Politecnico di Milano
Dipartimento di Matematica "F. Brioschi"

Dottorato di Ricerca in Ingegneria Matematica
XVII ciclo

Alessandro Foi

Anisotropic nonparametric image processing: theory, algorithms and applications

Dottorando: Alessandro Foi
Matricola: D01290

Relatore: Prof. Karen Egiazarian

Coordinatore: Prof. Marco Fuhrman

Contents

Foreword

The research work on which this thesis is based has been performed during my stay at the Institute of Signal Processing - Tampere University of Technology (TUT), Finland, in the last two years of doctoral studies. This work has been accomplished under the supervision of Professors Karen Egiazarian and Vladimir Katkovnik. I am greatly indebted towards both of them, for their highly professional guidance and for being so patient with me.

I wish to express my gratitude to the Collegio dei Docenti del Dottorato di Ricerca in Ingegneria Matematica, and, in particular, to its Coordinator, Professor Marco Fuhrman, for giving me the chance to pursue my research studies in Finland.

My special thanks go to Professors Jaakko Astola, director of Tampere International Center for Signal Processing (TICSP) - TUT, and Moncef Gabbouj, director of the Institute of Signal Processing - TUT, for the highly rewarding professional and educational experience which they have offered me.

Finally, I would like to thank the institutions that, during the three years of doctoral studies, have provided me with the necessary financial support. They include: Dipartimento di Matematica - Politecnico di Milano, Institute of Signal Processing - TUT, Tampere International Center for Signal Processing (TICSP) - TUT, Dipartimento di Matematica - Università degli Studi di Milano, Instituto Nazionale di Alta Matematica (INdAM), and Nokia Research Center (NRC) - Tampere.

Tampere, April 2005 Alessandro Foi

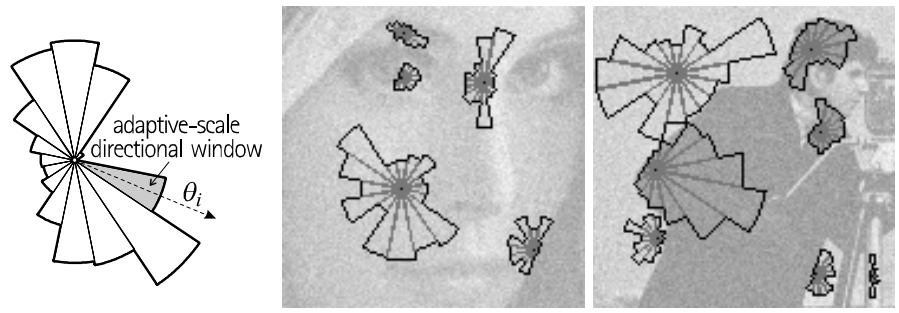


Figure 1: Anisotropic local approximations achieved by combining a number of adaptive-scale directional windows. The examples show some of these windows selected by the directional *LPA-ICI* for the noisy *Lena* and *Cameraman* images.

Introduction

When estimating an image from its noisy observations, a trade-off between noise suppression (variance) and smoothing (bias) has to be considered. Common images are nonstationary, often characterized by localized features. Therefore, images should be treated adaptively: for example, one would achieve a higher noise suppression where the original image is smooth, than in the vicinity of sharp transitions such as edges, where oversmoothing should be avoided.

So, the desired balance between variance and bias depends on the image's local features. How to control this balance is a key problem in adaptive signal processing. A novel and original strategy to achieve such adaptation is the main subject of this thesis.

The presented approach is based on the Intersection of Confidence Intervals (*ICI*) rule for pointwise-adaptive estimation. Originally, the method has been developed for 1D signals [28, 42]. The idea was generalized for 2D image processing, where adaptive-size quadrant windows have been used [43].

The main intention of these techniques is to obtain – in a data-driven way – a largest local neighborhood of the estimation point in which the underlying model fits the data.

Our main assumption [20, 46] is that this vicinity is a starshaped body which can be approximated by some sectorial decomposition with, say, K non-overlapping sectors. Such a sectorial approximation is shown in Figure 1. We use special directional kernels defined on a sectorial support. By endowing these kernels with a scale parameter, we are able to use the *ICI* rule for the pointwise selection of an adaptive scale, which defines the length of the sectorial support.

Anisotropy is enabled by allowing different adaptive scales for different directions. Thus, the *ICI* rule is exploited K times, once for each sector. In this way, we reduce a complex multidimensional shape adaptation problem, to a number of scalar optimizations.

The directional estimates corresponding to the adaptive-scale sectors are combined into the final *anisotropic* estimate. The resulting estimator is truly anisotropic, and its support can have quite an exotic shape. It is highly sensitive with respect to change-points, and allow to reveal fine elements of images from noisy observations, thus showing a remarkable advantage in the proposed strategy.

Several algorithms are developed, based on this estimator. Denoising is the main, and most natural application, but also deconvolution, and derivative estimation are problems where the anisotropic adaptation approach can play a significant role in order to achieve an improved restoration performance.

Outline

A necessary compromise

The relevance of the work behind this thesis is twofold: not only a quite general theoretical framework for a novel estimator has been developed, but also several algorithmic solutions to concrete problems of applicative interest have been developed, optimized, and compared against other state-of-the-art techniques. In many occasions, the proposed new techniques have proven to outperform the best methods appeared in the literature of our knowledge.

This thesis aims to cover these two rather different aspects, the theoretical and the algorithmical, in order to provide a complete view on the scope of the research that was carried out. In order not to overload the reader, it has been decided not to go too deep into the details of neither of these two aspects. Instead, our motivation was to emphasize the links between the theoretical modeling and the practical implementations. Consequently, in the theoretical exposition some topics are here purposely neglected (e.g. the use of a vector scale-parameter and some convergence analyses that have purely abstract importance), and in the description of the algorithms some aspects are not discussed (e.g. the procedures for the optimization of the algorithm parameters and some computational issues). Likewise, non essential remarks have been dropped or relegated to footnotes.

Nevertheless, we try to keep the exposition on a general-enough level: for example, the local polynomial approximation is presented as a very particular case of the general form of the moving least-squares method, described in the context of Hilbert-space approximations, and the asymptotic accuracy analyses are also done in a rather general way. Although, some finer aspects are eventually and inevitabily "rounded off", the resulting formulas are lean, and allow – in the author's opinion – a much easier understanding of the general applicability of the developed methods.

It has been also decided not to follow the standard "definition-theorem-proof-corollary" formalism, typical of most mathematical texts. Instead, a more informal and discoursive style is used.

Structure of the thesis

The thesis is structured in two parts. Firstly, we consider the theoretical background on which the developed technique is based: the *LPA,* the *ICI*, and the anisotropic fusing of the directional *LPA-ICI* estimates are presented. Denoising is considered as the underlying basic problem at which the approach is targeted. A recursive filtering strategy and an extension of the anisotropic

denoising method for the gradient estimation problem are also discussed. In the second part we present a number of different algorithms and applications in which the proposed techniques can be successfully exploited: besides denoising, also deblurring, inverse-halftoning and other related problems are considered.

Publications

Most of the material presented in this thesis appears in the following publications (sorted according to the thesis' presentation order):

1. [20]: Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "A novel anisotropic local polynomial estimator based on directional multiscale optimizations", *Proc. 6th IMA Int. Conf. Math. in Signal Processing*, Cirencester (UK), pp. 79-82, December 2004.
2. [46]: Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Directional varying scale approximations for anisotropic signal processing", *Proc. XII European Signal Proc. Conf., EUSIPCO 2004,* Vienna, pp. 101-104, September 2004.
3. [22]: Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Anisotropic local approximations for pointwise adaptive signal-dependent noise removal", (accepted) *XIII European Signal Proc. Conf., EUSIPCO 2005*, September 2005.
4. [47]: Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Anisotropic local likelihood approximations", *Proc. of Electronic Imaging 2005*, 5672-19, 2005.
5. [23]: Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", (accepted) *IEEE 2005 Int. Conf. Image Processing, ICIP 2005,* September 2005.
6. [21]: Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "Inverse halftoning based on the anisotropic LPA-ICI deconvolution", *Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Proc., SMMSP 2004, Vienna, Austria*, pp. 49-56, September 2004.
7. [14]: Ercole, C., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatio-temporal pointwise adaptive denoising of video: 3D non-parametric approach", *Proc. of the 1st International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM2005*, Scottsdale, AZ, January 2005.
8. [24]: Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool", (accepted) *SMMSP 2005,* Riga, June 2005.

Not everything from these publications is included here. Some topics where purposely discarded as they were only marginally relevant to the general idea of the presented approach. Instead, here we expand those aspects that are believed to be more useful for a clear understanding of the proposed method, and – with the same intention – add new material and considerations that have not been published.

Notation and conventions

We tried to use, as much as possible, well-known notational symbology. However, since – even for the most basic concepts – there exist in the literature various and equivocal notations, we gradually explain in the text the meaning of the used notation.

Nevertheless, as a useful reference, we declare here below, some of the most significant conventions that we follow.

To avoid ambiguity, we use often – but not always – the symbol $\triangleq$ to indicate the definition of a function, a variable, or a quantity.

The Fourier transform, continuous or discrete, is denoted by $\mathcal{F}$. We always use a "normalized" Fourier transform so that it realizes an isometry: $\|f\|_2 = \|\mathcal{F}(f)\|_2$. We usually indicate the Fourier transform of a function with the corresponding capital letter: $\mathcal{F}(p) = P$.

The symbol $\circledast$ denotes the convolution, $(g \circledast z)(x) = \int g(x-v)\, z(v)\, dv \ \ \forall x$. The central dot $\cdot$ indicates a "mute variable". For example, the above definition of the convolution can be written also as $g \circledast z = \int g(\cdot - v)\, z(v)\, dv$ (x being the mute variable).

The conjugate transpose of a matrix or vector is denoted by the superscript ^{T}.

Throughout the text, we use the hat decoration $\widehat{\ }$ to indicate estimated values (such as "$\hat{y}$ is the estimate of y"). The tilde decoration $\sim$ is used, usually in conjunction with a pedix x, to indicate a change of variable like the one used in convolutions: $\tilde{f}_x(v) = f(x-v)$. The same decoration is used not only for functions but also for sets: $\tilde{A}_x = \{v : (x-v) \in A\}$.

For a function $f : X \to \mathbb{R}$, we define its support, $\operatorname{supp} f \subseteq X$, as the subset on which the function is non-zero, $\operatorname{supp} f = \{x \in X : f(x) \neq 0\}$. The other way round, for a subset $A \subseteq X$, we define its characteristic function, $\chi_A : X \to \{0,1\}$, as the binary function that has A as its support: $\chi_A(x) = 1 \iff x \in A$, $\operatorname{supp} \chi_A = A$.

For the images used in the many figures and simulations, unless differently noted, we assume that the data-range is $[0,1]$, where zero and one correspond, respectively, to black and white. However, to allow an easier comparison with the other methods in the literature, we normalize our numerical criteria results to the range $[0,255]$. Therefore, it should not be surprising that – for example – the mean squared error is usually much larger than 1, even when the restored image is almost indistinguishable from the original.

Besides the usual ℓ^p norms $\|\cdot\|_p$ from the mathematical analysis, we use also the following well-known criteria functions to assess the objective quality of an

estimate $\hat{y}$ of y, obtained from a noisy observation z (where all the signals are defined on a domain X of size $|X|$):

$$\begin{aligned}
\text{(signal-to-noise ratio)} \quad SNR &= 20\log_{10}\left(\frac{\|y\|_2}{\|y-\hat{y}\|_2}\right), \\
\text{(improvement in } SNR\text{)} \quad ISNR &= 20\log_{10}\left(\frac{\|y-z\|_2}{\|y-\hat{y}\|_2}\right), \\
\text{(peak } SNR\text{)} \quad PSNR &= 20\log_{10}\left(\frac{255\cdot\sqrt{|X|}}{\|y-\hat{y}\|_2}\right), \\
\text{(blurred } SNR\text{)} \quad BSNR &= 20\log_{10}\left(\frac{\|y-\text{mean}\,(y)\|_2}{\|y-\hat{y}\|_2}\right).
\end{aligned}$$

Let us remind that the other "engineering metrics", namely the mean squared error (MSE), the root MSE ($RMSE$), the mean absolute error (MAE), and the maximum absolute difference (MAX), are – respectively – the square of the ℓ^2 norm (divided by $|X|$), the ℓ^2 norm (divided by $\sqrt{|X|}$), the ℓ^1 norm (divided by $|X|$), and the ℓ^∞ norm of $y-\hat{y}$.

Finally, we warn the reader that the notation used in this thesis does not exactly match the different notations that we used in aforementioned publications.

Part I

Theoretical background and basic methods

Chapter 1

Local approximations, the moving least-squares method, and the local polynomial approximation (*LPA*)

We introduce the local approximation approach in a very general form, within the framework of the best approximation in Hilbert spaces with respect to orthonormal systems and frames. Only the main ideas of these very general methods are discussed here. We refer the reader to classical textbooks such as [83] and [49] (for the functional analysis prerequisites), to [35] or [66] (where the theory of frames is discussed in connection to wavelets), and to [70] or the paper [16] (where a complete overview of the frame techniques in matrix form using weights is given). The polynomial and the discrete case – on which the actual developed algorithms are based – follow as particular instances of these methods. Although the local polynomial approximation can also be derived directly from the classical weighted least-squares method, we believe that by seeing it as a particular case of more general "geometrically flavoured" techniques may help in better understanding the common points and the differences with other methods that are based on Hilbert space approximations, such as wavelets, overcomplete expansions, etc.

To facilitate the reader, we often present formulas for the the non-orthogonal series expansions accompanied by the corresponding familiar expressions for the orthonormal expansions.

1.1 Analysis, reconstruction and approximation in Hilbert spaces

Let $\mathcal{H}$ be a Hilbert space. We denote, respectively, by $\langle\ ,\ \rangle_{\mathcal{H}}$ and $\|\cdot\|_{\mathcal{H}}$ the inner product and the norm in the space $\mathcal{H}$. The norm is defined from the inner

product as $\|\cdot\|_{\mathcal{H}}^2 = \langle \cdot, \cdot \rangle_{\mathcal{H}}$.

Let $\mathcal{M} \subseteq \mathcal{H}$ be a closed subspace. The dimension of $\mathcal{M}$, $\dim(\mathcal{M})$, is equal to the cardinality of the set of its linearly independent generators (in the sense of Schauder bases). It is possibly infinite, however it is at most countable whenever $\mathcal{H}$ is a separable[1] space. We will only consider separable spaces.

Given any function $f \in \mathcal{H}$, the best approximating element $\hat{\varphi} \in \mathcal{M}$ to f,

$$\hat{\varphi} = \operatorname*{argmin}_{\varphi \in M} \|f - \varphi\|_{\mathcal{H}},$$

exists, is unique, and is the orthogonal projection of f onto $\mathcal{M}$.

1.1.1 Projection onto an orthonormal basis

Let $\{\phi_n\}_n$ be a family of orthonormal functions that generates $\mathcal{M}$,

$$\langle \phi_n, \phi_l \rangle = \delta(l - n) \qquad \forall l, n,$$

where δ is a Kronecker delta function ($\delta(x) = 1$ if $x = 0$, $\delta(x) = 0$ $\forall x \neq 0$). It is always possible, given a closed subspace $\mathcal{M}$, to construct an orthornormal basis for it.

