusters only. In general, it can be replaced with a path which goes through the shortest cumulated length of low density areas among all possible paths. Equivalently, this path can be viewed as a geodesic, *i.e.*, a shortest path in a modified metric.

For practical reasons, the potential paths were restricted to sequences of edges of the complete graph of the samples. A consequence of this constraint is that the prototype of a cluster was defined as the medoid of the cluster in terms of the specific metric. The metric was defined as a weighted $\mathcal{L}^2$ norm. At x, the weight $w(x)$ was chosen inversely proportional to the local density, which can be inferred by the distance $\rho_k(x)$ of x to its k-th nearest neighbor among the samples. To make the weight a dimensionless quantity, $\rho_k(x)$ was divided by $\rho_1(x)$. To emphasize the effect of low densities, the exponential was taken. However, to maintain the condition $w(x) = 1$ when $\rho_k(x) = \rho_1(x)$,[#1] the final expression was

$$w(x) = \exp\left(\frac{\rho_k(x)}{\rho_1(x)} - 1\right) . \tag{11.1}$$

In this metric, the distance between x and y (*i.e.*, the weight of the edge linking x and y if they are vertices of the complete graph mentioned earlier) is computed by weighting the $\mathcal{L}^2$ norm with $w(t)$, $t \in [x, y]$.[#2] The presented result was obtained with $k = 2$. Apart from the metric, the density clustering algorithm was identical to k-medoids.

The density clustering performed much better than k-means and k-medoids. Yet, its modified metric does not account for the nature of the features but only for their distribution. One might object that a specific distribution might be a consequence of the nature of the features. In part, this is true. For example, histograms normalized so that the sum of their elements is equal to one lie on the subset of the $\mathcal{L}^1$-ball of radius 1 with positive coordinates. However, a given experimental context certainly implies a particular distribution within this subset. In conclusion, if the samples are unevenly distributed in the feature space, the actual distribution may have more importance than the nature of the features.

Note that this experiment was made to illustrate an intuition and might be classical in clustering.

[#1] Whether this is necessary or not has not been investigated.

[#2] The use of a weighted graph to perform clustering follows approaches such as the ones proposed in Isomap [Ten+00], random walk-based clustering [Yen+05], and, more generally, spectral clustering [Shi&Mal00, Bel&Niy01, Nad+05]. However, there are significant differences between the weighted graphs in these methods and the one employed here. For example, a complete graph is built here while the methods cited above build *neighborhood* graphs. More importantly, in these methods, the edge weights represent the similarity between vertices, often directly related to the $\mathcal{L}^2$ length of the edge. Instead, it is proposed to assign weights linked to the local sample density along the edges. As a consequence, an edge longer than another one in the $\mathcal{L}^2$ sense may get a smaller weight. In any case, the purpose of this section is not to compare clustering methods but rather to illustrate the fact that local sample density can be used to define a metric which, when used in place of the $\mathcal{L}^2$ norm (or maybe another feature-adapted norm) in a classical clustering algorithm, can produce good results. In that respect, the interpretation of the proposed density clustering as a local density, kNN-based method is straightforward.

APPENDICES

Appendix A

Shape derivative and active contour

A.1 A variational approach to segmentation

Image or video segmentation can be performed with the following variational approach: the problem is formulated as the minimization of an energy depending on assumed characteristics of the objects of interest [Mum&Sha89, Cas+97, Che+99, Cre&Soa03, Jeh+03, Li+04, Li&Yez05]. For simplicity, it is supposed that there is a unique object. Typically, the energy is a sum of a domain integral and a boundary integral [Mum&Sha89, Par&Der99, Gas+04]

$$E(\Gamma) = \int_{\Omega} \phi_f(\Gamma, x)\, \mathrm{d}x + \int_{\Gamma} \varphi_f(\Gamma, s)\, \mathrm{d}s \tag{A.1}$$

where Ω is an open set of $\mathbb{R}^2$, Γ is the oriented boundary $\partial\Omega$ of Ω, s is the arc-length parameterization of Γ, and f is the image or video to be segmented.[#1] The energy (A.1) is designed to have a unique global minimum[#2] at $\Omega^\star$, the domain of the object of interest. The function ϕ is sometimes referred to as the descriptor of the object. For example, if ϕ is equal to

$$\phi(\Gamma, x) = (f(x) - \mu(\Gamma))^2 \tag{A.2}$$

where $\mu(\Gamma)$ is the average value of f in Ω, then ϕ is equal to zero on Ω if and only if f is constant on Ω. Therefore, ϕ is a descriptor of objects of constant intensity. Similarly, φ is the descriptor of the object boundary.

[#1] For convenience, ϕ and φ will be used instead of ϕ_f and φ_f.

[#2] A problem of maximization is trivially turned into a problem of minimization.

If the object background can also be characterized, then the energy can be *symmetrized* as follows

$$E(\Gamma) = \int_{\Omega} \phi(\Gamma, x)\, \mathrm{d}x + \int_{\Gamma} \varphi(\Gamma, s)\, \mathrm{d}s + \int_{\Omega^c} \phi^c(\Gamma^c, x)\, \mathrm{d}x \tag{A.3}$$

where Ω^c is the complement $D\backslash\bar{\Omega}$ of Ω in the image or frame domain D. This combination of an integral on Ω and an integral on Ω^c is called region competition [Yez+99, Deb+01, Cre&Sch02]. Nevertheless, in the following developments, the last integral of (A.3) will be discarded since its handling is similar to the one of the first integral.

A.2 Active contour

A possible method to minimize the energy (A.1) is to define an initial contour and to deform it iteratively in such a way that its energy decreases from one iteration to the next. The contour eventually converges toward a (possibly local) minimizer. This process is known as active contour [Kas+88, Cas+93, Cas+97, Cha&Ves01]. When ϕ and φ do not depend on Γ, the minimization procedure can be based on a local strategy involving energy evaluations only [Ger&Ref96]. Otherwise, the influences of local deformations are linked together and the contour deformation should be performed at once. The appropriate deformation can be derived from the derivative of the energy with respect to the contour [Kas+88, Cas+93, Cas+97].

A.3 Shape derivative and evolution equation

The derivative of energy (A.1) with respect to Γ can be obtained by a calculus of variations [Cas+97, Aub+03]. However, if the descriptor ϕ or φ depends on Γ, this can be complex. Some studies on shape gradients [Sch92, Sok&Zol92, Del&Zol01, Jeh+03] offer a convenient and general basis for this differentiation. The shape derivative of (A.1) is a function $\mathrm{d}E(\Gamma, V)$ of Γ and a velocity V defined on Ω but restricted to Γ. For a given Γ, the velocity should be chosen such that the shape derivative is negative, thus indicating a way to deform Γ in order to decrease its energy. As noted by [Cha+05], the straightforward choice is to take the opposite of the gradient of (A.1) associated with the $\mathcal{L}^2$ inner product on Γ. However, one might want to use other descent directions, for example, to improve the convergence rate [Hin&Rin03], to increase spatial coherence and avoid as much as possible to converge to irrelevant local minima [Cha+05], to simplify implementation [Ove&Sol05, Sol&Ove05], or to respect the principles of an underlying physical model or for improved stability and convergence rate [Doc+05]. These alternatives may result from *(i)* designing other inner products or from *(ii)* directly designing

descent directions V verifying [Ove&Sol05, Sol&Ove05]

$$\mathrm{d}E(\Gamma, V) \leq 0 \; . \tag{A.4}$$

In any case, the active contour evolution equation has the form

$$\frac{\partial \Gamma}{\partial \tau} = V(\tau) \; . \tag{A.5}$$

Appendix B

General expressions of the shape derivative

B.1 Boundary energy

Let us consider the following boundary energy

$$E(\Gamma) = \int_\Gamma \varphi(s)\ \mathrm{d}s \tag{B.1}$$

where Γ is the oriented boundary $\partial\Omega$ of an open set Ω of $\mathbb{R}^2$ and s is the arc-length parameterization of Γ. The shape derivative of (B.1) is equal to [Sok&Zol92, Del&Zol01]

$$\mathrm{d}E(\Gamma, V) = \int_\Gamma \left(\frac{\partial \varphi(s)}{\partial N} - \varphi(s)\ \kappa(s) \right)\ N(s) \cdot V(s)\ \mathrm{d}s \tag{B.2}$$

where N is the inward unit normal of Γ and κ is the curvature of Γ.