The orthogonal projection of f onto $\mathcal{M}$ can be explicitly written, in terms of the orthonormal basis elements, as

$$\hat{\varphi} = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \phi_n(x). \tag{1.1}$$

If f belongs to $\mathcal{M}$, then obviously $\hat{\varphi} = f$ and (1.1) is just a perfect-reconstruction formula.

If $\{\phi_n\}_n$ is not orthonormal, but just orthogonal, it suffices to divide each inner product by the squared norm of the corresponding base element, obtaining essentially a formula like (1.1).

1.1.2 Non-orthogonal case: frames

When the family $\{\phi_n\}_n$ is non-orthogonal, but simply a set of generators for $\mathcal{M}$, not necessarily linearly-independent, the best approximation is not given in the simple form (1.1). Of course, one possible approach could be to orthonormalize $\{\phi_n\}_n$ using the Gram-Schmidt procedure. However, in many applications it may be of significant importance to represent signals with respect to specific generators; the Gram-Schmidt procedure typically mixes the generators, and the resulting orthonormal system, while achieving a simple reconstruction formula, might not retain the possibly meaningful structure of the original set of non-orthogonal generators.

We assume that $\{\phi_n\}_n$ is a *frame* [35, 16, 66] for $\mathcal{M}$, i.e. that there exists two constants A and B (usually called the *frame bounds*), $0 < A \leq B < \infty$, such that

$$A \|f\|_{\mathcal{H}}^2 = \sum_n |\langle f, \phi_n \rangle_{\mathcal{H}}|^2 \leq B \|f\|_{\mathcal{H}}^2 \qquad \forall f \in \mathcal{M}.$$

[1] A metric space is said to be separable if there exists a countable dense subset.

A frame constituted by linearly independent functions is called a *Riesz basis*. Given a frame, one can define the *frame analysis operator* $T : \mathcal{M} \to \ell^2(\#\{\phi_n\})$ (where # denotes the cardinality of a set) by

$$Tf = \{\langle f, \phi_n \rangle_{\mathcal{H}}\}_n$$

and its adjont $T^* : \ell^2(\#\{\phi_n\}) \to \mathcal{M}$,

$$\langle f, T^*(\{c_n\}_n)\rangle_{\mathcal{H}} = \langle Tf, \{c_n\}_n \rangle_{\ell^2(\#\{\phi_n\})} = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \overline{c_n} = \sum_n \langle f, c_n \phi_n \rangle_{\mathcal{H}}.$$

The frame bounds ensure the continuity of T and the existence and continuity of the adjoint T^*, in particular $\|T\| = \|T^*\| \leq \sqrt{B}$. Continuity implies that

$$T^*(\{c_n\}_n) = \sum_n c_n \phi_n,$$

and thus T^* is called the *frame synthesis (or reconstruction) operator*.

Let $S = T^*T$. We have

$$Sf = T^*Tf = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \phi_n \quad \forall f \in \mathcal{M}.$$

The operator S is sometimes called the *frame operator*. It can be shown that S is positive and admits an inverse operator S^{-1} (in $\mathcal{M}$). Both S and S^{-1} are self-adjoint. This inverse is used to construct another frame, called *the dual frame*, $\{\breve{\phi}_n\}_n = \{S^{-1}\phi_n\}_n$. It is the dual frame that allows to achieve a formula similar to (1.1) for non-orthogonal systems of generators. First we note that for every element f in $\mathcal{M}$ the following identities hold:

$$\begin{aligned} f = SS^{-1}f &= \textstyle\sum_n \langle S^{-1}f, \phi_n \rangle_{\mathcal{H}} \phi_n = \\ &= \textstyle\sum_n \langle f, S^{-1}\phi_n \rangle_{\mathcal{H}} \phi_n = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \phi_n \\ f = S^{-1}Sf &= S^{-1} \textstyle\sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \phi_n = \\ &= \textstyle\sum_n \langle f, \phi_n \rangle_{\mathcal{H}} S^{-1}\phi_n = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \breve{\phi}_n \end{aligned} \qquad \forall f \in \mathcal{M}. \quad (1.2)$$

The above formulas show that, restricted to $\mathcal{M}$, SS^{-1} is the identity operator on $\mathcal{M}$, $SS^{-1}\big|_{\mathcal{M}} = I_{\mathcal{M}}$. It can also be written in terms of the pseudo-inverse $T^\dagger$ of the operator T as [66]

$$SS^{-1} = (T^*T)^{-1} T^*T = T^\dagger T$$

with $T^\dagger = (T^*T)^{-1} T^*$.

It can be shown that if the frame is a Riesz basis, then the frame and the dual frame are a biorthogonal pair.

The "perfect reconstruction in the subspace" formulas (1.2) lead to the best approximation formulas (or "partial reconstruction") for frames.

Let $\hat{\varphi} \in \mathcal{M}$ be the best approximation of $f \in \mathcal{H}$. Since $\hat{\varphi}$ is the orthogonal projection of f onto $\mathcal{M}$, f can be written as $f = \hat{\varphi} + f^{\perp_{\mathcal{M}}}$ where $f^{\perp_{\mathcal{M}}}$ is the orthogonal complement of f with respect to the subspace $\mathcal{M}$. Therefore,

$$\langle f^{\perp_{\mathcal{M}}}, \varphi \rangle_{\mathcal{H}} = 0 \quad \forall \varphi \in \mathcal{M},$$

and thus $\langle f, \varphi \rangle = \left\langle \hat{\varphi} + f^{\perp_{\mathcal{M}}}, \varphi \right\rangle = \langle \hat{\varphi}, \varphi \rangle$ for all $\varphi \in \mathcal{M}$. In particular, this is true also for the frame and dual frame, thus

$$\begin{array}{l} \hat{\varphi} = \sum_n \langle \hat{\varphi}, \breve{\phi}_n \rangle_{\mathcal{H}} \, \phi_n = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \, \phi_n \\ \hat{\varphi} = \sum_n \langle \hat{\varphi}, \phi_n \rangle_{\mathcal{H}} \, \breve{\phi}_n = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \, \breve{\phi}_n \end{array} \qquad \forall f \in \mathcal{H}. \tag{1.3}$$

These formulas give the best approximation of any element in $\mathcal{H}$ as the analysis and synthesis using a frame $\{\phi_n\}_n$ and its dual $\{\breve{\phi}_n\}_n$. In fact, they show that the analysis-synthesis (by the frame and its dual, respectively) operator is indeed the orthogonal projection operator onto the subspace generated by the frame: $SS^{-1} = T^\dagger T = P_{\mathcal{M}}$.

1.2 The space L^2 with a windowing measure

1.2.1 Definition of the space

It is of particular interest to consider the case where $\mathcal{H} = L^2\left(\mathbb{R}^d, \mu\right)$ and μ is a Lebesgue-Stieltjes measure. The space $L^2\left(\mathbb{R}^d, \mu\right)$ consists of all the functions[2] $f : \mathbb{R}^d \to \mathbb{R}$ such that $\int |f(x)|^2 \, d\mu < \infty$, where the inner product is defined as $\langle f, g \rangle_{\mathcal{H}} = \int f(x) \, \overline{g(x)} d\mu$ and the norm as $\|f\|_{\mathcal{H}} = \left(\int |f(x)|^2 \, d\mu \right)^{1/2}$. As usual, the inner product induces the norm: $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}}$.

If μ is absolutely continuous and denoting by w its density function, it is immediate to check that the integration with respect to the measure is nothing but a weighted/windowed integration (or weighted average) of the function over the support of w, where w is the weighting/windowing function:

$$\int f(x) \, d\mu = \int f(x) \, w(x) \, dx.$$

This formula is valid also for non absolutely continuous measures, but the corresponding w is not an ordinary function but rather a generalized function and the integration must be considered in distributional sense. The discrete case can be thought using directly the Hilbert space ℓ^2 of discrete sequences (and its higher-dimensional counterparts) or, equivalently, by considering L^2 endowed with a piecewise-constant right-continuous discrete measure (with Dirac weights on the integers), and by restricting the considered functions to piecewise-constant functions (with discontinuities on the integers)[3].

[2] In L^2, functions are always considered modulo differences in measure-zero sets (otherwise the integral does not define a norm, but rather a semi-norm, since there might different elements in the space whose "would-be"-norm is zero). Therefore, one should not talk about functions, but rather about equivalence-classes of functions (where the equivalence relation is the pointwise equality almost everywhere, i.e. outside of a set of measure zero). Nevertheless, for the sake of simplicity of language, it is common practice to speak of L^2 as a space of functions, tacitly assuming that the above distinction is understood and the equivalence relation between functions is given for granted. Although usually (e.g. with the ordinary Lebesgue measure), sets of measure zero are intuitively small or meager, in our considerations they can be quite large.

[3] On this aspect, it is worth reminding the following two theorems by Helly (e.g. [58]): (Helly's first theorem) given a sequence of bounded variation measures μ_n on a closed ball U such that the total variations of μ_n on U are bounded by a same constant K_1, $V_U(\mu_n) \leq K_1$ $\forall n$, and $\mu_n \to \mu$ pointwise everywhere in U, then the limit measure μ is also of bounded

For any absolutely continuous measure μ such that w is uniformly bounded, $L^2\left(\mathbb{R}^d, x\right) \subset L^2\left(\mathbb{R}^d, \mu\right)$, i.e. the usual L^2 endowed with the ordinary Lebesgue measure is a subset of $L^2\left(\mathbb{R}^d, \mu\right)$. If w decays to zero at infinity, say rapidly enough, then it is easy to realize that the inclusion is proper. It is an obvious fact for a compactly supported w.

1.2.2 Best approximation in $L^2\left(\mathbb{R}^d, \mu\right)$

Following the general framework illustrated in Section 1.1, we consider the best approximation problem for orthonormal systems as well as for frames.

Let $\{\phi_n^{\text{O N}}\}_n$ and $\{\phi_n\}_n$ be, respectively, an orthonormal system and a frame generating the same closed subspace $\mathcal{M} \subseteq \mathcal{H} = L^2\left(\mathbb{R}^d, \mu\right)$. The orthonormality of $\{\phi_n^{\text{O N}}\}_n$ means that

$$\langle \phi_n^{\text{O N}}, \phi_l^{\text{O N}} \rangle_{\mathcal{H}} = \int \phi_n^{\text{O N}}(x)\, \overline{\phi_l^{\text{O N}}(x)} d\mu = \int \phi_n^{\text{O N}}(x)\, \overline{\phi_l^{\text{O N}}(x)} w\left(x\right) dx = \delta\left(l-n\right) \quad \forall l, n.$$

Just as in the general case discussed in Section 1.1, given any function $f \in L^2\left(\mathbb{R}^d, \mu\right)$, the best approximation $\hat{\varphi}$ of f in $\mathcal{M}$,

$$\hat{\varphi} = \underset{\varphi \in M}{\operatorname{argmin}} \left\| f - \varphi \right\|_{\mathcal{H}} = \underset{\varphi \in M}{\operatorname{argmin}} \int \left| f - \varphi \right|^2 d\mu = \\ = \underset{\varphi \in M}{\operatorname{argmin}} \int \left| f\left(x\right) - \varphi\left(x\right) \right|^2 w\left(x\right) dx, \quad (1.4)$$

is unique and can be obtained by the orthogonal projection of f onto $\mathcal{M}$ as

$$\hat{\varphi} = \sum_n \langle f, \phi_n^{\text{O N}} \rangle_{\mathcal{H}}\, \phi_n^{\text{O N}} = \sum_n \int f(v)\, \overline{\phi_n^{\text{O N}}(v)} d\mu \phi_n^{\text{O N}} = \sum_n \int f(v)\, \overline{\phi_n^{\text{O N}}(v)} w(v) dv \phi_n^{\text{O N}},$$

by using the analysis and synthesis with the frame $\{\phi_n\}_n$ and its dual $\{\breve{\phi}_n\}_n$ as

$$\hat{\varphi} = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}}\, \phi_n = \sum_n \int f\left(v\right) \overline{\breve{\phi}_n(v)} d\mu \phi_n = \sum_n \int f\left(v\right) \overline{\breve{\phi}_n(v)} w\left(v\right) dv \phi_n$$

or, similarly, as

$$\hat{\varphi} = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}}\, \breve{\phi}_n = \sum_n \int f\left(v\right) \overline{\phi_n(v)} d\mu \breve{\phi}_n = \sum_n \int f\left(v\right) \overline{\phi_n(v)} w\left(v\right) dv \breve{\phi}_n.$$

Observe that, because of the homogeneity of the norm, the solution $\hat{\varphi}$ of (1.4) is not affected by a multiplication by a positive constant of the window function w (or of the measure μ).

There are a number of iterative techniques (e.g. the extrapolated Richardson and the conjugate gradient iterations [66]) that allow the computation of the

variation and $\int f d\mu_n \to \int f d\mu$ for every continuous f; (Helly's second theorem) if a sequence of measures μ_n not only satisfies $V_U\left(\mu_n\right) \leq K_1\ \forall n$, but there exists also another constant K_2 such that $\sup_{x \in U, n \in \mathbb{N}} \left|\mu_n\left(x\right)\right| \leq K_2$, then there exist a subsequence μ_{n_k} that converges pointwise everywhere in U (and thus the first theorem can be applied). Since it is possible to approximate, in measure, a step function as accurately as possible by using continuous functions, these two theorems can be used to justify and formalize the use of the continuous-domain analysis for the discrete case.

dual frame. When the number of generators is finite, the dual frame $\breve{\phi}_n$ can be computed from *the Gramian matrix* $\mathbf{\Phi}$ formed by the inner products of the frame elements one against each other [70, 16],

$$\mathbf{\Phi}(i,j) = \langle \phi_i, \phi_j \rangle_{\mathcal{H}} = \int \phi_i(v)\,\overline{\phi_j(v)} d\mu = \int \phi_i(v)\,\overline{\phi_j(v)} w(v)\,dv,$$

as

$$\breve{\phi}_n = \sum_l \mathbf{\Phi}^{\dagger}(l,n)\,\phi_l, \qquad \breve{\phi} = \phi \mathbf{\Phi}^{\dagger}. \tag{1.5}$$

The use of the pseudo-inverse, based on the singular-value decomposition, arises from the fact that the Gramian matrix $\mathbf{\Phi}$ is in general rank-deficient and ill-conditioned (it is however full-rank when the frame is a basis).

1.3 Pointwise evaluation of the approximation and its kernel representation

Suppose that we are interested not in the whole expression of the best approximating element $\hat{\varphi}$ but only in its value at a particular point. Without loss of generality, let us assume that this point is the origin 0 and that $w(0) > 0$, i.e. that the origin belongs to the support of the window $\operatorname{supp} w = \{v \in \mathbb{R}^d : w(v) > 0\}$.