Note that this result can also be obtained by a calculus of variations [Cas+97].

B.2 Domain energy

Let Ω be an open set of $\mathbb{R}^2$ and let T be a transformation of Ω such that

$$\begin{cases} \Omega = T(\tau = 0, \Omega) \\ \Omega_T(\tau) = T(\tau, \Omega) \\ x_T(\tau) = T(\tau, x),\ x \in \Omega \end{cases} . \tag{B.3}$$

The deformation at x is defined as

$$V_T(x) := \lim_{\tau \to 0} \frac{x_T(\tau) - x}{\tau} \tag{B.4}$$

$$= \frac{\partial T}{\partial \tau}(0, x)\ . \tag{B.5}$$

For clarity, $\Omega_T(\tau)$, $x_T(\tau)$, and $V_T(x)$ are referred to as $\Omega(\tau)$, $x(\tau)$, and $V(x)$, respectively. The vector field V is called the velocity of Ω.

Let us consider the following domain energy

$$E(\Gamma) = \int_{\Omega} \phi(\Gamma, x)\, \mathrm{d}x \tag{B.6}$$

where Γ is the oriented boundary $\partial\Omega$ of Ω. The energy of the transformed domain $\Omega(\tau)$ is equal to

$$E(\Gamma, T, \tau) = \int_{\Omega(\tau)} \phi(\Gamma(\tau), x)\, \mathrm{d}x\ . \tag{B.7}$$

Then, the shape derivative of (B.6) is defined as

$$\mathrm{d}E(\Gamma, T) := \lim_{\tau \to 0} \frac{E(\Gamma, T, \tau) - E(\Gamma, T, 0)}{\tau}\ . \tag{B.8}$$

It is equal to [Sok&Zol92, Del&Zol01]

$$\mathrm{d}E(\Gamma, T) = \int_{\Omega} \left.\frac{\partial \phi(\Gamma(\tau), x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x - \int_{\Gamma} \phi(\Gamma, s)\ N(s) \cdot V(s)\ \mathrm{d}s \tag{B.9}$$

where s is the arc-length parameterization of Γ and N is the inward unit normal of Γ. Since V appears explicitly in the shape derivative expression, $\mathrm{d}E(\Gamma, T)$ can be replaced with $\mathrm{d}E(\Gamma, V)$.

Note that this result can also be obtained by a calculus of variations [Aub+03].

Appendix C

Rewriting the shape derivative as a boundary integral

Two (sets of) conditions are proposed to allow practical rewriting of the shape derivative (B.9) into an expression without any domain integral.

C.1 Recursive applications of the shape derivative

Under some conditions on the dependency of ϕ on Γ, applying recursively (B.9) to its first integral leads to an expression containing no domain integral [Jeh+03]

$$\mathrm{d}E(\Gamma, V) = -\int_{\Gamma} \Psi(\Gamma, s)\ N(s) \cdot V(s)\ \mathrm{d}s\ . \tag{C.1}$$

Let us assume that

$$\begin{cases} \phi_i(\Gamma, x) = \phi_i(g_i(\Gamma), x), i \in [1, n-1] \\ g_i(\Gamma) = \int_{\Omega} \phi_{i+1}(\Gamma, x)\ \mathrm{d}x, i \in [1, n-1] \\ \phi_n(\Gamma, x) = \phi_n(x) \end{cases} \tag{C.2}$$

where $\phi_1 = \phi$ (note that g_i has the same form as (B.6)). The first integral of (B.9) reads

$$\begin{aligned}
\int_\Omega \left.\frac{\partial \phi_1(g_1(\Gamma(\tau)),x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x &= \int_\Omega \left.\frac{\partial \phi_1(k,x)}{\partial k}\right|_{g_1(\Gamma)} \left.\frac{\partial g_1(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \mathrm{d}x && \text{(C.3)}\\
&= \left.\frac{\partial g_1(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \int_\Omega \left.\frac{\partial \phi_1(k,x)}{\partial k}\right|_{g_1(\Gamma)} \mathrm{d}x && \text{(C.4)}\\
&:= \mathrm{d}g_1(\Gamma,V) \int_\Omega \left.\frac{\partial \phi_1(k,x)}{\partial k}\right|_{g_1(\Gamma)} \mathrm{d}x && \text{(C.5)}\\
&= \mathrm{d}g_1(\Gamma,V)\; A_1(\Gamma) && \text{(C.6)}
\end{aligned}$$

where $\mathrm{d}g_1(\Gamma,V)$ is equal to

$$\mathrm{d}g_1(\Gamma,V) = \int_\Omega \left.\frac{\partial \phi_2(\Gamma(\tau),x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x - \int_\Gamma \phi_2(\Gamma,s)\; N(s)\cdot V(s)\; \mathrm{d}s\;. \quad \text{(C.7)}$$

Then, the development leading to (C.6) can be repeated with the successive domain integrals present in $\mathrm{d}g_i(\Gamma,V)$, i increasing, until

$$\mathrm{d}g_{n-1}(\Gamma,V) = -\int_\Gamma \phi_n(s)\; N(s)\cdot V(s)\; \mathrm{d}s \quad \text{(C.8)}$$

which does not contain any domain integral since ϕ_n is independent of Γ. Gathering the successive boundary integrals together, the shape derivative is equal to

$$\begin{aligned}
\mathrm{d}E(\Gamma,V) &= -\int_\Gamma \Psi(\Gamma,s)\; N(s)\cdot V(s)\; \mathrm{d}s && \text{(C.9)}\\
&= -\int_\Gamma \left(\sum_{i=1}^{n} \phi_i(g_i(\Gamma),s) \prod_{j=1}^{i-1} A_j(\Gamma)\right) N(s)\cdot V(s)\; \mathrm{d}s && \text{(C.10)}
\end{aligned}$$

where $A_j(\Gamma)$ is equal to

$$A_j(\Gamma) = \int_\Omega \left.\frac{\partial \phi_j(x,k)}{\partial k}\right|_{g_j(\Gamma)} \mathrm{d}x \quad \text{(C.11)}$$

with the following convention

$$\phi_n(g_n(\Gamma), s) = \phi_n(s) \ . \tag{C.12}$$

For example, ϕ may involve the variance g_1 of f in Ω. The variance involves the average value g_2 of f in Ω. Finally, the average value involves f but no terms depending on Γ.

C.2 Domain integral equal to zero

Under some conditions, Ψ in (C.1) is simply equal to ϕ [Roy+06]. In other words, the first integral of (B.9) is equal to zero. For example, if ϕ is given by (A.2), *i.e.*, $(f(x) - \mu(\Gamma))^2$, the integrand of the first integral of (B.9) is equal to

$$\left.\frac{\partial \phi(\Gamma(\tau), x)}{\partial \tau}\right|_{\tau=0} = \left.\frac{\partial \left((f(x) - \mu(\Gamma(\tau)))^2\right)}{\partial \tau}\right|_{\tau=0} \tag{C.13}$$

$$= -2\,(f(x) - \mu(\Gamma))\ \left.\frac{\partial \mu(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \tag{C.14}$$

where $\mu(\Gamma)$ can be written as follows

$$\mu(\Gamma) = \int_\Omega f(x)\ \mathrm{d}x \ / \int_\Omega \mathrm{d}x \ . \tag{C.15}$$

Then, the first integral of (B.9) is equal to

$$\int_\Omega \left.\frac{\partial \phi(\Gamma(\tau), x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x = -2\ \left.\frac{\partial \mu(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \int_\Omega (f(x) - \mu(\Gamma))\ \mathrm{d}x \tag{C.16}$$

$$= -2\ \left.\frac{\partial \mu(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \left(\int_\Omega f(x)\ \mathrm{d}x - \mu(\Gamma) \int_\Omega \mathrm{d}x\right) \tag{C.17}$$

$$= 0 \ . \tag{C.18}$$

In general, a sufficient condition for the first integral of (B.9) to be equal to zero is

$$\begin{cases} \phi(\Gamma, x) = \phi(g(\Gamma), x) \\ g(\Gamma) = \arg\min_k \int_\Omega \phi(k, x)\ \mathrm{d}x \end{cases} . \tag{C.19}$$

In other words, $g(\Gamma)$ is the minimizer of (B.6) seen as a function of g with Γ fixed. For convenience, the following notation will be used

$$g(\Gamma) = \arg\min_k E_\Gamma(k) \ . \tag{C.20}$$

If g is assumed to be differentiable, the development of (C.6) can be continued using the new condition (C.19)

$$\int_\Omega \left.\frac{\partial \phi(g(\Gamma(\tau)),x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x = \mathrm{d}g(\Gamma,V) \int_\Omega \left.\frac{\partial \phi(k,x)}{\partial k}\right|_{g(\Gamma)} \mathrm{d}x \tag{C.21}$$

$$= \mathrm{d}g(\Gamma,V)\, \frac{\partial}{\partial k} \int_\Omega \phi(k,x)\, \mathrm{d}x \Big|_{g(\Gamma)} \tag{C.22}$$

$$= \mathrm{d}g(\Gamma,V)\, \left.\frac{\partial E_\Gamma(k)}{\partial k}\right|_{g(\Gamma)} . \tag{C.23}$$

If $g(\Gamma)$ is a constraint-free minimizer of E_Γ, then it can be concluded immediately that (C.23) is equal to zero. Otherwise, let us assume that $g(\Gamma)$ is the minimizer of E_Γ under the constraint

$$\xi(g(\Gamma)) = 0\,. \tag{C.24}$$

Then, there exists a Lagrange multiplier λ such that

$$\left.\frac{\partial E_\Gamma(k)}{\partial k}\right|_{g(\Gamma)} = \lambda\, \left.\frac{\partial \xi(k)}{\partial k}\right|_{g(\Gamma)} . \tag{C.25}$$

Therefore,

$$\int_\Omega \left.\frac{\partial \phi(g(\Gamma(\tau)),x)}{\partial \tau}\right|_{\tau=0} \mathrm{d}x = \lambda\, \left.\frac{\partial g(\Gamma(\tau))}{\partial \tau}\right|_{\tau=0} \left.\frac{\partial \xi(k)}{\partial k}\right|_{g(\Gamma(\tau=0))} \tag{C.26}$$

$$= \lambda\, \left.\frac{\partial \xi(g(\Gamma(\tau)))}{\partial \tau}\right|_{\tau=0} . \tag{C.27}$$

By definition, for any τ, $\xi(g(\Gamma(\tau)))$ has the same value (equal to zero). It can be concluded that (C.27) is equal to zero.

As brought to our attention [Ano07], noting that $E(\Gamma)$ can be written as $F(\Gamma, g(\Gamma))$ leads to an immediate proof of the result of Section C.2.

Appendix D

Denoising energy derivative

Let $\tilde{y}_i$ be the set of colors of a noisy neighborhood extracted around the pixel of index i in a noisy image. Let D be the set of pixel indices within the image domain. Let $D_{\tilde{y}_i}$ be the subset of D of the indices of the pixels whose neighborhood is equal to $\tilde{y}_i$. Let X be the random variable modeling pixel color and let Y be the random vector modeling neighborhood colors. The entropy of X conditional on Y being equal to $\tilde{y}_i$ can be approximated by the Ahmad-Lin estimator [Ahm&Lin76]

$$h(X|Y=\tilde{y}_i) \approx -\frac{1}{|D_{\tilde{y}_i}|} \sum_{s \in D_{\tilde{y}_i}} \log f_{x|y}(x_s|\tilde{y}_i) \tag{D.1}$$

where

$$f_{x|y}(x|y) = \frac{1}{|D_y|} \sum_{t \in D_y} K(x - x_t) \; . \tag{D.2}$$

Therefore,

$$h(X|Y=\tilde{y}_i) = -\frac{1}{|D_{\tilde{y}_i}|} \sum_{s \in D_{\tilde{y}_i}} \log \left[\frac{1}{|D_{\tilde{y}_i}|} \sum_{t \in D_{\tilde{y}_i}} K(x_s - x_t) \right] \; . \tag{D.3}$$

The derivative of (D.3) with respect to the pixel color x_i is equal to

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_i} = -\frac{1}{|D_{\tilde{y}_i}|} \sum_{s \in D_{\tilde{y}_i}} \frac{1}{f_{x|y}(x_s|\tilde{y}_i)} \frac{1}{|D_{\tilde{y}_i}|} \sum_{t \in D_{\tilde{y}_i}} \frac{\partial K(x_s - x_t)}{\partial x_i} \; . \tag{D.4}$$

The last term in (D.4) has the following expression

$$\frac{\partial K(x_s - x_t)}{\partial x_i} = \begin{cases} (1 - \delta_{t-i}) \nabla K(x_i - x_t) & \text{if } s = i \\ -\delta_{t-i} \nabla K(x_s - x_t) & \text{otherwise.} \end{cases} \tag{D.5}$$

Then,

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_i} = \underbrace{-\frac{1}{|D_{\tilde{y}_i}|}\frac{1}{f_{x|y}(x_i|\tilde{y}_i)}\frac{1}{|D_{\tilde{y}_i}|}\sum_{\substack{t\in D_{\tilde{y}_i}\\ t\neq i}}\nabla K(x_i-x_t)}_{s=i} + \underbrace{\mathcal{A}}_{s\neq i} \tag{D.6}$$

$$= -\frac{1}{|D_{\tilde{y}_i}|}\frac{1}{f_{x|y}(x_i|\tilde{y}_i)}\frac{1}{|D_{\tilde{y}_i}|}\sum_{t\in D_{\tilde{y}_i}}\nabla K(x_i-x_t)+\mathcal{A} \tag{D.7}$$

$$= -\frac{1}{|D_{\tilde{y}_i}|}\frac{1}{f_{x|y}(x_i|\tilde{y}_i)}\frac{1}{|D_{\tilde{y}_i}|}\nabla\sum_{t\in D_{\tilde{y}_i}}K(x_i-x_t)+\mathcal{A} \tag{D.8}$$

$$= -\frac{1}{|D_{\tilde{y}_i}|}\frac{\nabla f_{x|y}(x_i|\tilde{y}_i)}{f_{x|y}(x_i|\tilde{y}_i)}+\mathcal{A} \tag{D.9}$$

where

$$\mathcal{A} = \frac{1}{|D_{\tilde{y}_i}|^2}\sum_{\substack{s\in D_{\tilde{y}_i}\\ s\neq i}}\frac{\nabla K(x_s-x_i)}{f_{x|y}(x_s|\tilde{y}_i)} \tag{D.10}$$

$$= \frac{1}{|D_{\tilde{y}_i}|^2}\sum_{s\in D_{\tilde{y}_i}}\frac{\nabla K(x_s-x_i)}{f_{x|y}(x_s|\tilde{y}_i)}\,. \tag{D.11}$$

By multiplying the numerator and the denominator of the first term in (D.9) with $f_Y(\tilde{y}_i)$, one gets

$$\frac{\nabla f_{x|y}(x_i|\tilde{y}_i)}{f_{x|y}(x_i|\tilde{y}_i)}\frac{f_Y(\tilde{y}_i)}{f_Y(\tilde{y}_i)} = \frac{\nabla f_z}{f_z}(z_i)\cdot\frac{\partial z_i}{\partial x_i} \tag{D.12}$$

where z_i is the vector obtained by concatenation of x_i and $\tilde{y}_i$. Finally, we have

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_i} = -\frac{1}{|D_{\tilde{y}_i}|}\frac{\nabla f_z}{f_z}(z_i)\cdot\frac{\partial z_i}{\partial x_i}+\mathcal{A}\,. \tag{D.13}$$

Note that, in practice, $\mathcal{A}$ will be neglected, as suggested by Section E.2 – a k nearest neighbor (kNN) approximation is also provided in Section F.2.