Formally, we obtain for an orthonormal system $\{\phi_n^{\mathrm{ON}}\}_n$,

$$\hat{\varphi}(0) = \sum_n \langle f, \phi_n^{\mathrm{ON}} \rangle_{\mathcal{H}}\, \phi_n^{\mathrm{ON}}(0) = \sum_n \int f(v)\,\overline{\phi_n^{\mathrm{ON}}(v)} w(v)\,dv \phi_n^{\mathrm{ON}}(0), \tag{1.6}$$

or, for a frame $\{\phi_n\}_n$ (which generates the same subspace),

$$\hat{\varphi}(0) = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}}\, \phi_n(0) = \sum_n \int f(v)\,\overline{\breve{\phi}_n(v)} w(v)\,dv \phi_n(0), \tag{1.7}$$

$$\hat{\varphi}(0) = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}}\, \breve{\phi}_n(0) = \sum_n \int f(v)\,\overline{\phi_n(v)} w(v)\,dv \breve{\phi}_n(0). \tag{1.8}$$

However, one should question the meaning of $\hat{\varphi}(0)$, $\phi_n^{\mathrm{ON}}(0)$, $\phi_n(0)$ and $\breve{\phi}_n(0)$. Since we are in L^2, the value of a function at a particular point is not a well-defined quantity. Even more, the convergence in L^2 does not imply the pointwise convergence, whereas the above formulas assume implicitly the pointwise convergence at 0. There are a number of theorems that ensure this convergence under particular hypotheses, the main of such hypotheses being that 0 is a Lebegue point for the considered functions. In the case of the classical Fourier series, it is interesting to point out that the continuity of f is not enough to ensure pointwise convergence, and in particular[4] for every point x there exist a continuous function such that its Fourier series diverges at x. Nevertheless, bounded variation of the function guarantees the pointwise convergence of its Fourier series everywhere (e.g. [49]). A proper setting for this problem lies in the theory of *reproducing kernel Hilbert spaces*[5] (*RKHS*) (see [64, 63]). For a

[4] because of the Banach-Steinhaus unboundedness theorem.

[5] A Hilbert space of functions is said to be a *reproducing kernel Hilbert space* (RKHS) if the evaluation-at-x functional is continuous for every x in the domain of the functions. In an RKHS, the evaluation-at-x functional can be then expressed, according to the Riesz representation theorem, as the inner product against a function K_x. The *reproducing kernel* K is defined as $K(x,y) = \langle K_x, K_y \rangle$. It has the property that $\langle f, K(x,\cdot) \rangle = f(x)\ \forall x$.

generic frame and measure the formulation of such convergence theorems becomes rather involved (see e.g. [50], [103]).

To avoid these complications (which go far beyond the scope of this chapter), and since in practice nobody uses infinitely many frame elements, we restrict our attention to a finite set of generators. Moreover, we assume that all the reconstructing elements are continuous at 0 (or, more weakly, that 0 is a Lebegue point for all of them). This easily guarantees that the reconstruction of the best approximating element is well-defined in pointwise sense at 0.

Under these additional assumptions, formulas (1.6-1.8) no longer have only formal meaning, and take the form

$$\begin{aligned}
\hat{\varphi}(0) &= \sum_n \int f(v)\,\overline{\phi_n^{\mathrm{ON}}(v)} w(v)\,dv \phi_n^{\mathrm{ON}}(0), \\
\hat{\varphi}(0) &= \sum_n \int f(v)\,\overline{\sum_l \mathbf{\Phi}^{\dagger}(l,n)\,\phi_l(v)} w(v)\,dv \phi_n(0), \\
\hat{\varphi}(0) &= \sum_n \int f(v)\,\overline{\phi_n(v)} w(v)\,dv \sum_l \mathbf{\Phi}^{\dagger}(l,n)\,\phi_l(0),
\end{aligned}$$

or, equivalently,

$$\begin{aligned}
\hat{\varphi}(0) &= \int f(v)\left(w(v)\sum_n \overline{\phi_n^{\mathrm{ON}}(v)}\phi_n^{\mathrm{ON}}(0)\right)dv = \int f(v)\,g_{\mathcal{M}}(v)\,dv, \\
\hat{\varphi}(0) &= \int f(v)\left(w(v)\sum_n \overline{\sum_l \mathbf{\Phi}^{\dagger}(l,n)\,\phi_l(v)}\phi_n(0)\right)dv = \\
&= \int f(v)\left(w(v)\sum_{l,n} \overline{\phi_l(v)\,\mathbf{\Phi}^{\dagger}(l,n)}\phi_n(0)\right)dv = \int f(v)\,g_{\mathcal{M}}(v)\,dv, \\
\hat{\varphi}(0) &= \int f(v)\left(w(v)\sum_n \overline{\phi_n(v)}\sum_l \mathbf{\Phi}^{\dagger}(l,n)\,\phi_l(0)\right)dv = \\
&= \int f(v)\left(w(v)\sum_{l,n} \overline{\phi_n(v)}\mathbf{\Phi}^{\dagger}(l,n)\,\phi_l(0)\right)dv = \int f(v)\,g_{\mathcal{M}}(v)\,dv.
\end{aligned}$$

It is easy to realize that the three kernels $g_{\mathcal{M}}$ from the three lines above are actually exactly identical in $\mathcal{H} = L^2\left(\mathbb{R}^d, \mu\right)$, since they represent three identical bounded[6] linear functionals from the dual space $\mathcal{H}^*$.[7]

Summarizing, given any function in $f \in L^2\left(\mathbb{R}^d, \mu\right)$, let $\varphi \in \mathcal{M}$ be the best approximating element to f in $\mathcal{M}$, and

$$\hat{\varphi}(0) = \int f(v)\,g_{\mathcal{M}}(v)\,dv,$$

[6] Boundedness can be obtained easily from the frame bound since $|\langle f, \phi_n\rangle| \leq \sqrt{\sum_l |\langle f, \phi_l\rangle|^2} \leq \sqrt{B}\,\|f\|_{\mathcal{H}}$ and the sums are finite: $|\hat{\varphi}(0)| \leq \sum_n \langle f, \phi_n\rangle \left|\breve{\phi}_n(0)\right| \leq \|f\|_{\mathcal{H}} \sqrt{B} \sum_n \left|\breve{\phi}_n(0)\right| = \|f\|_{\mathcal{H}} \cdot$constant. This holds because of the finite number of generators. If they were infinitely many, one would need to verify that $\|\langle f, \phi\rangle\|_{\ell^1} \leq \|f\|_{\mathcal{H}} \cdot$constant$\leq A^{-\frac{1}{2}} \|\langle f, \phi\rangle\|_{\ell^2} \cdot$constant, which in general does not hold (the opposite inequality holds).

[7] To be precise, the kernel that represents the functional is not $g(v)$ but $\overline{g(v)}/w(v)$, since the inner product in $\mathcal{H}$ has always the factor $w(v)$ from the measure μ and the complex conjugation.

where the kernel $g_{\mathcal{M}}$ is expressed as

$$g_{\mathcal{M}}(v) = w(v) \sum_{l,n} \overline{\phi_l(v)\, \mathbf{\Phi}^{\dagger}(l,n)} \phi_n(0), \tag{1.9}$$

using the frame elements $\{\phi_n\}_n$. Roughly speaking, given a finite set of generators we obtain the value of the best approximation in the origin as the integration against a particular kernel. A fact of remarkable importance for applications is that this kernel can be precalculated, since it does not depend on the function f but only on $\mathcal{M}$ (i.e. the span of $\{\phi_n\}_n$ and the measure μ).

1.4 Moving least-squares (*MLS*) method

Let a function $z : \mathbb{R}^d \to \mathbb{R}$ be such that $\tilde{z}_x(\cdot) = z(x - \cdot) \in L^2\left(\mathbb{R}^d, \mu\right)$ for all $x \in \mathbb{R}^d$. We can then compute, using the above procedure, the value in the origin of the best approximation $\hat{\varphi}_x$ of $\tilde{z}_x$, $\hat{\varphi}_x = \operatorname{argmin}_{\varphi \in \mathcal{M}} \|\varphi - \tilde{z}_x\|_{\mathcal{H}}$, as

$$\hat{\varphi}_x(0) = \int \tilde{z}_x(v)\, g_{\mathcal{M}}(v)\, dv = \int z(x - v)\, g_{\mathcal{M}}(v)\, dv = (z \circledast g_{\mathcal{M}})(x). \tag{1.10}$$

We immediately get that the convolution against $g_{\mathcal{M}}$ yields, for every point x, the pointwise value (in x) of the best approximation of the function z in the subspace $\tilde{\mathcal{M}}_x = \{\varphi : \varphi(x - \cdot) \in \mathcal{M}\}$. Here, "best approximation" and the "subspace" structure are intended with respect to the windowing measure $\tilde{\mu}_x$ ("centered" at x) defined as $\tilde{\mu}_x(\cdot) = -\mu(x - \cdot)$, that is implicitly imposed on a superspace, say $\tilde{\mathcal{H}}_x = L^2\left(\mathbb{R}^d, \tilde{\mu}_x\right)$, of $\tilde{\mathcal{M}}_x$.

By letting x vary in $\mathbb{R}^d$, we obtain

$$z_{\mathcal{M}}^{MLS} = \hat{\varphi}_{(\cdot)}(0) = z \circledast g_{\mathcal{M}}. \tag{1.11}$$

The approximation approach corresponding to equation (1.11) is called the *moving least-squares (MLS) method*, and $z_{\mathcal{M}}^{MLS}$ is the *MLS-estimate* of z (onto the space $\mathcal{M}$). The term "moving" refers to the fact that, as the point x moves in $\mathbb{R}^d$, a pointwise-varying space-subspace pair $\left(\tilde{\mathcal{H}}_x, \tilde{\mathcal{M}}_x\right)$ is considered for the approximation. The kernel $g_{\mathcal{M}}$ is sometimes called the *moving kernel*, and, for a fixed x, the coordinates $\{(x - \cdot)\}$ are the *moving coordinates*.

1.4.1 Translation-invariance

According to the previous definitions, we have that $z \in \tilde{\mathcal{H}}_x \ \forall x$, and, more generally, that the space-subspace pairs are translation-invariant:

$$\begin{gathered} f \in \tilde{\mathcal{H}}_x \iff f(\cdot + \delta) \in \tilde{\mathcal{H}}_{x+\delta} \iff f(\cdot - x) \in \tilde{\mathcal{H}}_0 \iff f(x - \cdot) \in \mathcal{H}, \\ \|f\|_{\tilde{\mathcal{H}}_x} = \|f(\cdot + \delta)\|_{\tilde{\mathcal{H}}_{x+\delta}} = \|f(\cdot - x)\|_{\tilde{\mathcal{H}}_0} = \|f(x - \cdot)\|_{\mathcal{H}}, \\ f \in \tilde{\mathcal{M}}_x \iff f(\cdot + \delta) \in \tilde{\mathcal{M}}_{x+\delta} \iff f(\cdot - x) \in \tilde{\mathcal{M}}_0 \iff f(x - \cdot) \in \tilde{\mathcal{M}}. \end{gathered}$$

This translation-invariance is directly linked with the use of the convolution operation $\circledast$ in (1.11), or, more precisely, with the fact that the kernel $g_{\mathcal{M}}$ in the integral $\int z(x - v)\, g_{\mathcal{M}}(v)\, dv$ does not depend on x. Indeed, (1.10) can be

interpreted – using the signal-processing terminology – as a particular *linear time-invariant filter*.

One may wish to consider a more flexible definition of the form

$$\hat{\varphi}_x(0) = \int z(x-v)\, g_{\mathcal{M}_x}(v)\, dv, \qquad z^{MLS} = \hat{\varphi}_{(\cdot)}(0), \tag{1.12}$$

where the kernel $g_{\mathcal{M}_x}$ – or, equivalently, the space $\mathcal{M}_x$ – depends on the point x. Although in (1.12) $\hat{\varphi}_x(0) = \int z(x-v)\, g_{\mathcal{M}_x}(v)\, dv = (z \circledast g_{\mathcal{M}_x})(x)$ is indeed – for any fixed x – a convolution, z^{MLS} is *not* obtained from a convolution![8]

Estimates of the form (1.12) are discussed in Chapter 2 for a specific family of subspaces. In Section 2.4, we introduce an algorithm, so-called *ICI* rule, that selects a pointwise-adaptive $g_{\mathcal{M}_x}$ depending on the function z.

For the time being, let us return to the standard *convolutional MLS* (1.10), in which the approximation of z is obtained as a convolution against a single – say "space-invariant" – kernel $g_{\mathcal{M}}$.

1.4.2 Moving least-squares denoising

One of the most relevant applications of the moving least-square method is denoising.

Let noisy observations be given in the form $z = y + \eta$, where y is the true (typically unknown) signal and η is some noise. The "moving least-squares on $\mathcal{M}$" estimate $\hat{y}$ of y is given by the convolution

$$\hat{y} = z \circledast g_{\mathcal{M}} = \int z(\cdot - v)\, g_{\mathcal{M}}(v)\, dv,$$

where the kernel $g_{\mathcal{M}}$ corresponds to an appropriate closed subspace $\mathcal{M}$.

"Appropriate" means that $\tilde{y}_x = y(\cdot - x)$ is well-approximated in $\mathcal{M}$ (i.e. low bias) and that η is concentrated in the orthogonal complement $\mathcal{M}^{\perp}$ of $\mathcal{M}$ (i.e. low variance).

The above formula has general validity also for the discrete case, as long as the data grid is equispaced (i.e. uniform sampling) and unbounded[9] because these two facts allow to use the same windowing function w for every point.

[8] For clarity, from $\hat{\varphi}_x(0) = (z \circledast g_{\mathcal{M}_x})(x)\ \forall x \in \mathbb{R}^d$, does not follow that $z^{MLS} = z \circledast g_{\mathcal{M}_x}$, because x appears not only as the function's argument (evaluation of the convolution), but also as a parameter for the convolution kernel. Mathematically speaking, there is no such thing as a "convolution against a space-variant kernel", and the equation $z^{MLS}(x) = \int z(x-v)\, g_{\mathcal{M}_x}(v)\, dv = (z \circledast g_{\mathcal{M}_x})(x)$ has only pointwise meaning (i.e. for a fixed x).

[9] The unboundedness of the discrete domain is a technicality, required in order to enable the definition of the convolution against $g_{\mathcal{M}}$. However, in practice it is enough to evaluate the convolution only within the domain of the image (or of the data of interest), which is typically bounded. To do so, it suffices that, for every x in the image domain, the support of $g_{\tilde{\mathcal{M}}_x} = g_{\mathcal{M}}(x - \cdot)$ does not trespass the boundary of an imaginary "operational" domain, which is added outside the image domain. These are the so-called *boundary conditions*.

It is common practice to fictiously define the data outside the image domain as zero, or according to some periodicization of the image.

Our choice, is to define this fictious operational data equal to a very large constant, say $1/\varepsilon$, so that the estimates that are computed using this data, which does not belong to the actual observations, are clearly and unmistakably invalidated and discarded. Roughly speaking, we may say that our data is confined by an infinitely-high wall.

1.5 Local polynomial approximation (*LPA*)

The Local Polynomial Approximation (*LPA*) is simply the moving least-squares method[10] where the subspace $\mathcal{M}$ is a closed subspace of real polynomials (i.e. a subspace of real polynomials whose order is bounded). The term "local" refers to the window function w.

1.5.1 Characterization of an *LPA*

In the construction of an *LPA*, only the subspace $\mathcal{M}$ needs to be specified, i.e. $\mathcal{M}$ as a set of functions and the window (or the measure) which defines the inner product and norm that endow the set with a space structure. These two items completely characterize an *LPA*.

As a set, the subspace can be defined through a family of linearly independent polynomials that generates it. This linear independence needs to be checked on the support of the window: while in the continuous case this is a trivial requirement (unless the support of w is a finite set), in the discrete case one has to ensure that there are enough samples in the support of w.