Let us now study the derivative of (D.3) with respect to the pixel color x_j, $j\neq i$. If $\tilde{y}_j$ happens to be equal to $\tilde{y}_i$, then j belongs to $D_{\tilde{y}_i}$ and a development similar to

the one above can be made

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_j} = -\frac{1}{|D_{\tilde{y}_i}|}\sum_{s\in D_{\tilde{y}_i}} \frac{1}{f_{x|y}(x_s|\tilde{y}_i)} \frac{1}{|D_{\tilde{y}_i}|}\sum_{t\in D_{\tilde{y}_i}} \frac{\partial K(x_s-x_t)}{\partial x_j} \tag{D.14}$$

$$= \underbrace{-\frac{1}{|D_{\tilde{y}_i}|}\frac{1}{f_{x|y}(x_j|\tilde{y}_i)}\frac{1}{|D_{\tilde{y}_i}|}\sum_{\substack{t\in D_{\tilde{y}_i}\\ t\neq j}} \nabla K(x_j-x_t)}_{s=j} + \underbrace{\mathcal{B}}_{s\neq j} \tag{D.15}$$

$$= -\frac{1}{|D_{\tilde{y}_i}|}\frac{\nabla f_{x|y}(x_j|\tilde{y}_i)}{f_{x|y}(x_j|\tilde{y}_i)} + \mathcal{B} \tag{D.16}$$

where

$$\mathcal{B} = \frac{1}{|D_{\tilde{y}_i}|^2}\sum_{\substack{s\in D_{\tilde{y}_i}\\ s\neq j}} \frac{\nabla K(x_s-x_j)}{f_{x|y}(x_s|\tilde{y}_i)} \tag{D.17}$$

$$= \frac{1}{|D_{\tilde{y}_i}|^2}\sum_{s\in D_{\tilde{y}_i}} \frac{\nabla K(x_s-x_j)}{f_{x|y}(x_s|\tilde{y}_i)}\,. \tag{D.18}$$

Since, by assumption, $\tilde{y}_i$ is equal to $\tilde{y}_j$, it can be concluded that

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_j} = -\frac{1}{|D_{\tilde{y}_i}|}\frac{\nabla f_z}{f_z}(z_j)\cdot\frac{\partial z_j}{\partial x_j} + \mathcal{B}\,. \tag{D.19}$$

Again, $\mathcal{B}$ will be neglected.

On the other hand, if $\tilde{y}_j$ is not equal to $\tilde{y}_i$, then j does not belong to $D_{\tilde{y}_i}$ and, therefore,

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_j} = 0\,. \tag{D.20}$$

In conclusion, when neglecting terms such as $\mathcal{A}$ and $\mathcal{B}$, the derivative of (D.3) is equal to

$$\frac{\partial h(X|Y=\tilde{y}_i)}{\partial x_j} = \begin{cases} -\dfrac{1}{|D_{\tilde{y}_i}|}\dfrac{\nabla f_z}{f_z}(z_j)\cdot\dfrac{\partial z_j}{\partial x_j} & \text{if } j\in D_{\tilde{y}_i} \\ 0 & \text{otherwise.} \end{cases} \tag{D.21}$$

The energy involved in (8.4) is, up to a multiplicative constant,

$$\mathcal{E} = \sum_{s\in D} h(X|Y=\tilde{y}_s)\,. \tag{D.22}$$

The image domain D can be partitioned into n subdomains D_{t_i}, $i \in [1..n]$, corresponding to distinct noisy neighborhoods $\tilde{y}_{t_i}$

$$\mathcal{E} = \sum_{i=1}^{n} \sum_{s \in D_{t_i}} h(X|Y = \tilde{y}_s) \tag{D.23}$$

$$= \sum_{i=1}^{n} \sum_{s \in D_{t_i}} h(X|Y = \tilde{y}_{t_i}) \tag{D.24}$$

$$= \sum_{i=1}^{n} |D_{t_i}| \, h(X|Y = \tilde{y}_{t_i}) \, . \tag{D.25}$$

Combining (D.21) and (D.25), it can be concluded that the derivative of (D.22) with respect to x_j is equal to

$$\frac{\partial \mathcal{E}}{\partial x_j}(x_j) = -\frac{\nabla f_z}{f_z}(z_j) \cdot \frac{\partial z_j}{\partial x_j} \, . \tag{D.26}$$

Appendix E

Derivative of the Kullback-Leibler divergence

E.1 Expression

The Kullback-Leibler divergence is equal to

$$\mathfrak{D}_{\mathrm{KL}}(f_{T_\varphi}, f_R) = H^{\times}(f_{T_\varphi}, f_R) - H(f_{T_\varphi}) \tag{E.1}$$

where the cross entropy $H^{\times}(f_{T_\varphi}, f_R)$ can be approximated by

$$H^{\times}(f_{T_\varphi}, f_R) \simeq -\frac{1}{|T_\varphi|} \sum_{s \in T_\varphi} \log f_R(s), \tag{E.2}$$

and the differential entropy $H(f_{T_\varphi})$ can be approximated by the Ahmad-Lin estimator [Ahm&Lin76]

$$H_{\mathrm{AL}}(T_\varphi) = -\frac{1}{|T_\varphi|} \sum_{s \in T_\varphi} \log f_{T_\varphi}(s) \ . \tag{E.3}$$

In (E.3), the probability density function (PDF) is by definition equal to

$$f_{T_\varphi}(s) = \frac{1}{|T_\varphi|} \sum_{t \in T_\varphi} K_\sigma(s - t) \ . \tag{E.4}$$

The same estimation (replacing T_φ with R) will be used in (E.2).

Therefore, we have

$$\mathcal{E}(\varphi) := \sum_{s \in T_\varphi} \log f_R(s) - \log f_{T_\varphi}(s) \tag{E.5}$$

$$\simeq \ - \, |T_\varphi| \, \mathfrak{D}_{\mathrm{KL}}(f_{T_\varphi}, f_R). \tag{E.6}$$

Note that $|T_\varphi|$ is constant for all candidate regions in a given frame. Consequently, taking the derivative of (E.5) 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. The sample set T_φ is equal to

$$T_\varphi = \{(I_{\text{tgt}}(x+u, y+v), x/\alpha, y/\alpha),\ (x, y) \in \Omega\}\ . \tag{E.7}$$

The derivative of (E.5) with respect to $\varphi = (\alpha, u, v)$ is equal to

$$\nabla\mathcal{E}(\varphi) = \sum_{s\in T_\varphi}\left(\frac{1}{f_R(s)}\frac{1}{|R|}\sum_{t\in R}\frac{\partial}{\partial\varphi}K_\sigma(s-t) - \frac{1}{f_{T_\varphi}(s)}\frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\partial}{\partial\varphi}K_\sigma(s-t)\right) \tag{E.8}$$

$$= \sum_{s\in T_\varphi}\left(\frac{1}{f_R(s)}\frac{1}{|R|}\sum_{t\in R}\mathcal{D}_s(T_\varphi)\,\nabla K_\sigma(s-t) - \frac{1}{f_{T_\varphi}(s)}\frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\partial}{\partial\varphi}K_\sigma(s-t)\right) \tag{E.9}$$

where

$$\mathcal{D}_s(T_\varphi) = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{\alpha^2}[s_x \quad s_y] \\ \nabla I^Y_{\text{tgt}}\begin{pmatrix}s_x+u\\ s_y+v\end{pmatrix} & \nabla I^U_{\text{tgt}}\begin{pmatrix}s_x+u\\ s_y+v\end{pmatrix} & \nabla I^V_{\text{tgt}}\begin{pmatrix}s_x+u\\ s_y+v\end{pmatrix} & \big[\,0\,\big]_{[2\times 2]} \end{bmatrix}\ . \tag{E.10}$$

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

$$\nabla\mathcal{E}(\varphi) = \sum_{s\in T_\varphi}\mathcal{D}_s(T_\varphi)\left(\frac{\nabla f_R(s)}{f_R(s)} - \frac{\nabla f_{T_\varphi}(s)}{f_{T_\varphi}(s)} + \frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\nabla K_\sigma(t-s)}{f_{T_\varphi}(t)}\right)\ . \tag{E.11}$$

E.2 Term interpretation

Let us focus on the following term of (E.11)

$$\mathcal{A}(s) := \frac{1}{|T_\varphi|}\sum_{t\in T_\varphi}\frac{\nabla K_\sigma(t-s)}{f_{T_\varphi}(t)}\ . \tag{E.12}$$

When the number of samples $|T_\varphi|$ tends toward infinity, $\mathcal{A}$ tends toward

$$\mathcal{A}_\infty(s) = \int_{\mathbb{R}^d} f_{T_\varphi}(t)\,\frac{\nabla K_\sigma(t-s)}{f_{T_\varphi}(t)}\,\mathrm{d}t\ . \tag{E.13}$$

The kernel K_σ is radially symmetric. Therefore, for all x and y such that $x = -y$, we have

$$\nabla K_\sigma(x) = -\nabla K_\sigma(y). \tag{E.14}$$

Therefore, (E.13) convergences (at least weakly) toward zero. It will then be neglected. If, nonetheless, one wants to evaluate it, Section F.2 provides a k nearest neighbor (kNN) approximation.

Appendix F

Derivative of the Kullback-Leibler divergence: kNN implementation

F.1 kNN-based expression

The first two terms enclosed in parentheses in (E.11) can be approximated using the mean shift (2.4). The expression of the mean (2.5) can be replaced with its kNN equivalent [Fuk&Hos75]

$$\bar{s}_{\rho_k(s)} = \frac{1}{k} \sum_{t \in W_{\rho_k(s)}} t \ . \tag{F.1}$$

In the third term enclosed in parentheses in (E.11), the PDF f_{T_φ} can also be replaced with its kNN expression (4.3). Therefore, using the mean shift approximation, the derivative of the Kullback-Leibler divergence can be written as a kNN-based expression

$$k \ \nabla \mathcal{E}(\varphi) = \sum_{s \in T_\varphi} \mathcal{D}_s(T_\varphi) \left(\frac{d+2}{\rho_k^2(R,s)} \sum_{t \in W_{\rho_k(R,s)}} (t-s) - \frac{d+2}{\rho_k^2(T_\varphi,s)} \sum_{t \in W_{\rho_k(T_\varphi,s)}} (t-s) \right.$$
$$\left. + v_d \sum_{t \in T_\varphi} \rho_k^d(T_\varphi,t) \ \nabla K_{\rho_k(T_\varphi,t)}(t-s) \right) \tag{F.2}$$

where $K_{\rho_k(T_\varphi,t)}(\cdot - s)$ is a window of radius $\rho_k(T_\varphi,t)$ centered at s.

F.2 Term approximation

Let us now focus on the following term of (F.2) (which corresponds to the kNN version of (E.12))

$$\mathcal{A}_{\mathrm{kNN}}(s) := \sum_{t \in T_\varphi} \rho_k^d(T_\varphi, t)\ \nabla K_{\rho_k(T_\varphi,t)}(t-s)\ . \tag{F.3}$$

In light of Appendix E.2, this term could be neglected if $|T_\varphi|$ is large enough. Nevertheless, let us propose an approximation of it.

The window $K_{\rho_k(T_\varphi,t)}(\cdot - s)$ at t is equal to $[\rho_k^d(T_\varphi, t)\ v_d]^{-1}$ if $|t-s| \leq \rho_k(T_\varphi, t)$ and zero otherwise. A finite difference approximation can be used to write

$$\nabla K_{\rho_k(T_\varphi,t)}(t-s) = \begin{cases} \frac{1}{\rho_k^d(T_\varphi,t)\ v_d}\ \frac{s-t}{|s-t|} & \text{if } |s-t| = \rho_k(T_\varphi,t) \\ 0 & \text{otherwise.} \end{cases} \tag{F.4}$$

Therefore, the term (F.3) can be approximated by

$$\mathcal{A}_{\mathrm{kNN}}(s) \simeq \frac{1}{v_d} \sum_{\substack{t \in T_\varphi \\ |t-s| = \rho_k(T_\varphi,t)}} \frac{s-t}{\rho_k(T_\varphi,t)}\ . \tag{F.5}$$

This approximation leads to the final expression (9.8) of the kNN-based derivative of (5.11). Note that, in practice, the summation condition $|t-s| = \rho_k(T_\varphi, t)$ should be understood as $|t-s| \in \rho_k(T_\varphi, t) \pm \epsilon$ for a small ϵ.

LIST OF FIGURES & TABLES

List of Figures

List of Tables

BIBLIOGRAPHY & INDEX

Bibliography

[Ahm&Lin76] Ahmad, I. A. and Lin, P. E. (1976). A nonparametric estimation of the entropy for absolutely continuous distributions. *IEEE Trans. Inf. Theory*, 22:372–375.

[Ang+08a] Angelino, C. V., Debreuve, É., and Barlaud, M. (2008a). Image restoration using a kNN-variant of the mean shift. In *International Conference on Image Processing*, San Diego (CA), USA.

[Ang+08b] Angelino, C. V., Debreuve, É., and Barlaud, M. (2008b). A nonparametric minimum entropy image deblurring algorithm. In *International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas (NV), USA.

[Ano07] Anonymous Reviewer (2007). Thorough review of [Deb+07].

[Ant+08] Anthoine, S., Debreuve, É., Piro, P., and Barlaud, M. (2008). Using neighborhood distributions of wavelet coefficients for on-the-fly, multiscale-based image retrieval. In *Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt, Austria.

[Aru+02] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, 50:174–188.

[Aub+03] Aubert, G., Barlaud, M., Faugeras, O., and Jehan-Besson, S. (2003). Image segmentation using active contours: Calculus of variations or shape gradients? *SIAM J. Appl. Math.*, 63:2128–2154.

[Awa&Whi06] Awate, S. P. and Whitaker, R. T. (2006). Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:364–376.

[Bab+07] Babu, R. V., Pérez, P., and Bouthemy, P. (2007). Robust tracking with motion estimation and local kernel-based color modeling. *Image Vis. Comput.*, 25:1205–1216.

[Bar&Com04] Barash, D. and Comaniciu, D. (2004). A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Video Computing*, 22:73–81.

[Bar+09] Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch: a randomized correspondence algorithm for structural image editing. In *Special Interest Group on GRAPHics and Interactive Techniques*, New Orleans (LA), USA.

[Bel&Niy01] Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Neural Information Processing Systems*, Vancouver (BC), Canada.

[Ber+07] Bertalmío, M., Caselles, V., and Pardo, A. (2007). Movie denoising by average of warped lines. *IEEE Trans. Image Process.*, 16:2333–2347.

[Ber+00] Bertalmío, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Special Interest Group on GRAPHics and Interactive Techniques*, New Orleans (LA), USA.

[Ber+03] Bertalmío, M., Vese, L. A., Sapiro, G., and Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, 12:882–889.

[Bla&Ana96] Black, M. J. and Anandan, P. (1996). The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Und.*, 63:75–104.

[Bol+07] Boltz, S., Debreuve, É., and Barlaud, M. (2007). High-dimensional statistical distance for region-of-interest tracking: application to combining a soft geometric constraint with radiometry. In *International Conference on Computer Vision and Pattern Recognition*, Minneapolis (MN), USA.

[Bol+09] Boltz, S., Debreuve, É., and Barlaud, M. (2009). High-dimensional statistical measure for region-of-interest tracking. *IEEE Trans. Image Process.*, 18:1266–1283.

[Bol+08] Boltz, S., Herbulot, A., Debreuve, É., Barlaud, M., and Aubert, G. (2008). Motion and appearance nonparametric joint entropy for video segmentation. *Int. J. Comput. Vision*, 80:242–259.

[Bol+06] Boltz, S., Wolsztynski, É., Debreuve, É., Thierry, É., Barlaud, M., and Pronzato, L. (2006). A minimum-entropy procedure for robust motion estimation. In *International Conference on Image Processing*, Atlanta (GA), USA.

[Bou+07] Boulanger, J., Kervrann, C., and Bouthemy, P. (2007). Space-time adaptation for patch-based image sequence restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:1096–1102.