The usual choice is to define it by means of a maximum polynomial order m. For the higher-dimensional case, compact multi-index notation is used: $m = (m_1, \ldots, m_d)$, $v = (v_1, \ldots, v_d)$, $o = (o_1, \ldots, o_d)$, and $v^o = (v_1^{o_1} \cdots v_d^{o_d})$. With this notation, as a set $\mathcal{M}$ is defined as

$$\mathcal{M} = \left\{ \varphi : \varphi(v) = \sum\nolimits_{o=0}^{m} c_o v^o,\ c_o \in \mathbb{R} \right\}.$$

It is common practice to impose that $|o| = o_1 + \cdots + o_d \leq \max_i \{m_i\}$. We denote the class of all admissible orders as O_m. With this notation, we can express $\mathcal{M}$ as

$$\mathcal{M} = \left\{ \varphi : \varphi(v) = \sum\nolimits_{o \in O_m} c_o v^o,\ c_o \in \mathbb{R} \right\}. \tag{1.13}$$

1.5.2 Function and derivative estimation kernels

Clearly, $\hat{\varphi}$ does not depend on the frame used to construct $\mathcal{M}$. Since the linear independence of the system of generators guarantees the invertibility of the Gramian matrix[11], bases (for $\mathcal{M}$) of linearly independent polynomials are favourable. Traditionally, orthonormal bases[12] have been preferred because the corresponding reconstruction formula has a simpler form. Nevertheless, it is

[10] In the literature the term "moving least-squares method" is often used to denote the particular case of the *LPA*. However, in this text we prefer to present the "moving least-squares method" as a general technique that can also be used for approximations other than the polynomial one.

[11] Nevertheless, the use of the pseudoinverse may be required in order to achieve the numerical stability of the solution.

[12] Such families of orthonormal polynomials $\{\phi_n\}_n$ can be obtained via the Gram-Schmidt orthonormalization procedure from the Taylor basis $\left\{ \frac{x^n}{n!} \right\}_{n=0}^{m}$, or from any other collection of linearly independent polynomials that span all possible orders up to m. Depending on the measure μ, this procedure yields - starting from the Taylor basis - various families of well-known orthogonal polynomials. For example, when $w(x) = \chi_{[-1,1]}$ (i.e. when μ is any multiple of the ordinary Lebesgue measure $\mu(x) = x$ on the interval $[-1, 1] \subset \mathbb{R}$), we obtain the Legendre polynomials, or, when $w(x) = \mu'(x) = \frac{1}{\sqrt{1-x^2}}$, the Chebyshev polynomials.

very convenient to use a basis whose generators are monomes, and in particular an ideal choice is to use the standard Taylor basis,

$$\{\phi_n\}_{n=1}^{N} = \left\{\frac{v^o}{o!}\right\}_{o\in O_m}, \tag{1.14}$$

where $o! = o_1! \cdots o_d!$. The previous equality defines the correspondence between the subindex $n \in \mathbb{N}$ and the order $o = o(n) \in \mathbb{N}^d$ of the corresponding monome $\phi_n(v) = \frac{v^{o(n)}}{o(n)!}$. Indeed, since $\phi_n(0) = \delta(n-1)$, the kernel expression is much simplified:

$$g_{\mathcal{M}}(v) = w(v) \sum_{l,n} \phi_l(v) \mathbf{\Phi}^{\dagger}(l,n) \phi_n(0) = w(v) \sum_{l} \phi_l(v) \mathbf{\Phi}^{-1}(l,1). \tag{1.15}$$

More importantly, since $\hat{\varphi}$ is expressed in terms of the Taylor basis, the coefficients (i.e. the inner products against the dual frame) are, up to a change of sign, the values of the (partial) derivatives[13] in the origin $\hat{\varphi}$. Precisely,

$$\hat{\varphi} = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \phi_n = \sum_n \left(\int f(v) \sum_l \mathbf{\Phi}^{-1}(l,n) \phi_l(v) w(v) dv \right) \phi_n = \\ = \sum_n \left((-1)^{|o(n)|} \left(D^{(o(n))} \hat{\varphi} \right)(0) \right) \phi_n,$$

where $D^{(o)} = \frac{\partial^{|o|}}{\partial v^o} = \frac{\partial^{o_1}}{\partial v_1^{o_1}} \cdots \frac{\partial^{o_d}}{\partial v_d^{o_2}}$ and $|o| = o_1 + \cdots + o_d$. Since the coefficients for the reconstruction are uniquely defined (because $\hat{\varphi} \in \mathcal{M}$ and the generators are a basis for $\mathcal{M}$) it follows that

$$\int f(v) \sum_l \mathbf{\Phi}^{-1}(l,n) \phi_l(v) w(v) = (-1)^{|o(n)|} \left(D^{(o(n))} \hat{\varphi} \right)(0).$$

This allows to generalize formula (1.15) to a family of kernels,

$$g_{\mathcal{M}}^{(o(n))}(v) \triangleq w(v) \sum_l \phi_l(v) \mathbf{\Phi}^{-1}(l,n), \tag{1.16}$$

such that the convolution against them yields an estimate for the function and all its derivatives (up to the order m),

$$\hat{y}^{(o(n))} = z \circledast g_{\mathcal{M}}^{(o(n))} = \int z(\cdot - v) g_{\mathcal{M}}^{(o(n))}(v) dv,$$

where $\hat{y}^{(o)}(x) = (-1)^{|o|} \left(D^{(o)} \hat{\varphi}_x \right)(0)$ and $\hat{\varphi}_x = \operatorname{argmin}_{\varphi \in \mathcal{M}} \|\varphi - \tilde{z}_x\|_{\mathcal{H}}$. Because of the change of variables inside of the convolution operation,

$$\hat{y}^{(o)}(x) = \left(D^{(o)} \widetilde{\hat{\varphi}_x} \right)(x), \tag{1.17}$$

with $\widetilde{\hat{\varphi}_x} = \hat{\varphi}_x(\cdot - x)$ being the best approximation of z in the space $\tilde{\mathcal{M}}_x$ (see Section 1.4). Thus, $\hat{y}^{(o)}(x)$ is in fact an estimate of the o-th (partial) derivative of y. The kernel $g_{\mathcal{M}}(v) = g_{\mathcal{M}}^{(0)}(v)$ is called a *function-estimation kernel* and the kernels $g_{\mathcal{M}}^{(o)}(v)$, $o \neq 0$, are called *derivative-estimation kernels*.

[13] Strictly speaking, derivatives have sense only in the continuous domain. In the discrete domain they can be defined in many different ways, e.g. by finite-differencing, or more generally, by, means of Taylor expansions or approximations (in fact, continuous-domain considerations, such as those we made above, can be taken as justifications of the definition for the discrete case). The subject of derivation of discrete functions is discussed more extensively in Chapter 7.

1.5.3 Vanishing moments

Because of the perfect-reconstruction property for polynomials in $\mathcal{M}$, the following equations, or *moments conditions*, hold[14],

$$\int g_{\mathcal{M}}(v)\,dv = 1, \tag{1.18}$$

$$\int v^o g_{\mathcal{M}}(v)\,dv = 0, \quad o \in O_m \setminus \{0\}. \tag{1.19}$$

The relations from the last equation are often called *vanishing moments conditions*.

1.5.4 Zero-order *LPA*

For the zero-order *LPA* (i.e. $m = 0$), the set of generators consists of only the constant term, $\{\phi_n\}_n = \{\phi_1\} = \{1\}$, and the function estimation kernel coincides with the normalized window function,

$$g_{\mathcal{M}} = w\phi_1 \left(\int \phi_1(v)\,\phi_1(v)\,w(v)\,dv \right)^{-1} = \frac{w}{\int w(v)\,dv}. \tag{1.20}$$

Obviously, there are no derivative estimation kernels.

1.6 Finite case and matrix notation

In this section we rewrite the formulas for the general weighted least-squares approximation using vector/matrix notation. This is particularly useful for the numerical implementation of the moving-least squares method.

Let us consider the case where the number of generators N (elements of the frame) is finite and the data grid has a finite number of samples I. The sample points are denoted by v_i, $i = 1, \ldots, I$. The data grid is thus represented as a vector[15]. Each one of the generators ϕ_n is written as a column vector

$$\phi_n = \begin{bmatrix} \phi_n(v_1) \\ \vdots \\ \phi_n(v_I) \end{bmatrix},$$

and the whole frame is then representable as an $N \times I$ matrix $\boldsymbol{\phi}$, the *frame matrix*[16],

$$\boldsymbol{\phi} = [\phi_1 \cdots \phi_N] = \begin{bmatrix} \phi_1(v_1) & \cdots & \phi_N(v_1) \\ \vdots & & \vdots \\ \phi_1(v_I) & \cdots & \phi_N(v_I) \end{bmatrix}.$$

[14] One and zero are, respectively, the values in the origin of the constant function identically equal to one and of the polynomial v^o.

[15] Although we represent the sample points as a vector, it does not mean that the data is necessarily one-dimensional, since discrete higher-dimensional data can be "reshaped" into a 1D vector.

[16] If $\{\phi_n\}_n$ is an orthonormal system, the corresponding frame matrix is an orthonormal matrix with respect to the weights w, i.e. $\boldsymbol{\phi}^T \mathbf{w} \boldsymbol{\phi} = \mathbf{I}$.

Similarly, a function $f \in \mathcal{H}$ can be written as the column vector

$$f = \begin{bmatrix} f(v_1) \\ \vdots \\ f(v_I) \end{bmatrix}.$$

With this notation, the operators T and T^* can be written as

$$Tf = \{\langle f, \phi_n \rangle_{\mathcal{H}}\}_n^N = \begin{bmatrix} \langle f, \phi_1 \rangle_{\mathcal{H}} \\ \vdots \\ \langle f, \phi_N \rangle_{\mathcal{H}} \end{bmatrix} = \boldsymbol{\phi}^T \mathbf{w} f = \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix},$$
$$T^* \left(\{c_n\}_{n=1}^N \right) = \sum_n c_n \phi_n = \boldsymbol{\phi} \mathbf{c},$$

where $\mathbf{w} = \operatorname{diag}([w(v_1) \cdots w(v_I)])$ is the diagonal matrix composed by the weights w. In other words, multiplication of the frame matrix against a coefficient vector achieves the reconstruction operation, while multiplication against the conjugate transpose matrix achieves the analysis (i.e. it yields the inner products between the vector and the frame elements). In this last operation, the weights must be placed accordingly to the windowing measure which defines the inner products.

According to this matrix notation and analogously to formula (1.5), the dual frame $\{\breve{\phi}_n\}_{n=1}^N$ can be expressed by the *dual frame matrix* as

$$\breve{\boldsymbol{\phi}} = \boldsymbol{\phi} \left(\boldsymbol{\phi}^T \mathbf{w} \boldsymbol{\phi} \right)^\dagger, \tag{1.21}$$

The matrix $\boldsymbol{\phi}^T \mathbf{w} \boldsymbol{\phi} = \boldsymbol{\Phi}$ is again the Gramian matrix, formed by the inner products of the frame elements one against each other, and † denotes the pseudo-inverse.

1.6.1 Best approximation (weighted least-squares solution)

From (1.21) it is straightforward to derive the equivalent matrix form of the equations (1.3):

$$\begin{array}{l} \hat{\varphi} = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \phi_n = \boldsymbol{\phi} \breve{\boldsymbol{\phi}}^T \mathbf{w} f = \boldsymbol{\phi} (\boldsymbol{\phi} \boldsymbol{\Phi}^\dagger)^T \mathbf{w} f = \boldsymbol{\phi} \boldsymbol{\Phi}^\dagger \boldsymbol{\phi}^T \mathbf{w} f, \\ \hat{\varphi} = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \breve{\phi}_n = \breve{\boldsymbol{\phi}} \boldsymbol{\phi}^T \mathbf{w} f = \boldsymbol{\phi} \boldsymbol{\Phi}^\dagger \boldsymbol{\phi}^T \mathbf{w} f. \end{array} \quad \forall f \in \mathcal{H}. \tag{1.22}$$

1.6.2 Pointwise evalution of the best approximation and corresponding kernel

If we are interested only in the value of the best approximation at a particular point, say $v_1 = 0$, then from formula (1.22) we obtain

$$\hat{\varphi}(0) = [\phi_1(0) \cdots \phi_N(0)] \boldsymbol{\Phi}^\dagger \boldsymbol{\phi}^T \mathbf{w} f.$$

Thus, the matrix form of the kernel $g_{\mathcal{M}}$ is

$$g_{\mathcal{M}} = \left([\phi_1(0) \cdots \phi_N(0)] \boldsymbol{\Phi}^\dagger \boldsymbol{\phi}^T \mathbf{w} \right)^T = \mathbf{w} \boldsymbol{\phi} \boldsymbol{\Phi}^\dagger [\phi_1(0) \cdots \phi_N(0)]^T.$$

1.6.3 Vector form for *LPA* function and derivative estimation kernels

For the standard *LPA* (generated by the Taylor polynomials), the function estimation kernel takes the vector form

$$g_{\mathcal{M}} = \mathbf{w}\boldsymbol{\phi}\boldsymbol{\Phi}^{-1}\left[1\ 0\cdots 0\right]^{T}.$$

Derivative estimation kernels $g_{\mathcal{M}}^{(o)}$ are expressed by

$$g_{\mathcal{M}}^{(o(n))} = \mathbf{w}\boldsymbol{\phi}\boldsymbol{\Phi}^{-1}\left[0\cdots 0\ 1\ 0\cdots 0\right]^{T},$$

where the "indicator" vector on the right-hand side has the one in the n-th place.

1.7 Some examples of *LPA* kernels

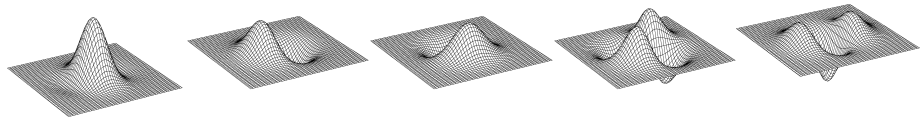
Figure 1.1, 1.2, and 1.3 show some examples of function and derivative estimation discrete *LPA* kernels calculated for various maximum orders m and windowing functions w. For the case $m = (2,1)$ – illustrated in Figure 1.1 and Figure 1.2 for two different w – the admissible orders that satisfy $o \leq m$, $|o| \leq \max_i \{m_i\}$ are $O_m = \{(0,0),(1,0),(0,1),(1,1),(2,0)\}$ and the corresponding basis functions $\phi_n(v) = \frac{v^{o(n)}}{o(n)!}$ used in the generation of $\mathcal{M}$ are

$$\phi_1 = 1, \quad \phi_2 = v_1, \quad \phi_3 = v_2, \quad \phi_4 = v_1 v_2, \quad \text{and} \quad \phi_5 = \frac{v_1^2}{2}.$$

A much larger set of the generators is used when $m = (5,3)$. To each one of them corresponds a function or derivative estimation kernel. Figure 1.3 shows only a few of them.

f ∂_{x_1} ∂_{x_2} $\partial^2_{x_1x_2}$ $\partial^2_{x_1^2}$

Figure 1.1: Function and derivative estimation *LPA* kernels obtained for $m = (2,1)$ using the characteristic function of a disk as the window function w.



f ∂_{x_1} ∂_{x_2} $\partial^2_{x_1x_2}$ $\partial^2_{x_1^2}$

Figure 1.2: Function and derivative estimation *LPA* kernels obtained for $m = (2,1)$ using a Gaussian window function w.

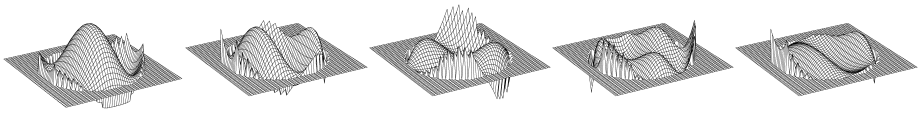


Figure 1.3: The function estimation *LPA* kernel and some of the derivative estimation *LPA* kernels obtained for $m = (5, 3)$ using the characteristic function of a disk as the window function w.