[Bri+00] Brigger, P., Hoeg, J., and Unser, M. (2000). B-spline snakes: a flexible tool for parametric contour detection. *IEEE Trans. Image Process.*, 9:1484–1496.

[Bro+04] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, Prague, Czech Republic.

[Bro&Cre07] Brox, T. and Cremers, D. (2007). Iterated nonlocal means for texture restoration. In *International Conference on Scale Space Methods and Variational Methods in Computer Vision*, Ischia, Italy.

[Bro+03] Brox, T., Rousson, M., Deriche, R., and Weickert, J. (2003). Unsupervised segmentation incorporating colour, texture, and motion. In *Computer Analysis of Images and Patterns*, Groningen, The Netherlands.

[Bru+05] Bruhn, A., Weickert, J., and Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vision*, 61:211–231.

[Bua+05a] Buades, A., Coll, B., and Morel, J.-M. (2005a). A non-local algorithm for image denoising. In *International Conference on Computer Vision and Pattern Recognition*, San Diego (CA), USA.

[Bua+05b] Buades, A., Coll, B., and Morel, J.-M. (2005b). A review of image denoising algorithms, with a new one. *SIAM Multiscale Model. Simul.*, 4:490–530.

[Bua+06] Buades, A., Coll, B., and Morel, J. M. (2006). Neighborhood filters and PDE's. *Nümer. Math.*, 105:1–34.

[Bug&Per07] Bugeau, A. and Pérez, P. (2007). Detection and segmentation of moving objects in highly dynamic scenes. In *International Conference on Computer Vision and Pattern Recognition*, Minneapolis (MN), USA.

[Bur&Ade83] Burt, P. J. and Adelson, E. H. (1983). The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, COM-31:532–540.

[Car+08] Carlsson, G., Ishkhanov, T., de Silva, V., and Zomorodian, A. (2008). On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, 76:1–12.

[Cas+93] Caselles, V., Catté, F., Coll, T., and Dibos, F. (1993). A geometric model for active contours. *Nümer. Math.*, 66:1–31.

[Cas+97] Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *Int. J. Comput. Vision*, 22:61–79.

[Cha&Ves01] Chan, T. and Vese, L. A. (2001). Active contours without edges. *IEEE Trans. Image Process.*, 10:266–277.

[Cha+97] Charbonnier, P., Blanc-Féraud, L., Aubert, G., and Barlaud, M. (1997). Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.*, 6:298–311.

[Cha+05] Charpiat, G., Keriven, R., Pons, J.-C., and Faugeras, O. (2005). Designing spatially coherent minimizing flows for variational problems based on active contours. In *International Conference on Computer Vision*, Beijing, China.

[Che+99] Chesnaud, C., Réfrégier, P., and Boulet, V. (1999). Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:1145–1156.

[Col+05] Collins, R., Zhou, X., and Teh, S. K. (2005). An open source tracking testbed and evaluation web site. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, Breckenridge (CO), USA.

[Com03] Comaniciu, D. (2003). An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:281–288.

[Com&Mee02] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:603–619.

[Com+00] Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. In *International Conference on Computer Vision and Pattern Recognition*, Hilton Head Island (SC), USA.

[Cos&Her04] Costa, J. and Hero, A. O. (2004). Manifold learning using Euclidean K-nearest neighbor graphs. In *International Conference on Acoustics, Speech, and Signal Processing*, Montreal (QC), Canada.

[Cov&Tho91] Cover, T. and Thomas, J. (1991). *Elements of information theory*. Wiley-Interscience.

[Cra+08] Craenea, M. D., Macq, B., Marquesc, F., Salembier, P., and Warfield, S. (2008). Unbiased group-wise alignment by iterative central tendency estimation. *Math. Model. Nat. Phenom.*, 3:2–32.

[Cre+07] Cremers, D., Rousson, M., and Deriche, R. (2007). A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *Int. J. Comput. Vision*, 72:195–215.

[Cre&Sch02] Cremers, D. and Schnörr, C. (2002). Motion competition: variational integration of motion segmentation and shape regularization. In *DAGM-Symposium, Pattern Recognition*, Zürich, Switzerland.

[Cre&Soa03] Cremers, D. and Soatto, S. (2003). Variational space-time motion segmentation. In *International Conference on Computer Vision*, Nice, France.

[Cri+04] Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.*, 13:1200–1212.

[Dab+07] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. O. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16:2080–2095.

[Deb+01] Debreuve, É., Barlaud, M., Aubert, G., Laurette, I., and Darcourt, J. (2001). Space-time segmentation using level set active contours applied to myocardial gated SPECT. *IEEE Trans. Med. Imag.*, 20:643–659.

[Deb+06] Debreuve, É., Barlaud, M., Marmorat, J.-P., and Aubert, G. (2006). Active contour segmentation with a parametric shape prior: link with the shape gradient. In *International Conference on Image Processing*, Atlanta (GA), USA.

[Deb+07] Debreuve, É., Gastaud, M., Barlaud, M., and Aubert, G. (2007). Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. *J. Math. Imaging Vis.*, 28:47–66.

[Del&Zol01] Delfour, M. C. and Zolésio, J.-P. (2001). *Shapes and geometries: Analysis, differential calculus and optimization*. Society for Industrial and Applied Mathematics.

[Del&Mon01] Delingette, H. and Montagnat, J. (2001). Shape and topology constraints on parametric active contours. *Comput. Vis. Image Und.*, 83:140–171.

[Do&Vet02] Do, M. N. and Vetterli, M. (2002). Wavelet based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Trans. Image Process.*, 11:146–158.

[Doc+05] Doğan, G., Morin, P., Nochetto, R. H., and Verani, M. (2005). Finite element methods for shape optimization and applications. *Preprint*.

[Efr&Fre01] Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Special Interest Group on GRAPHics and Interactive Techniques*, Los Angeles (CA), USA.

[Efr&Leu99] Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, Corfu, Greece.

[Elg+03] Elgammal, A., Duraiswami, R., and Davis, L. S. (2003). Probabilistic tracking in joint feature-spatial spaces. In *International Conference on Computer Vision and Pattern Recognition*, Madison (WI), USA.

[Exc06] Excerpt from the movie “Poltergay” directed by Eric Lavaine (2006). Produced by Fabio Conversi, François Cornuau, and Vincent Roget.

[Fix&Hod51] Fix, E. and Hodges, J. L. (1951). Discriminatory analysis, nonparametric discrimination: consistency properties. Technical report, USAF School of Aviation Medicine, Randolph Field.

[Fre&Zha04] Freedman, D. and Zhang, T. (2004). Active contours for tracking distributions. *IEEE Trans. Image Process.*, 13:518–526.

[Fuk90] Fukunaga, K. (1990). *Introduction to statistical pattern recognition (2nd Ed.)*. Academic Press Professional, Inc.

[Fuk&Hos75] Fukunaga, K. and Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory*, 21:32–40.

[Gas+04] Gastaud, M., Barlaud, M., and Aubert, G. (2004). Combining shape prior and statistical features for active contour segmentation. *IEEE Trans. Circuits Syst. Video Technol.*, 14:726–734.

[Geo+03] Georgescu, B., Shimshoni, I., and Meer, P. (2003). Mean shift based clustering in high dimensions: A texture classification example. In *International Conference on Computer Vision*, Nice, France.

[Ger&Ref96] Germain, O. and Réfrégier, P. (1996). Optimal snake-based segmentation of a random luminance target on a spatially disjoint background. *Opt. Lett.*, 21:1845–1847.

[Gil&Osh07] Gilboa, G. and Osher, S. (2007). Nonlocal linear image regularization and supervised segmentation. *SIAM Multiscale Model. Simul.*, 6:595–630.

[Gor+05] Goria, M. N., Leonenko, N. N., Mergel, V. V., and Inverardi, P. L. N. (2005). A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *J. Nonparametr. Stat.*, 17:277–297.

[Got97] Gottingen TSTOOL (1997). DPI Göttingen, TSTOOL toolbox for nearest neighbor statistics. Code available at: http://www.dpi.physik.uni-goettingen.de/tstool/.

[Hau&Cho93] Haug, E. and Choi, K. K. (1993). *Methods of engineering mathematics*. Prentice Hall, Englewood Cliiffs.

[Her+06] Herbulot, A., Jehan-Besson, S., Duffner, S., Barlaud, M., and Aubert, G. (2006). Segmentation of vectorial image features using shape gradients and information measures. *J. Math. Imaging Vis.*, 25:365–386.

[Hin&Rin03] Hintermüller, M. and Ring, W. (2003). A second order shape optimization approach for image segmentation. *SIAM J. Appl. Math.*, 64:442–467.

[Hor&Sch81] Horn, B. K. P. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[Hua&Mum99] Huang, J. and Mumford, D. (1999). Statistics of natural images and models. In *International Conference on Computer Vision and Pattern Recognition*, Ft. Collins (CO), USA.

[Ihl03] Ihler, A. (2003). Kernel density estimation toolbox for Matlab. Code available at: http://ttic.uchicago.edu/~ihler/code/kde.php.

[Isa&Bla98] Isard, M. and Blake, A. (1998). CONDENSATION - Conditional density propagation for visual tracking. *Int. J. Comput. Vision*, 29:5–28.

[Jac+01] Jacob, M., Blu, T., and Unser, M. (2001). A unifying approach and interface for spline-based snakes. In *SPIE International Symposium on Medical Imaging: Image Processing*, San Diego (CA), USA.

[Jac+04] Jacob, M., Blu, T., and Unser, M. (2004). Efficient energies and algorithms for parametric snakes. *IEEE Trans. Image Process.*, 13:1231–1244.

[Jeh+03] Jehan-Besson, S., Barlaud, M., and Aubert, G. (2003). DREAM^{2}S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation. *Int. J. Comput. Vision*, 53:45–70.

[Kad&Bra01] Kadir, T. and Brady, M. (2001). Saliency, scale and image description. *Int. J. Comput. Vision*, 45:83–105.

[Kad+04] Kadir, T., Zisserman, A., and Brady, M. (2004). An affine invariant salient region detector. In *European Conference on Computer Vision*, Prague, Czech Republic.

[Kas+88] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *Int. J. Comput. Vision*, 1:321–332.

[Ker&Bou06] Kervrann, C. and Boulanger, J. (2006). Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. Image Process.*, 15:2866–2878.

[Kim+05] Kim, J., Fisher, J. W. F., Yezzi, A., Çetin, M., and Willsky, A. (2005). Nonparametric methods for image segmentation using information theory and curve evolution. *IEEE Trans. Image Process.*, 14:1486–1502.

[Kin+05] Kindermann, S., Osher, S., and Jones, P. (2005). Deblurring and denoising of images by nonlocal functionals. *SIAM Multiscale Model. Simul.*, 4:1091–1115.

[Koz&Leo87] Kozachenko, L. and Leonenko, N. N. (1987). On statistical estimation of entropy of random vector. *Problems Infor. Transmiss.*, 23:95–101.

[Laz+06] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *International Conference on Computer Vision and Pattern Recognition*, New York (NY), USA.

[Lee+03] Lee, A. B., Pedersen, K. S., and Mumford, D. (2003). The nonlinear statistics of high-contrast patches in natural images. *Int. J. Comput. Vision*, 54:83–103.

[Leo+08] Leonenko, N. N., Pronzato, L., and Savani, V. (2008). A class of Rényi information estimators for multidimensional densities. *Annals of Statistics*, 36:2153–2182.

[Li+04] Li, H., Elmoataz, A., Fadili, J., and Ruan, S. (2004). Dual front evolution model and its application in medical imaging. In *Medical Image Computing and Computer-Assisted Intervention*, Rennes/Saint-Malo, France.

[Li&Yez05] Li, H. and Yezzi, A. (2005). Local or global minima: flexible dual front active contours. In *Computer Vision for Biomedical Image Applications*, Nice, France.

[Li&Wan03] Li, J. and Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:1075–1088.

[Lin91] Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory*, 37:145–151.

[Lob&Vie95] Lobregt, S. and Viergever, M. (1995). A discrete dynamic contour model. *IEEE Trans. Med. Imag.*, 14:12–23.

[Lof&Que65] Loftsgaarden, D. O. and Quesenberry, C. P. (1965). A nonparametric estimate of a multivariate density function. *Ann. Math. Statist.*, 36:1049–1051.

[Low04] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110.

[Luc&Kan81] Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, Vancouver (BC), Canada.

[Mar05] Marmorat, J.-P. (2005). Private communication on shape gradient in the context of a parametric active contour.

[Mez+06] Mezaris, V., Kompatsiaris, I., and Strintzis, M. G. (2006). Object-based mpeg-2 video indexing and retrieval in a collaborative environment. *Multimed. Tools Appl.*, 30:255–272.

[Mik&Sch05] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:1615–1630.

[MIT02] MIT Media Lab: VisMod Group (2002). Texture images "VisTex". http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html.

[Mol&Dub91] Moloney, C. and Dubois, E. (1991). Estimation of motion fields from image sequences with illumination variation. In *International Conference on Acoustics, Speech, and Signal Processing*, Toronto (ON), Canada.

[Mor&Yu09] Morel, J.-M. and Yu, G. (2009). ASIFT: a new framework for fully affine invariant image comparison *to appear*. *SIAM J. Imag. Sciences*.

[Mum&Sha89] Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685.

[Nad+05] Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. (2005). Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Neural Information Processing Systems*, Vancouver (BC), Canada.

[Neg&Yu93] Negahdaripour, S. and Yu, C. (1993). A generalized brightness change model for computing optical flow. In *International Conference on Computer Vision*, Berlin, Germany.

[Ove&Sol05] Overgaard, N. C. and Solem, J. E. (2005). An analysis of variational alignment of curves in images. In *International Conference on Scale Space and PDE methods in Computer Vision*, Hofgeismar, Germany.

[Par&Der99] Paragios, N. and Deriche, R. (1999). Geodesic active regions for motion estimation and tracking. In *International Conference on Computer Vision*, Corfu, Greece.

[Par&Der02] Paragios, N. and Deriche, R. (2002). Geodesic active regions and level set methods for supervised texture segmentation. *Int. J. Comput. Vision*, 46:223–247.

[Per+02] Pérez, P., Hue, C., Vermaak, J., and Gangnet, M. (2002). Color-based probabilistic tracking. In *European Conference on Computer Vision*, Copenhagen, Denmark.

[Per&Mal90] Perona, P. and Malik, J. (1990). Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639.

[Pie+05] Pierpaoli, E., Anthoine, S., Huffenberger, K., and Daubechies, I. (2005). Reconstructing Sunyaev-Zeldovich clusters in future CMB experiments. *Mon. Not. R. Astron. Soc.*, 359:261–271.

[Pir+08a] Piro, P., Anthoine, S., Debreuve, É., and Barlaud, M. (2008a). Image retrieval via Kullback-Leibler divergence of patches of multiscale coefficients in the kNN framework. In *International Workshop on Content-Based Multimedia Indexing*, London, UK.

[Pir+08b] Piro, P., Anthoine, S., Debreuve, É., and Barlaud, M. (2008b). Sparse Multiscale Pacthes for image processing. In *Emerging Trends in Visual Computing (ETVC)*, Palaiseau, France.

[Pir+08c] Piro, P., Anthoine, S., Debreuve, É., and Barlaud, M. (2008c). Sparse Multiscale Patches (SMP) for image categorization. In *International Conference on Multimedia Modeling*, Sophia Antipolis, France.

[Plu+03] Pluim, J. P. W., Maintz, J. B. A., and Viergever, M. A. (2003). Mutual information based registration of medical images: a survey. *IEEE Trans. Med. Imag.*, 22:986–1004.

[Pog08] Poggi, G. (2008). Informal discussion.

[Por+03] Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. P. (2003). Image denoising using a scale mixture of Gaussians in the wavelet domain. *IEEE Trans. Image Process.*, 12:1338–1351.