Chapter 2

Adaptive nonparametric estimation

2.1 Parametric vs. nonparametric estimation

The most widely used techniques in signal and image processing are based on transform-domain representations. The Fourier transform, the wavelet decompositions, and the discrete cosine transform (DCT), are probably the best known examples of such techniques.

The most appealing and desired property of any transform-domain representation is *sparsity*. It means that the signal can be well represented in terms a few significant transform-domain coefficients. The efficiency of transform-domain techniques largely depends on this sparsity. Hence, a lot of research has been done into the design of bases or frames that can represent sparsely the widest possible range of useful or significant signals.

Strictly speaking, and using a terminology from statistics, the quest for sparsity corresponds to trying to describe the data using as few as possible model parameters. Such model parameters, namely the transform coefficients, represent the signal on a global level.

The so-called nonparametric approach is radically different. No restriction is assumed on the number of model parameters, in fact, no global parametric representation is assumed at all. It offers an original approach to signal processing problems (e.g. [40], [15], [41], [65]). It basically results in kernel filtering with the kernels designed using some moving window local approximations. The *LPA* is probably the most significant nonparametric method in the literature. The signal is described only locally, and every point is characterized by its own local model, which can be independent from the models corresponding to all other points.

Adaptive versions of these algorithms are able to produce efficient filtering with a level of localization (scale, bandwidth) which is pointwise-adaptive (e.g. [67], [78]). This pointwise-adaptive scale selection is based on the following idea, known as the Lepski's approach. The algorithm searches for a largest local vicinity of the point of estimation where the estimate fits well to the data. The estimates $\hat{y}_h(x)$ are calculated for a set of window sizes (scales) $h \in H$ and compared. The adaptive scale is defined as the largest of those for which

estimate does not differ significantly from the estimators corresponding to the smaller window sizes.

The intersection of confidence intervals (*ICI*) rule ([28],[42]) is one of the versions of this approach, and appeared to be quite efficient for the adaptive scale image restoration [43, 44, 46, 20, 22, 47, 23, 21, 14, 24].

Curiously, the long-term evolution of the transform-domain techniques for image restoration shows a distinct trend towards localization (e.g. compactly supported wavelet decompositions [66]) and an increased number of parameters (e.g. frames, overcomplete expansions [90], translation invariant filtering [9, 8]). This trend not only suggests the inadequateness of conventional models, but – to some extent – hints that the peculiarities of the nonparametric approach are of significant importance for a superior image-restoration filtering.

2.2 Scale

The "scale" of a kernel will probably be the most frequently encountered concept in this thesis. Despite its importance, the definition of scale is rather simple.

In the continuous domain, since Taylor monomes are used for the basis elements for the space $\mathcal{M}$, a change of variable in the window w,

$$w_h(\cdot) \triangleq w(\cdot/h), \qquad h \in \mathbb{R}^+,$$

yields, by equation (1.15), a function estimation kernel that can be obtained also be the direct change of variables

$$g_h(\cdot) = h^{-d} g_{\mathcal{M}}(\cdot/h). \tag{2.1}$$

The parameter h is called the *scale* of the kernel[1]. This definition of scale is relative, since it depends on w. To make it absolute, it is usually, often tacitly, assumed that w has unitary size, length, or diameter. Thus, h corresponds to the size, length, or diameter of w_h (and hence of g_h).

Formulas analogous to (2.1) hold for the derivative kernels as well,

$$g_h^{(o)}(\cdot) = h^{-d} h^{-|o|} g_{\mathcal{M}}^{(o)}(\cdot/h), \tag{2.2}$$

and the function and derivative estimates can be obtained as

$$\hat{y}_h = \int z(\cdot - v)\, g_h(v)\, dv = \\ = \int z(\cdot - v)\, h^{-d} g_{\mathcal{M}}(v/h)\, dv = \int z(\cdot - hv)\, g_{\mathcal{M}}(v)\, dv,$$

$$\hat{y}_h^{(o)} = \int z(\cdot - v)\, g_h^{(o)}(v)\, dv = \\ = \int z(\cdot - v)\, h^{-d} h^{-|o|} g_{\mathcal{M}}^{(o)}(v/h)\, dv = h^{-|o|} \int z(\cdot - hv)\, g_{\mathcal{M}}^{(o)}(v)\, dv.$$

[1] Although it is possible to consider also a vector scale parameter $h = (h_1, \ldots, h_d)$, $h_i \in \mathbb{R}^+$, for the sake of simplicity, we will present here only the scalar case $h \in \mathbb{R}^+$.

2.2.1 *LPA* kernels as smoothers

Equation (2.1) makes rather easy to realize that *LPA* kernels can be used effectively as smoothers, where the scale h acts as a bandwidth parameter. In fact, by going into the frequency domain, we have

$$\hat{Y}_h = ZG_h = Z\mathcal{F}\left(h^{-d} g_{\mathcal{M}}\left(\cdot/h\right)\right) = Zh^{-d}\mathcal{F}\left(g_{\mathcal{M}}\left(\cdot/h\right)\right) = ZG_{\mathcal{M}}\left(h\cdot\right),$$

where $\mathcal{F}$ is used to denote the Fourier transform[2]. Indeed, the moment condition $\int g_{\mathcal{M}} = 1$ implies that $G_{\mathcal{M}}(0) = 1$. Since the decay-rate towards infinity of the Fourier transform of a function depends on the smoothness (in the sense of order of differentiability) of the function itself, and recalling formula (1.15) – which states that *LPA* kernels are obtained as a finite sum of windowed polynomials – it is clear that the decay-rate of the *LPA* kernels depends only on the smoothness of the window function w. Traditional window functions are designed in such a way that the desired frequency response of the resulting kernel is achieved. In practice, $G_{\mathcal{M}}$ works as a low-pass filter.

Figure 2.1 illustrates the low-pass property of the *LPA* kernels, showing four different function-estimation kernels obtained from four different values of the scale parameter h: as h increases, the band shrinks around the origin, where the frequency response is one because of the moment condition.

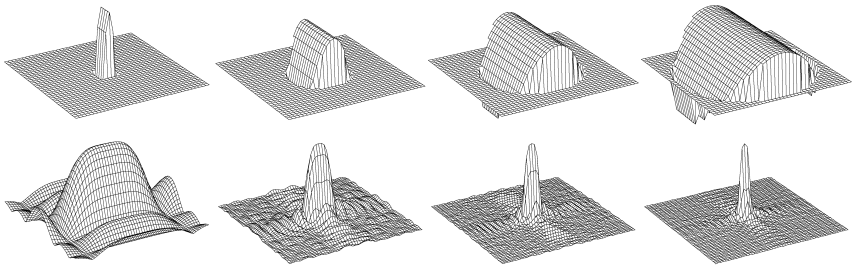


Figure 2.1: Function estimation *LPA* kernels obtained for $m = (2, 1)$ using the characteristic function of disk as the window function w_h. Kernels are shown for four different values of h, from small (left) to large (right). The absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure. To improve the visualization, the vertical scale of the space-domain plots is larger for the kernels of a larger scale.

2.3 Accuracy analysis of the *LPA* kernels

To use the *LPA* as a denoising tool, it is of fundamental importance to understand the statistical properties of the *LPA* estimates. In this section we discuss the mean squared error of these estimates, following the standard approach based on the bias and variance analysis. These two quantities will be expressed as functions of the scale parameter h, leading to an analytical formulation of the bias-variance tradeoff for varying-scale *LPA* estimates. Minimization of the MSE with respect to h will provide us with an *ideal* estimate. The bias and

[2] We always consider a normalized Fourier transform such that it realizes an isometry within L^2.

variance analysis presented here will play an essential role in the development of an algorithmic solution capable of achieving an adaptive estimate close to the ideal one.

2.3.1 Bias and variance

Let the observations be of the form

$$z = y + \sigma\eta, \tag{2.3}$$

where η is an identically distributed Gaussian white noise, $\eta(\cdot) \sim \mathcal{N}(0,1)$. Overall, the original true signal y suffers degradation from an additive noise term $\sigma\eta$ with variance σ^2.

The bias and variance, i.e. the deterministic and stochastic errors, of the *LPA* estimate $\hat{y}_h(x) = (z \circledast g_h)(x)$ are, respectively,

$$\begin{aligned} bias\{\hat{y}_h(x)\} &= m_{\hat{y}_h(x)} = E\{y(x) - \hat{y}_h(x)\} = y(x) - E\{\hat{y}_h(x)\} = \\ &= y(x) - (E\{z\} \circledast g_h)(x) = y(x) - (y \circledast g_h)(x), \end{aligned}$$

and

$$\begin{aligned} &var\{\hat{y}_h(x)\} = \sigma^2_{\hat{y}_h(x)} = E\left\{\left(E\{\hat{y}_h(x)\} - \hat{y}_h(x)\right)^2\right\} = \\ &= E\left\{\left(((y-z) \circledast g_h)(x)\right)^2\right\} = \left(\sigma_z^2 \circledast g_h^2\right)(x) = \sigma^2 \int g_h^2(v)\,dv = \sigma^2 \|g_h\|_2^2. \end{aligned} \tag{2.4}$$

The above relation can also be easily obtained recalling that "the variance of the sum of independent variables is the sum of their variances".

The pointwise mean squared error (MSE), or (quadratic) risk, can be decomposed in the sum of the squared bias and the variance[3],

$$l_{\hat{y}_h(x)} = E\left\{\left(y(x) - \hat{y}_h(x)\right)^2\right\} = m^2_{\hat{y}_h(x)} + \sigma^2_{\hat{y}_h(x)}.$$

Our goal is to determine h in such a way that $l_{\hat{y}_h(x)}$ is minimized.

[3] The bias-variance decomposition of the MSE can be obtained from the following elementary manipulation:

$$\begin{aligned} E\left\{(y-\hat{y})^2\right\} &= E\left\{(\hat{y} - E\{\hat{y}\} + E\{\hat{y}\} - y)^2\right\} = \\ &= E\left\{(\hat{y} - E\{\hat{y}\})^2\right\} + 2E\{(\hat{y} - E\{\hat{y}\})(E\{\hat{y}\} - y)\} + E\left\{(E\{\hat{y}\} - y)^2\right\}. \end{aligned}$$

The middle term has a factor, $E\{\hat{y} - E\{\hat{y}\}\} = E\{\hat{y}\} - E\{\hat{y}\}$, which is equal to zero, thus the mean squared error can be written as

$$\begin{aligned} E\left\{(y-\hat{y})^2\right\} &= E\left\{(\hat{y} - E\{\hat{y}\})^2\right\} + E\left\{(E\{\hat{y}\} - y)^2\right\} = \\ &= E\left\{(\hat{y} - E\{\hat{y}\})^2\right\} + (E\{\hat{y}\} - y)^2 = var\{\hat{y}\} + bias^2\{\hat{y}\}. \end{aligned}$$

2.3.2 Asymptotic error analysis

Let x be a fixed estimation point. By equation (2.1) we can rewrite the expression for the variance as an explicit function of the scale parameter,

$$\begin{aligned}\sigma^2_{\hat{y}_h(x)} &= \sigma^2 \|g_h\|_2^2 = \sigma^2 \left\|h^{-d} g_{\mathcal{M}}(\cdot/h)\right\|_2^2 = \\ &= \sigma^2 h^{-2d} \int g^2_{\mathcal{M}}(v/h)\, dv = \sigma^2 h^{-d} \int g^2_{\mathcal{M}}(v)\, dv = \sigma^2 h^{-d} \|g_{\mathcal{M}}\|_2^2 .\end{aligned}$$

Similarly, for the bias, by exploiting a similar change of variable, we obtain

$$m_{\hat{y}_h(x)} = y(x) - \int y(x-v)\, h^{-d} g_{\mathcal{M}}(v/h)\, dv = y(x) - \int y(x-hv)\, g_{\mathcal{M}}(v)\, dv.$$

Let us consider a Taylor-type expansion at x of the function y,

$$y(x-w) = y(x) + \sum_{o \in O_m \setminus \{0\}} \frac{(-1)^{|o|}}{o!} (D^o y)(x)\, w^o + R(w),$$

where the remainder term $R(w) = \sum_{o \notin O_m} \mathcal{O}(w^o)$ and $\mathcal{O}(w^o)$ denotes a function such that is asymptotic to w^o as $w \to 0$.

Because of the perfect-reconstruction property for polynomials whose monomes have orders in O_m, we conclude that the bias is made only from the contribution of this remainder:

$$m_{\hat{y}_h(x)} = y(x) - \int y(x-hv)\, g_{\mathcal{M}}(v)\, dv = \int R(hv)\, g_{\mathcal{M}}(v)\, dv.$$

There are many possible representation of the remainder R. However, the standard choice is to express it in Lagrange form. It means that $|R(w)| \leq \sum L_o |w^o|$ where L_o is some uniform bound on the o-th partial derivative of y. Depending on the order m of the *LPA*, the dimension d, of course, actual the smoothness of y, various accurate bounds the remainder can be obtained (e.g. [31],[15],[43],[44]).

Nevertheless, for all the coming considerations, it is enough to consider the following generic upper bound on the modulus of the asymptotic bias of the *LPA* estimate,

$$\bar{m}_{\hat{y}_h(x)} = r h^\alpha \|g_{\mathcal{M}}\|_1 = a h^\alpha, \tag{2.5}$$

and its variance

$$\sigma^2_{\hat{y}_h(x)} = \sigma^2 h^{-d} \|g_{\mathcal{M}}\|_2^2 = b^2 h^{-2\beta}. \tag{2.6}$$

The upper bound of the asymptotic risk is

$$\bar{l}_{\hat{y}_h(x)} = \bar{m}^2_{\hat{y}_h(x)} + \sigma^2_{\hat{y}_h(x)} = a^2 h^{2\alpha} + b^2 h^{-2\beta}.$$

It is a concave function, as shown in Figure 2.2.

Quite obviously b and β depend only on the kernels, since the variance of the estimate is not affected by the estimated function. On the other hand, the uniform bound L can affect only a. The coefficient α is influenced by the order m of the *LPA* and by the polynomial components of the remainder.

Similar asymptotical forms can be obtained also for the derivative estimates.

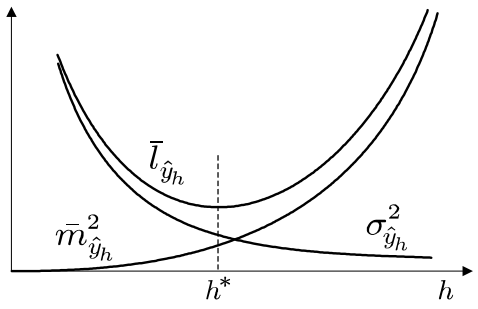


Figure 2.2: Asymptotic bias-variance trade-off: the variance $\sigma^2_{\hat{y}_h(x)}$, the upper bound for the squared bias $\bar{m}^2_{\hat{y}_h(x)}$ and the upper bound for the mean squared error $\bar{l}_{\hat{y}_h(x)}$. The ideal scale h^* minimizes $\bar{l}_{\hat{y}_h(x)}$.

2.3.3 Ideal scale

The pointwise *ideal scale* $h^* = h^*(x)$ is defined as the minimizer of $\bar{l}_{\hat{y}_h(x)}$,

$$h^* = \operatorname*{argmin}_h \bar{l}_{\hat{y}_h(x)},$$

and can be found by solving

$$\partial_h \bar{l}_{\hat{y}_h(x)} = 0.$$

It gives

$$\partial_h \bar{l}_{\hat{y}_h(x)} = 2a^2\alpha h^{2\alpha-1} - 2b^2\beta h^{-2\beta-1} = 0,$$

and the ideal scale h^* as

$$h^* = \left(\frac{\beta b^2}{\alpha a^2}\right)^{\frac{1}{2\alpha+2\beta}}. \tag{2.7}$$

Let us use this ideal scale into the bias and variance expressions (2.5) and (2.6). We obtain, respectively,

$$\begin{aligned} \bar{m}^2_{\hat{y}_{h^*}(x)} &= a^2\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{2\alpha}{2\alpha+2\beta}} = a^2\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{\alpha}{\alpha+\beta}}, \\ \sigma^2_{\hat{y}_{h^*}(x)} &= b^2\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{-2\beta}{2\alpha+2\beta}} = b^2\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{-\beta}{\alpha+\beta}}. \end{aligned}$$

Observe that the ratio between then upper bound of the squared bias and the variance at the ideal scale h^*,

$$\frac{\bar{m}^2_{\hat{y}_{h^*}(x)}}{\sigma^2_{\hat{y}_{h^*}(x)}} = a^2\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{\alpha}{\alpha+\beta}} b^{-2}\left(\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{\beta}{\alpha+\beta}} = a^2b^{-2}\frac{\beta b^2}{\alpha a^2} = \frac{\beta}{\alpha} = \gamma^2, \tag{2.8}$$

does not depend on a. It means that it does not depend on the local behaviour of the function at x.

The upper bound of the risk at the ideal scale h^*, so-called *ideal risk*, has the following form:

$$\bar{l}_{\hat{y}_{h^*}(x)} = \sigma^2_{\hat{y}_{h^*}(x)}\left(1+\gamma^2\right). \tag{2.9}$$

Since the ratio $\bar{m}^2_{\hat{y}_h(x)}/\sigma^2_{\hat{y}_h(x)}$ is a monotonically increasing function of h, we have that

$$\bar{m}^2_{\hat{y}_h(x)} \begin{cases} < \gamma^2 \sigma^2_{\hat{y}_h(x)} & \forall h < h^* \\ > \gamma^2 \sigma^2_{\hat{y}_h(x)} & \forall h > h^* \end{cases}, \tag{2.10}$$

These inequalities, which turn into an equality only at $h = h^*$, can be then used to test the hypothesis $h \lessgtr h^*$.

2.4 Intersection of Confidence Intervals (*ICI*) rule

The *ICI* rule [28, 42] is a practical method that, based on the analysis from the previous section, selects an adaptive scale $h^+(x)$ whose corresponding estimate $\hat{y}_{h^+(x)}$ is close to the ideal $\hat{y}_{h^*(x)}$. In what follows, we sketch a proof of the general validity of the method. We refer the reader to [28] or [12] for a more rigourous proof and for discussions about the convergence rate of this adaptive estimate.

2.4.1 The idea

Again, let x be a fixed estimation point and $\hat{y}_h(x)$ an *LPA* estimate at x. The total estimation error $|y(x) - \hat{y}_h(x)|$ can be bounded by the sum of the moduli of the bias error $m_{\hat{y}_h(x)}$ and the random error $r_{\hat{y}_h(x)} = E\{\hat{y}_h(x)\} - \hat{y}_h(x)$,

$$|y(x) - \hat{y}_h(x)| \le \left|m_{\hat{y}_h(x)}\right| + \left|r_{\hat{y}_h(x)}\right|.$$

The random error $r_{\hat{y}_h(x)}$ is a normal-distributed random variable with variance $\sigma^2_{\hat{y}_h(x)}$ and zero mean, $r_{\hat{y}_h(x)} \sim \mathcal{N}\left(0, \sigma^2_{\hat{y}_h(x)}\right)$. The following inequality holds with probability $p = 1 - \lambda$,

$$\left|r_{\hat{y}_h(x)}\right| \le \chi_{1-\lambda/2}\sigma_{\hat{y}_h(x)},$$

where $\chi_{1-\lambda/2}$ is a $(1-\lambda/2)$-th quantile of the normal distribution $\mathcal{N}(0,1)$. Hence, with same probability p,

$$|y(x) - \hat{y}_h(x)| \le \left|m_{\hat{y}_h(x)}\right| + \chi_{1-\lambda/2}\sigma_{\hat{y}_h(x)}.$$

From the inequalities (2.10) we obtain, for $h \le h^*(x)$,

$$|y(x) - \hat{y}_h(x)| \le (\gamma + \chi_{1-\lambda/2})\sigma_{\hat{y}_h(x)} = \Gamma\sigma_{\hat{y}_h(x)}, \quad \Gamma = (\gamma + \chi_{1-\lambda/2}). \tag{2.11}$$

Equivalently, we can express the above inequality as

$$\hat{y}_h(x) - \Gamma\sigma_{\hat{y}_h(x)} \;\le\; y(x) \;\le\; \hat{y}_h(x) + \Gamma\sigma_{\hat{y}_h(x)}, \qquad h \le h^*(x),$$

determining a confidence interval $\mathcal{D}(h)$,

$$\mathcal{D}(h) = \left[\hat{y}_h(x) - \Gamma\sigma_{\hat{y}_h(x)}, \hat{y}_h(x) + \Gamma\sigma_{\hat{y}_h(x)}\right],$$

for the estimate $\hat{y}_h(x)$: for $h \le h^*$, with probability p, we have $y(x) \in \mathcal{D}(h)$. According to (2.6), the width of the confidence intervals $\mathcal{D}(h)$ is a monotonically

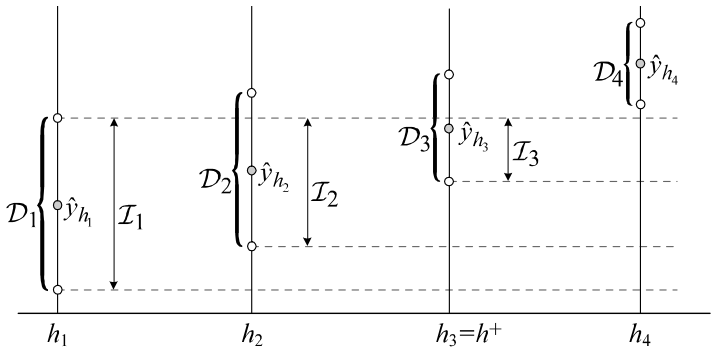


Figure 2.3: The Intersection of Confidence Intervals (*ICI*) rule.

decreasing function of h. We may say that the confidence intervals "shrink" as h increases.

Let $H = \{h_1, \ldots, h_J\}$, $h_1 < \cdots < h_J$, be an increasing set of scales and the corresponding estimates $\left\{\hat{y}_{h_j}(x)\right\}_{j=1}^{J}$. Each one of these estimates is a normal-distributed random variable with variance $\sigma^2_{\hat{y}_{h_j}(x)}$. With some probability p', all confidence intervals $\mathcal{D}(h_j)$, $h_j \leq h^*(x)$ have a point in common, namely, $y(x)$.

Let j^+ be the largest of those j for which all $\mathcal{D}(h_i)$ with $i \leq j$ have a point in common. Observe that $h_{j^+} \geq h^{*-} \triangleq \max\{h_j : h_j \leq h^*(x)\}$, i.e. all $\mathcal{D}(h_j)$ with $h_j \leq h^{*-}$ have non-empty intersection. This condition, together with the shrinking of the confidence intervals, ensures that the estimate $\hat{y}_{h^+}(x)$ is within a certain range from the true signal $y(x)$.

Indeed, $y(x) \in \bigcap_{h_j \leq h^{*-}} \mathcal{D}(h_j)$, hence $|y(x) - \hat{y}_{h^{*-}}| \leq \Gamma\sigma_{\hat{y}_{h^{*-}}(x)}$. Similarly, since $\mathcal{D}(h^{*-}) \cap \mathcal{D}(h^+) \neq \varnothing$, $|\hat{y}_{h^{*-}} - \hat{y}_{h^+}| \leq \Gamma\sigma_{\hat{y}_{h^{*-}}(x)} + \Gamma\sigma_{\hat{y}_{h^+}(x)}$. Combining these, we conclude that

$$|y(x) - \hat{y}_{h^+}| \leq 2\Gamma\sigma_{\hat{y}_{h^{*-}}(x)} + \Gamma\sigma_{\hat{y}_{h^+}(x)} \leq 3\Gamma\sigma_{\hat{y}_{h^{*-}}(x)}. \tag{2.12}$$

Provided that the set of scales H is sufficiently rich, one has $h^{*-} \simeq h^*$, and thus $\sigma_{\hat{y}_{h^{*-}}(x)} \simeq \sigma_{\hat{y}_{h^*}(x)}$. It follows that the error of the adaptive estimate $\hat{y}_{h^+}$ is at most 3Γ times the ideal deviation $\sigma_{\hat{y}_{h^*}(x)}$.

2.4.2 *ICI* adaptive-scale selection rule

The practical *ICI* adaptive-scale selection method ([42],[43],[28]) follows directly from the above considerations.

Given a set of varying scale kernel estimates $\left\{\hat{y}_{h_j}(x)\right\}_{j=1}^{J}$ such that $\sigma_{\hat{y}_{h_1}} > \cdots > \sigma_{\hat{y}_{h_J}}$, we determine a sequence of confidence intervals

$$\mathcal{D}_j = \left[\hat{y}_{h_j}(x) - \Gamma\sigma_{\hat{y}_{h_j}}, \hat{y}_{h_j}(x) + \Gamma\sigma_{\hat{y}_{h_j}}\right]$$

where $\Gamma > 0$ is a threshold parameter. The *ICI* rule can be stated as follows:

Consider the intersection of confidence intervals $\mathcal{I}_j = \bigcap_{i=1}^{j} \mathcal{D}_i$ and let j^+ be the largest of the indexes j for which $\mathcal{I}_j$ is non-empty, $\mathcal{I}_{j^+} \neq \varnothing$ and $\mathcal{I}_{j^+ + 1} = \varnothing$.

Then the adaptive scale h^+ *is defined as* $h^+ = h_{j^+}$ *and the adaptive-scale kernel estimate is therefore* $\hat{y}_{h^+}(x)$.

Roughly speaking *ICI* selects the coarsest scale estimate that is statistically compatible with all finer scales. In practice this means that adaptively, for every pixel, *ICI* allows the maximum degree of smoothing, stopping before oversmoothing begins.

The variance $\sigma^2_{\hat{y}_{h_j}}$ of the estimate $\hat{y}_{h_j}$ can be calculated from the kernel g_{h_j} by formula (2.4). Therefore, the *ICI* rule can be implemented in a straightforward manner for the selection of the the adaptive-scale kernel estimate $\hat{y}_{h^+}(x)$ given a set *LPA* kernels $\left\{g_{h_j}\right\}_{j=1}^{J}$ and an estimate $\hat{\sigma}^2$ of the variance σ^2 of the additive noise term $\sigma^2\eta$ in (2.3)[4].

2.4.3 *ICI* algorithm pseudo-code

Table (2.1) shows a pseudo-code of the *ICI* algorithm. It is assumed that a set of estimates $\left\{\hat{y}_{h_j}\right\}_{j=1}^{J}$ are given together with their standard deviations $\{\sigma_{\hat{y}_{h_j}}\}_{j=1}^{J}$. The estimates and the variances can be matrices, where each entry in the matrix corresponds, respectively, to a pointwise estimate $\hat{y}_{h_j}(x)$ and pointwise variance $\sigma_{\hat{y}_{h_j}(x)}$. In this pseudo-code we use the following formalism: the equality symbol $=$ is used to update the value of a variable[5], the "identically equal" symbol $\equiv$ between a matrix and a constant indicates that all entries of the matrix are defined to be identically equal to the constant, the symbol $\geq$ stands for a relational operator that returns 1 or 0 if the tested inequality is verified or not[6], NOT denotes the logical negation (i.e. $\text{NOT}(0) = 1$ and $\text{NOT}(1) = 0$), the products and divisions between matrices are computed as array-operations (i.e. in pointwise manner), and comments to the code are written after a slash mark, /.

[4] The estimate of σ can be obtained using standard techniques. For example, a widely used and simple-to-implement method is based on the *median of the absolute deviation* (MAD)[30], and gives a robust estimate of the standard deviation of the additive white Gaussian noise as

$$\hat{\sigma} = \frac{median(|d|)}{0.6745}$$

where $d = (d_i)$ is a vector formed by normalized differences between adjacent samples of the noisy observations z,

$$d_i = \frac{z(x_i) - z(x_{i+1})}{\sqrt{2}}.$$

This technique can be used also in conjuction with an orthonormal transform (as it preserves the statistical characterics of the additive white noise). A frequent choice ([11],[66]), is to apply the above median estimator on some wavelet detail (finest scale) coefficients $\langle z, \psi_j \rangle$ rather than on the differences d. Since most of the signal is usually compacted into wavelet approximation (coarser scales) coefficients, this "wavelet-domain MAD" reduces the impact of the signal's features on the estimation, thus avoiding the overestimation of σ. The "classic" MAD can be interpreted as a particular case of the wavelet-domain MAD, since for the Haar wavelet $\left(\langle z, \psi_j \rangle\right)_j = d$.

[5] For example, an apparently non-sense equation such as $a = a + 1$, simply means that the new value for the variable a is equal to its old value plus one: $a_{n+1} = a_n + 1$.

[6] Therefore, the T in the pseudo-code is a matrix composed by one and zeros.

$h^+ \equiv h_1$ $\hat{y}_{h^+} = \hat{y}_{h_1}$	initialization of adaptive scale and of corresponding estimate and variance
$U = \hat{y}_{h_1} + \Gamma\sigma_{\hat{y}_{h_1}}$ $L = \hat{y}_{h_1} - \Gamma\sigma_{\hat{y}_{h_1}}$	initialization of upper and lower bounds of intersection
`for` $j = 2, \ldots, J$	loop on j (scale index)
$U = \min\{U, \hat{y}_{h_j} + \Gamma\sigma_{\hat{y}_{h_j}}\}$ $L = \max\{L, \hat{y}_{h_j} - \Gamma\sigma_{\hat{y}_{h_j}}\}$	update bounds of intersection
$T = U \geq L$	test for non-empty intersection
$h^+ = h_j T + h^+_{\theta_k}\mathrm{NOT}(T)$ $\hat{y}_{h^+} = \hat{y}_{h_j} T + \hat{y}_{h^+}\mathrm{NOT}(T)$	update adaptive scale and adaptive-scale estimate
`end`	end loop on j (scale index)

Table 2.1: Pseudo-code of the *ICI* adaptive-scale selection algorithm.

2.4.4 Choice of Γ

Formula (2.12) reveals the role of the threshold parameter Γ in ensuring the fidelity of the adaptive estimate $\hat{y}_{h^+}$. At first glance, it may seem that Γ should be chosen as small as possible, so to minimize the risk of the adaptive-scale estimate. However, since the confidence intervals are a probalistic device, a too small Γ, i.e. a too small $\chi_{1-\lambda/2}$, makes the probabilities p and p' too small for the derived analysis to have any practical significance, and thus (2.12) may fail to be verified (as y may not belong to the interesection), resulting in a larger-than-predicted error of the adaptive estimate. This considerations suggest, the following, say, rule-of-thumb: "choose Γ small, so to increase the sensitivity, but not too small to avoid too frequent false alarms, i.e. empty intersections due to the randomness of the data". The adaptive h^+ is a monotonically increasing function of Γ. Therefore, qualitatively speaking, a smaller Γ produces smaller adaptive scales, and thus less smoothing, whereas a larger Γ corresponds to large adaptive scales, and thus more smoothing.