[Pre+05] Precioso, F., Barlaud, M., Blu, T., and Unser, M. (2005). Robust real-time segmentation of images and videos using a smooth-spline snake-based algorithm. *IEEE Trans. Image Process.*, 14:910–924.

[Puz+99] Puzicha, J., Rubner, Y., Tomasi, C., and Buhmann, J. M. (1999). Empirical evaluation of dissimilarity measures for color and texture. In *International Conference on Computer Vision*, Corfu, Greece.

[Rob&Mil03] Robinson, D. and Milanfar, P. (2003). Fast local and global projection-based methods for affine motion estimation. *J. Math. Imaging Vis.*, 18:35–54.

[Rom+01] Romberg, J. K., Choi, H., and Baraniuk, R. G. (2001). Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Trans. Image Process.*, 10:1056–1068.

[Roy+06] Roy, T., Debreuve, É., Barlaud, M., and Aubert, G. (2006). Segmentation of a vector field: dominant parameter and shape optimization. *J. Math. Imaging Vis.*, 24:259–276.

[Rub+00] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40:99–121.

[Sai02] Sain, S. R. (2002). Multivariate locally adaptive density estimation. *Comput. Stat. Data Anal.*, 39:165–186.

[Sch92] Schnörr, C. (1992). Computation of discontinuous optical flow by domain decomposition and shape optimization. *Int. J. Comput. Vision*, 8:153–165.

[Sco92] Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley.

[Shi&Mal00] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:888–905.

[Sil86] Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.

[Siv&Zis03] Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, Nice, France.

[Sok&Zol92] Sokolowski, J. and Zolésio, J.-P. (1992). *Introduction to shape optimization: Shape sensitivity analysis*. Springer-Verlag, Berlin.

[Sol&Ove05] Solem, J. E. and Overgaard, N. C. (2005). A geometric formulation of gradient descent for variational problems with moving surfaces. In *International Conference on Scale Space and PDE methods in Computer Vision*, Hofgeismar, Germany.

[Sri+03] Srivastava, A., Lee, A. B., Simoncelli, E. P., and Zhu, S.-C. (2003). On advances in statistical modeling of natural images. *J. Math. Imaging Vis.*, 18:17–33.

[Szu&Pic98] Szummer, M. and Picard, R. W. (1998). Indoor-outdoor image classification. In *International Workshop on Content-Based Access of Image and Video Databases*, Washington (DC), USA".

[Tat&Lac02] Taton, B. and Lachaud, J.-O. (2002). Deformable model with non-Euclidean metrics. In *European Conference on Computer Vision*, Copenhagen, Denmark.

[Ten+00] Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323.

[Ter&Sco92] Terrell, G. R. and Scott, D. W. (1992). Variable kernel density estimation. *Ann. Statist.*, 20:1236–1265.

[Tom&Man98] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *International Conference on Computer Vision*, Bombay, India.

[Tuy&Mik07] Tuytelaars, T. and Mikolajczyk, K. (2007). Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3:177–280.

[Una+05] Unal, G., Yezzi, A., and Krim, H. (2005). Information-theoretic active polygons for unsupervised texture segmentation. *Int. J. Comput. Vision*, 62:199–220.

[Uns+93] Unser, M., Aldroubi, A., and Eden, M. (1993). B-spline signal processing: part I – Theory. *IEEE Trans. Signal Process.*, 41:821–833.

[Vai+01] Vailaya, A., Figueiredo, M., Jain, A., and Zhang, H.-J. (2001). Image classification for content-based indexing. *IEEE Trans. Image Process.*, 10:117–130.

[Vio&Wel97] Viola, P. and Wells, W. M. (1997). Alignment by maximization of mutual information. *Int. J. Comput. Vision*, 24:137–154.

[Wei&Sch01] Weickert, J. and Schnörr, C. (2001). Variational optic flow computation with a spatio-temporal smoothness constraint. *J. Math. Imaging Vis.*, 14:245–255.

[Yan+03] Yang, C., Duraiswami, R., Gumerov, N. A., and Davis, L. (2003). Improved fast Gauss transform and efficient kernel density estimation. In *International Conference on Computer Vision*, Nice, France.

[Yen+05] Yen, L., Vanvyve, D., Wouters, F., Fouss, F., Verleysen, M., and Saerens, M. (2005). Clustering using a random walk based distance measure. In *European Symposium on Artificial Neural Network*, Bruges, Belgium.

[Yez+99] Yezzi, A., Tsai, A., and Willsky, A. (1999). A statistical approach to snakes for bimodal and trimodal imagery. In *International Conference on Computer Vision*, Corfu, Greece.

[Zha&Izq06] Zhang, Q. and Izquierdo, E. (2006). Optimizing metrics combining low-level visual descriptors for image annotation and retrieval. In *International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France.

[Zhu&Ma00] Zhu, S. and Ma, K.-K. (2000). A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.*, 9:287–290.

Bibliographical index

[Efr&Leu99] 69, 118
[Efr&Fre01] 69
[Elg+03] 6, 90, 91, 94

F

[Fix&Hod51] 29, 91
[Fre&Zha04] 6, 91
[Fuk90] 17, 18, 28, 44
[Fuk&Hos75] . . 17, 29, 44, 75, 151

G

[Gas+04] 133
[Geo+03] 29
[Ger&Ref96] 52, 134
[Gil&Osh07] 80
[Gor+05] 32, 33, 91
[Got97] 98

H

[Hau&Cho93] 50
[Her+06] 91
[Hin&Rin03] 50–52, 134
[Hor&Sch81] 120
[Hua&Mum99] 69

I

[Ihl03] 90, 98
[Isa&Bla98] 66

J

[Jac+01] 57
[Jac+04] 60
[Jeh+03] . . . 50, 51, 133, 134, 139

K

[Kad+04] 4, 6
[Kad&Bra01] 4, 6
[Kas+88] 134
[Ker&Bou06] 69, 113
[Kim+05] 6, 91
[Kin+05] 80, 88
[Koz&Leo87] 32, 33, 91

L

[Laz+06] 121
[Lee+03] 6, 44, 69, 91, 94
[Leo+08] 32, 33, 91
[Li&Yez05] 133
[Li+04] 133
[Lin91] 92
[Li&Wan03] 121
[Lob&Vie95] 55
[Lof&Que65] 28, 29, 91
[Low04] 4, 6, 90, 91
[Luc&Kan81] 120

M

[Mar05] 60
[Mez+06] 121
[Mik&Sch05] 4, 6
[Mol&Dub91] 120
[Mor&Yu09] 4, 6, 91
[Mum&Sha89] 133

N

[Nad+05] 130
[Neg&Yu93] 120

O

[Ove&Sol05] 134, 135

P

[Par&Der99] 133
[Par&Der02] 90
[Per+02] 90, 93, 96
[Per&Mal90] 70, 81
[Pie+05] 123
[Pir+08a] 123, 124
[Pir+08b] 123
[Pir+08c] 123, 124
[Plu+03] 6
[Pog08] 118
[Exc06] 109, 110
[Por+03] 121, 123
[Pre+05] 57
[Puz+99] 123

R

[Rob&Mil03] 66
[Rom+01] 121
[Roy+06] 65, 141
[Rub+00] 44

S

[Sai02] 28, 91
[Sch92] 50, 134
[Sco92] 27, 90
[Shi&Mal00] 130
[Sil86] 27
[Siv&Zis03] 123
[Sok&Zol92] 50, 51, 134, 137, 138
[Sol&Ove05] 51, 134, 135
[Sri+03] 44, 69
[Szu&Pic98] 121

T

[Tat&Lac02] 54
[Ten+00] 130
[Ter&Sco92] .. 7, 17, 27, 28, 31, 33, 91, 128
[Tom&Man98] 80
[Tuy&Mik07] 4, 6

U

[Una+05] 6, 60
[Uns+93] 57

V

[Vai+01] 121
[Bab+07] 94
[Vio&Wel97] 6, 31
[MIT02] 82

W

[Wei&Sch01] 94, 120

Y

[Yan+03] 90
[Yen+05] 130
[Yez+99] 134

Z

[Zha&Izq06] 121
[Zhu&Ma00] 66, 96, 99, 109