By the definition (2.11), the value of Γ depends on the quantile $\chi_{1-\lambda/2}$, as well as on the ideal ratio γ (2.8). Since γ varies together with the *LPA* design, we conclude that the choice of Γ is influenced by this design.

First we note that a rather detailed study of the convergence rate of the adaptive estimate is provided in [28] and [12]. There it is shown that, with respect to an asymptotic randomness of the noise of the order $\sigma^2 = \mathcal{O}(1/n)$, the best possible convergence rate is achieved for $\Gamma = \mathcal{O}(\sqrt{\ln n})$. This sort of asymptotical convergence-rate analysis, however, tells very little on what should be done in practice with Γ, as the same rate is achieved up to a constant factor.

Some experimental study and heuristical speculations on the performance of the *ICI* are presented in [89], showing that the *ICI* algorithm is quite stable with respect to variations of Γ. A similar conclusion was drawn already in [48]. It is shown in [42] and [43], that the statistical common-practice choices for the confidence intervals, with $\lambda = 0.05$ or $\lambda = 0.01$, result in quite large values of Γ. It also shown that these choices are not really adequate for signal-processing applications, and, in the same publications, an approach for the automatic selection of Γ based on cross-validation is proposed. Nevertheless, none of the

aforementioned theoretical, heuristical, or statistical methods for the choice of this threshold parameter can be considered conclusive, and experimental simulation is still the most effective and accurate methodology for the choice of Γ.

Depending on kernel design parameters such as the order of the *LPA* and the smoothness of w, the typical choices for the value of Γ fall between 0.5 and 2. Thanks to the aforementioned stability of the *ICI* algorithm with respect to Γ, only little adjustment in its value is needed for the algorithm optimization. After this adjustment, a fixed Γ can be used.

2.4.5 Examples with symmetric windows

Figure 2.4 illustrates the effect of the *ICI* algorithm, and the impact of the threshold parameter Γ on the choice of the adaptive scales. The observation z consists of a noisy version of the *Cameraman* image (degradated by the addition of Gaussian white noise with standard deviation equal to 0.1). The kernel family $\left\{g_{h_j}\right\}_{j=1}^{J}$ is composed by 35 zero-order *LPA* kernels. They are designed using as window w_{h_j} the characteristic function of a disk of radius h_j, $h_j = 1, \ldots, 35$ pixels, centered in the origin. The smallest-scale kernel is a Dirac delta.

One can see in the Figure that, even with very large values of Γ, the sensibility of the algorithm with respect to the edges and other structures in the images is quite high. This sensibility is shown by the smaller values of h^+ that are found in the vicinity of these structures[7]. Qualitatively, adaptive-scale estimates with a small Γ are still noisy, whereas a too large value of Γ produces estimates that are over-smooth. The impact of the Γ parameter on the overall *RMSE* is depicted in Figure 2.5, not only for the *Cameraman* image, but also for the *Lena* and *Boats* images. The plots show that for the three images the best found values of Γ are all close to 2, and that variations of ± 0.2 around these best found values do not affect the objective quality of reconstruction. Therefore, with this family of kernels, $\Gamma = 2$ can be used for all the images achieving a performance very close to the one achieved with an "oracle" Γ.

Finally, let us observe that the adaptive scale h^+ decreases when approaching the boundary of the image. This is an automatic feature of the *ICI* when proper[8] boundary conditions are imposed. In this example, the maximum possible value of $h^+(x)$ is equal to the distance of x from the boundary of the image.

[7] The white line (larger scales) along the edges is an unavoidable "defect" of kernels that are symmetric about the origin. It arises from the fact that the symmetric averages centered near the discontinuity are quite stable with respect to the scale. Essentially, it is a behaviour similar to that of the Fourier series on jump-discontinuites (they converge to the mid-point of the jump, which is equal to the limit of integral averages on a symmetric neighborhood collapsing around the discontinuity, so-called Lebesgue-limit). It is shown in the coming chapters that this "unpleasant" feature is not encountered when asymmetric kernels are used.

[8] See the footnote on the boundary conditions on page 11.

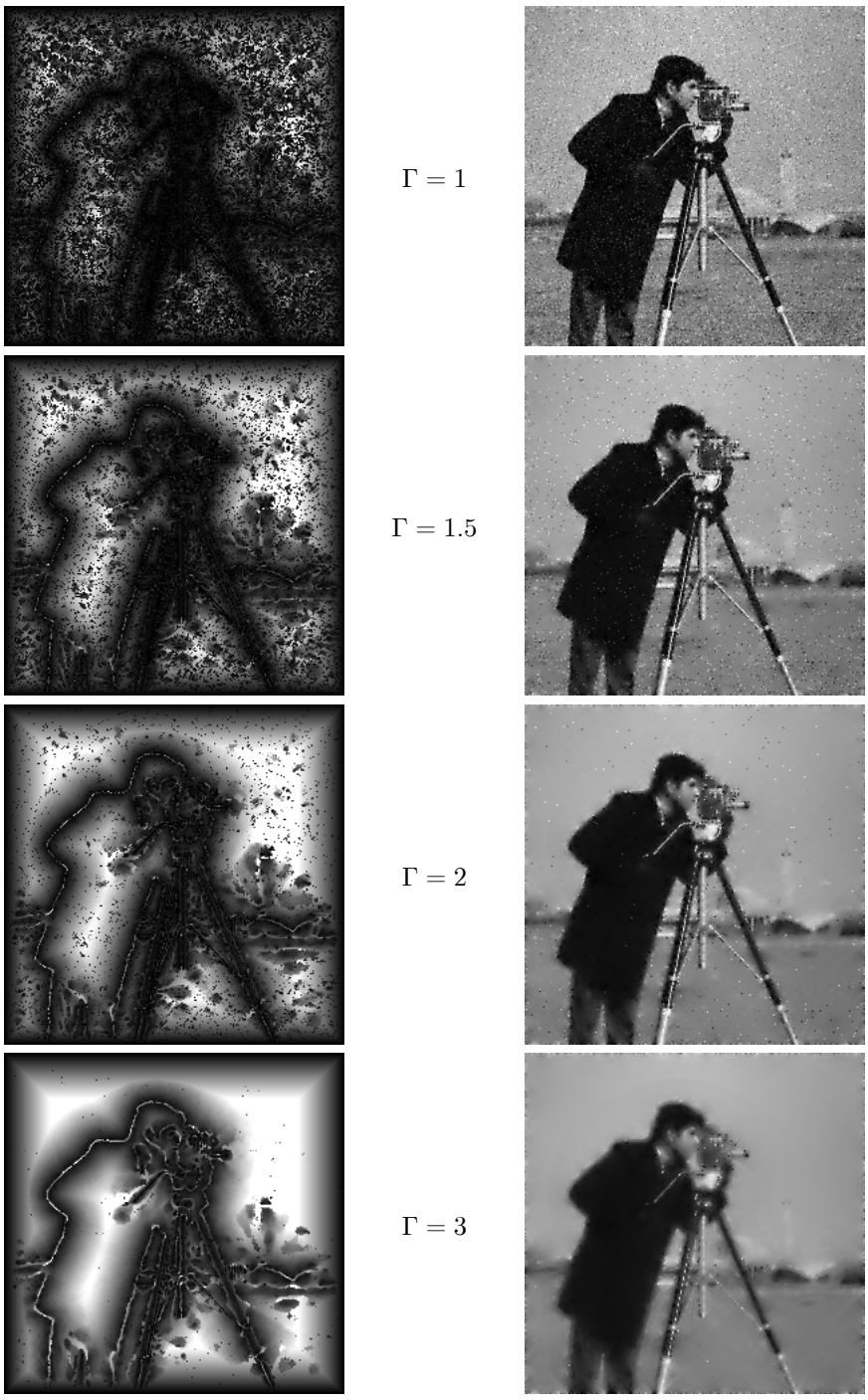


Figure 2.4: Adaptive scales (left) and adaptive-scale estimates (right) obtained for a wide range of values of the *ICI* threshold parameter Γ. From top to bottom, $\Gamma = 1, \Gamma = 1.5, \Gamma = 2$, and $\Gamma = 3$. The adaptive scales are represented using a darker shade of gray for the smaller scales, black being the smallest scale (which corresponds to a Dirac-delta estimate), and white being the maximum scale (corresponding to a kernel whose support is a disc of radius 35 pixels).

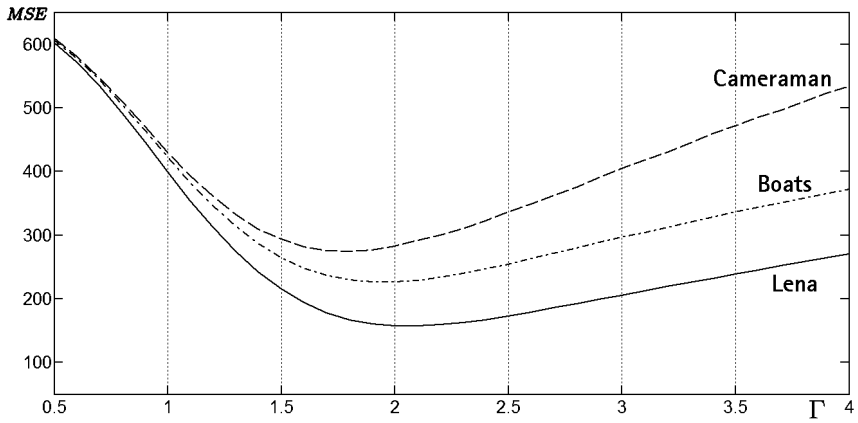


Figure 2.5: "*MSE* vs. Γ" plots for three different images ($\sigma = 0.1$). The same kernels used for the experiment shown in Figure 2.4 are used.

Chapter 3

Directional *LPA*

3.1 Motivation

Points, lines, edges, textures are present in all images. They are locally defined by position, orientation and scale. Often being of small size these specific features encode a great proportion of information contained in images. To deal with these features oriented/directional filters are used in many vision and image processing tasks, such as edge detection, texture and motion analysis, etc.

The key question is, how to design a kernel for a specified direction. A good initial idea arises from the definition of the *directional derivative* ∂_θ for the direction defined by the angle θ,

$$\partial_\theta y(x) = \lim_{\rho \to 0} \left(y(x_1 + \rho \cos\theta,\ x_2 + \rho \sin\theta) - y(x_1, x_2) \right) / \rho, \tag{3.1}$$

and more specifically, from that of the *right-hand*[1] *directional derivative* $\partial_{+\theta}$,

$$\partial_{+\theta} y(x) = \lim_{\rho \to 0^+} \left(y(x_1 + \rho \cos\theta,\ x_2 + \rho \sin\theta) - y(x_1, x_2) \right) / \rho. \tag{3.2}$$

Whenever y is a differentiable function, elementary calculations give the well known result

$$\partial_{+\theta} y(x) = \partial_\theta y(x) = \cos\theta \cdot \partial_{x_1} y(x) + \sin\theta \cdot \partial_{x_2} y(x). \tag{3.3}$$

Thus, in order to find the derivative for any direction θ it suffices to estimate the two derivatives on x_1 and x_2 only. This concept has been exploited and generalized by the so-called steerable filters [25].

Although continuous models of the discrete image intensity are widely used in image processing, estimates such as (3.3) are too rough in order to be useful for those applications where the sharpness and details are of first priority. For discrete images lacking global differentiability or continuity the only reliable way to obtain an accurate directional anisotropic information is to calculate variations of y in the desired direction θ and, say, to estimate the directional derivative by the finite difference counterpart of $\partial_{+\theta} y(x)$. In more general terms, this means that the estimation or image analysis should be based on directional

[1] The *left-hand directional derivative* $\partial_{-\theta}$ is defined similarly, by replacing $\lim_{\rho \to 0^+}$ with $\lim_{\rho \to 0^-}$ in (3.2).

kernels, templates or atoms which are quite narrow and concentrated in desirable directions. Since points, lines, edges and textures can exist at all possible positions, orientations and scales, one would like to use families of filters that can be tuned to all orientations, scales and positions.

Recent development shows an impressive success of methods for this sort of directional image/multivariable signal processing. In particular, narrow multidirectional items are the building blocks of the new ridgelet and curvelet transforms [90].

Conventionally, the *LPA* estimation kernels have simple-form supports (square, discs, etc.), symmetric with respect to the origin and/or the coordinate axes. In order to design narrow, directional kernels suitable for the analysis of oriented features a "directional" version of the standard *LPA* is proposed.

3.2 Directional *LPA*: a general definition

The directional version of the *LPA* method is simply an *LPA* where the basis of polynomials is given in a rotated coordinate system $\left\{u_i^\theta\right\}_{i=1}^d$ and where the windowing function, $w = w_\theta$, has a characteristic orientation along some direction θ. We have $u^\theta = \mathbf{U}_\theta v$, and $\mathbf{U}_\theta$ is a rotation matrix. For example, when $d = 2$, $\left(u_1^\theta, u_2^\theta\right)^T = (v_1 \cos\theta + v_2 \sin\theta, v_2 \cos\theta - v_1 \sin\theta) = \mathbf{U}_\theta v$.

For the sake of clarity, we drop the θ symbol from the notation u^θ when also another exponent is present, and assuming implicitly that $u = u^\theta$.

The polynomial Taylor basis in the rotated system is given as

$$\{\phi_n\}_{n=1}^N = \left\{\frac{u^o}{o!}\right\}_{o\in O_m}. \tag{3.4}$$

According to (3.4), and analogously to (1.13), the set $\mathcal{M}_\theta$ is expressed as

$$\mathcal{M}_\theta = \left\{\varphi : \varphi(v) = \sum_{o\in O_m} c_o u^o,\ c_o \in \mathbb{R}\right\}. \tag{3.5}$$

To introduce a space structure on $\mathcal{M}_\theta$, only the window function w_θ needs to be defined. The natural choice is to have the window w_θ elongated along the first axis u_1^θ of the rotated coordinate system. Typically, w_θ is obtained by rotating a "basic" window w_0 (which is elongated along v_1) through an angle θ, $w_\theta(v) = w_0(\mathbf{U}_\theta v)$. When also a scale parameter h is exploited, the resulting windows and kernels are denoted as $w_{h,\theta}$, $g_{h,\theta}$, respectively.

Just as for the standard *LPA*, also directional-*LPA* kernels satisfy perfect-reconstruction and moment conditions, with the only difference that these conditions hold with respect to the rotated coordinate system. In the discrete domain, the fact that the window is elongated along u_1 enables the design of kernels with $m_1 \gg m_i$, $i \neq 1$.

Derivative estimation kernels are defined exactly in the same way as for the standard *LPA* and, since the Taylor basis is expressed with respect to the coordinates u^θ, the kernels $g_{h,\theta}^{(o)}$ estimate the directional derivatives $\frac{\partial^{|o|}}{\partial u^o} = \frac{\partial^{o_1}}{\partial u_1^{o_1}} \cdots \frac{\partial^{o_d}}{\partial u_d^{o_d}}$.

In the continuous domain it is easy to realize that the directional-*LPA* kernel $g_{h,\theta}$ is simply a rotated copy of the standard *LPA* kernel $g_{h,0}$ corresponding to the basic window w_0,

$$g_{h,\theta}(v) = g_{h,0}(\mathbf{U}_\theta v).$$

We call $g_{h,0}$ *the basic kernel*. The above formula greatly simplifies the construction of narrow well-oriented local polynomial approximation kernels. Regrettably, this simplification has only theoretical significance, since it cannot be exploited successfully in the discrete domain.

3.3 Discrete directional-*LPA* kernel construction

In the discrete domain the rotation itself is not a trivial operation because the rotated rectangular data grid u does not properly match with its non-rotated version v. For this very reason, traditional rotation techniques are based on interpolation methods.

The construction of a directional-*LPA* kernel in the discrete domain needs to comprise two independent steps. First, the design of the oriented window $w_{h,\theta}$ is performed. Its support $\operatorname{supp} w_{h,\theta}$ is finite, non-symmetric, elongated and well-oriented along the direction θ. A good example of such a support can be a conical sector of length h. Second, the standard *LPA* procedure is applied using the polynomials in the rotated variables u^θ with the weights $w_{h,\theta}$.

We remark again that the simpler approach where a basic *LPA* kernel is rotated along the desired directions, although reasonable (as seen in the continuous domain), fails in the discrete domain. That is because the interpolation, which is necessary for the rotation, does not in general preserve the normalization (1.18) and the vanishing moment conditions (1.19), essential requirements for the accurate polynomial reproducing properties of *LPA* kernels. It is worth to point out that the initial motivitation in the development of the directional *LPA* is the design of narrow and sharp oriented kernels. It is a well-known fact in image processing that interpolation-based rotation (and more generally any traditional interpolation scheme) is not suitable for preserving the sharpness of the transformed data.

For example, the nearest-neighbor rotation is quite effective for preserving the sharpness of the rotated window function, but on the other hand it does not preserve not even the first vanishing moment condition $\int g_{h,\theta} = 1$. Contrary to this, rotation based on linear interpolation preserve this vanishing moment condition, but is not able to preserve the sharpness (i.e. the discontinuities at the support's boundary) of the kernel. Decoupling the directional kernel design in two independent steps, enables to achieve accurate moment conditions (with respect to the rotated coordinates) without imposing any restrictions on the rotated window function. In fact, any rotation method can be used to obtain the window $w_{h,\theta}$ from the basic window w_h.

3.4 Peculiarities of the directional-*LPA* kernel design

This technique allows to design estimators for smoothing and differentiation that are important on their own, and that can be used in many applications.

Indeed, they have a number of valuable benefits:

- Unlike many other transforms which start from the continuous domain and then discretized, this technique works directly in the multidimensional discrete domain;
- The designed kernels are truly multivariable, non-separable and directional with arbitrary orientation, width and length;
- The smoothing and the corresponding differentiating directional kernels can be designed;
- The kernel support can be flexibly shaped to any desirable geometry in order to capture geometrical structural and pictorial information. In this way a special design can be done for complex-form objects and specific applications;
- These kernels are by definition asymmetric, allowing efficient edge adaptation. Traditional symmetric or nearly-symmetric supports tend to produce either so-called ringing artifacts or oversmoothing in the vicinity of the edges.

The use of directional-*LPA* kernels for the estimation of directional (one-handed) derivatives and the gradient is the subject of Chapter 7.

3.5 Some examples of directional-*LPA* kernels

In Figure 3.1, Figure 3.2, and Figure 3.3, some discrete *LPA* kernels are shown, together with the absolute value of their Fourier transform for three different values of θ: $\theta_1 = 0$, $\theta_2 = \pi/4$, and $\theta_3 = \pi/2$. The kernels are obtained using the described technique using rotated copies of one basic window w equal to the characteristic function of a conical sector. Observe that these kernels are asymmetrical and that the origin is an extreme point of the window's support. More examples of directional-*LPA* kernels are given in the Figures 9.13-9.14 (page 140) and Figure 9.2 (page 127) from the coming chapters.

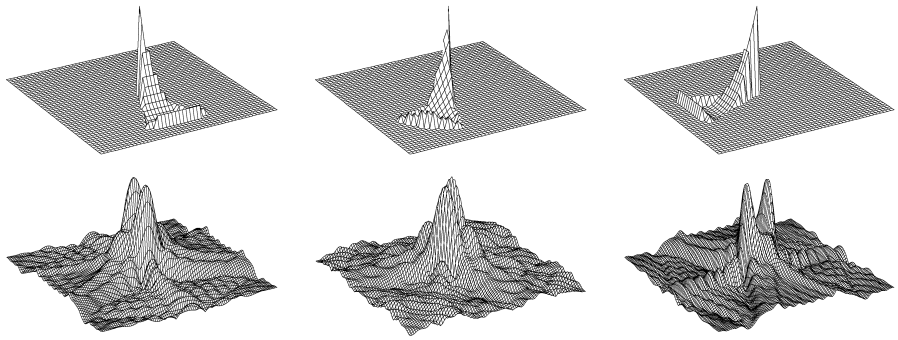


Figure 3.1: Function estimation directional *LPA* kernels, $o = (0,0)$, obtained for $m = (2,1)$ using the characteristic function of a conical sector as the window function w_θ. Kernels are shown for three different direction θ; the absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure.

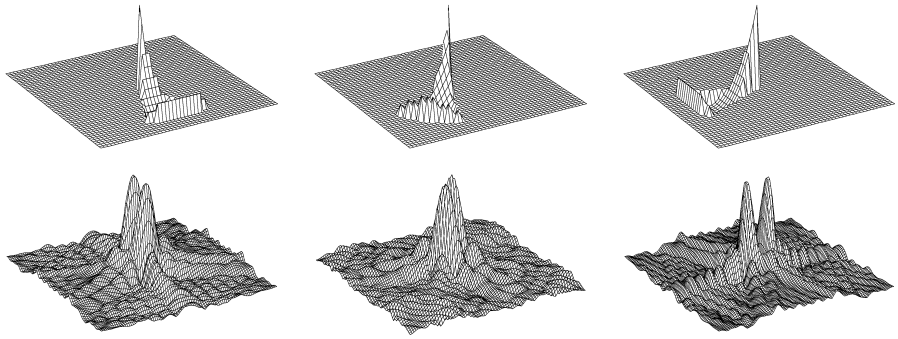


Figure 3.2: Partial-derivative estimation directional *LPA* kernels, $o = (1,0)$, obtained for $m = (2,1)$ using the characteristic function of a conical sector as the window function w_θ. Kernels are shown for three different direction θ and estimate the derivative $\partial_{u_1} = \partial_\theta$; the absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure.

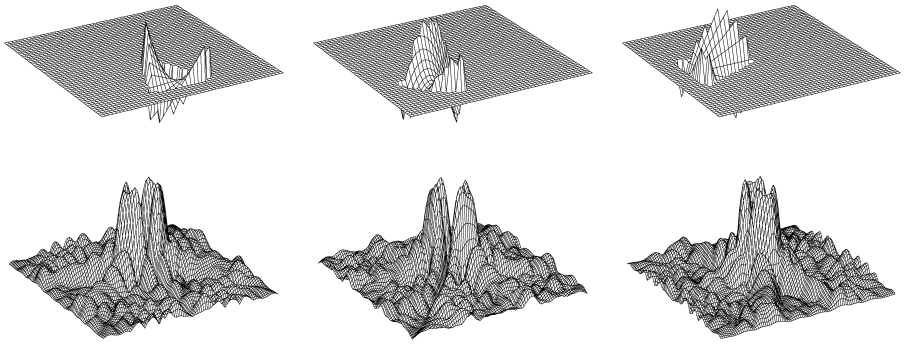


Figure 3.3: Partial-derivative estimation directional *LPA* kernels, $o = (0,1)$, obtained for $m = (2,1)$ using the characteristic function of a conical sector as the window function w_θ. Kernels are shown for three different directions θ and estimate the derivative ∂_{u_2}; the absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure.

Chapter 4

Anisotropic *LPA-ICI*

In this chapter we introduce a novel and original anisotropic estimator for image and multi-dimensional signal restoration. It is the main contribution of the thesis, and it provides the core of all developed applications that are presented in the coming chapters.

The proposed approach originates from the geometric idea of a starshaped estimation neighborhood topology. In this perspective, an optimal adaptation is achieved by selecting – in a pointwise fashion – an ideal starshaped neighborhood for the estimation point. In practice, this neighborhood is approximated by a sectorial structure composed by conical sectors of adaptive size. Special varying-scale kernels, supported on these sectors, are exploited in order to bring the original geometrical problem to a practical multiscale optimization. The directional *LPA* provides sufficient flexibility for the design of the required varying-scale kernels. At the same time, it enables the use of the *ICI* algorithm in order to perform the multiscale optimization in an efficient way.

The resulting estimator is truly anisotropic, providing a clean and accurate edge adaptation and an excellent restoration performance. Its implementation is fast, as it is based on simple convolutions and scalar optimizations. Although we focus initially on image processing, the approach is general and can be extended to higher-dimensional data.

A substantial part of this chapter is based on the publications [20][1] and [46][2].

4.1 Motivation and idea

Let $X \subset \mathbb{R}^2$ be the image domain. We consider the denoising problem of restoration of the image intensity y, $y : X \to \mathbb{R}$, from the noisy observations

$$z(x) = y(x) + \sigma\eta(x), \qquad \eta(x) \sim \mathcal{N}(0,1), \quad x \in X. \tag{4.1}$$

[1] [20]: Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "A novel anisotropic local polynomial estimator based on directional multiscale optimizations", *Proc. 6th IMA Int. Conf. Math. in Signal Processing*, Cirencester (UK), pp. 79-82, December 2004.

[2] [46]: Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Directional varying scale approximations for anisotropic signal processing", *Proc. XII European Signal Proc. Conf., EUSIPCO 2004,* Vienna, pp. 101-104, September 2004.

First, we introduce some basic notation that is used in the following sections.
Notation: We denote by χ_A the characteristic (or indicator) function of a set A, i.e. $\chi_A = 1$ on A, 0 elsewhere; $\mu(A)$ stands for the ordinary Lebesgue measure of the set A. It follows that

$$\int \chi_A(v)\,dv = \int_A 1dv = \int_A dv = \mu(A).$$

Similarly to the characteristic function χ_A, we also define the *normalized indicator* 1_A of the set A, $1_A \triangleq \chi_A/\mu(A)$. Hence, $\int 1_A(v)\,dv = 1$ and integration of a function f against the normalized indicator 1_A of a set A realizes the average of f on A,

$$\int f(v)\,1_A(v)\,dv = \int_A f(v)\,dv/\mu(A).$$

In what follows, we use the term *neighborhood* (of a point x) in a generic sense, meaning a simply connected set (containing x). Relations between sets are always considered up to a null-set.

4.1.1 Estimates with support optimization

Consider a conventional kernel estimator (filter) in the form

$$\hat{y}(x) = \int z(x-v)1_{U_x}(v)dv = \int 1_{U_x}(x-v)z(v)dv = \\ = \int 1_{\tilde{U}_x}(v)z(v)dv = \int_{\tilde{U}_x} z(v)dv/\mu(U_x), \quad (4.2)$$

where U_x is a neighborhood of the origin, and the uniform smoothing kernel 1_{U_x} has support U_x and constant value $1/\mu(U_x)$ on U_x. We use the decoration $\sim$ to denote the translated and mirrored neighborhood about the reference point x, $\tilde{U}_x(\cdot) = U_x(x-\cdot)$, distinguishing it from U_x, which is *always* about the origin.

The bias and the variance of the estimate (4.2) are, respectively,

$$m_{\hat{y}}(x) = y(x) - \int 1_{\tilde{U}_x}(v)y(v)dv \quad \text{and} \quad \sigma^2_{\hat{y}}(x) = \sigma^2/\mu(U_x).$$

The *ideal* support U^*_x, yielding the best mean squared error, can be found by minimization of the quadratic risk $l_{\hat{y}}(x)$:

$$U^*_x = \underset{U_x}{\operatorname{argmin}}\, l_{\hat{y}}(x), \qquad l_{\hat{y}}(x) = m^2_{\hat{y}}(x) + \sigma^2_{\hat{y}}(x). \quad (4.3)$$

Thus,

$$\hat{y}(x) = \int 1_{U^*_x}(x-v)z(v)dv = \int 1_{\tilde{U}^*_x}(v)z(v)dv \quad (4.4)$$

is the best local mean estimate of $y(x)$. The optimization (4.3) can be quite difficult to achieve. In order to make it practical, further specifications of the problem are required.

Starshaped unbiased estimates and the $\mathfrak{U}_x$ topology

We discuss here a simplified model, which will serve as a ground for the development of a more general approach. Let y be a binary black-and-white image, i.e. $y(x) \in \{0,1\}\ \forall x$, and let us restrict our attention to *starshaped unbiased* estimates. It means that we consider only sets U_x which are starshaped with respect to the origin[3] and such that $m_{\hat{y}}(x) = 0$.

The best estimate is obtained by minimization of the variance only or, equivalently, by maximization (with respect to the set inclusion $\subset$) of the set U_x. Unbiasedness holds if and only if $y(v) = y(x)$ for almost every $v \in \tilde{U}_x$. Under mild regularity assumptions on y (e.g. piecewise regular boundary of the level sets), such equality has to hold for every $v \in \tilde{U}_x$. Thus, the best unbiased estimate corresponds to the largest starshaped U_x such that $y(\tilde{U}_x) \equiv y(x)$. This procedure can be formalized nicely in a topological manner.

Let $\mathfrak{U}_x$ be the topology constituted by all sets U_x such that:

(i) $U_x \setminus \{0\}$ is an open set in the Euclidean topology;
(ii) U_x is starshaped with respect to 0;
(iii) $y(x - v) = y(x)\ \ \forall v \in U_x$.

The maximum (with respect to $\subset$) element in $\mathfrak{U}_x$ corresponds to the ideal starshaped unbiased estimate of $y(x)$,

$$U_x^* = \max \mathfrak{U}_x.$$

This suggests a risk minimization strategy based on a progressive set enlargement within this topology. This minimization may be achieved also by "decomposing" $\mathfrak{U}_x$ as follows. Let $\{S_i\}_{i=1}^K$ be a collection of K starshaped neighborhoods of the origin such that $\cup_{i=1}^K S_i = \mathbb{R}^2$ (e.g. a collection of conical sectors). Then, $\mathfrak{U}_x^{S_i} = \{U_x^{S_i} = U_x \cap S_i : U_x \in \mathfrak{U}_x\}$ are also topologies, and $U_x^* = \max \mathfrak{U}_x = \bigcup_{i=1}^K \max \mathfrak{U}_x^{S_i}$. It means that the optimization can be performed independently on each "subcomponent" $\mathfrak{U}_x^{S_i}$.

Examples of the ideal $\tilde{U}_x^*$ are given in Figure 4.1 for two images: the characteristic function of an open disc (top row) and the "Cheese" image (bottom row). In particular, for the first image, depending on the value of x, $\tilde{U}_x^*$ is the unit disc itself for $\|x\| < 1$, the tangent support halfplane to the circle at x for $\|x\| = 1$, and the union of all support halfplanes to the circle containing x for $\|x\| > 1$.

Although different points x', x'' may have $\tilde{U}_{x'}^* = \tilde{U}_{x''}^*$, the corresponding $U_{x'}^*$ and $U_{x''}^*$ are not equal, and in both examples each point x has its own different ideal neighborhood U_x^*. Adapting perfectly to the edges, they are typically non-convex and their shape can be rather complex.

Strictly speaking, starshapedness implies a *line-of-sight* model, in which the estimation point "cannot see" beyond obstacles. Therefore $\tilde{U}_x^*$ never contains points that are not "directly visible" from x: this leads to an important aspect that we exploit in the following sections.

Despite the apparent simplicity of the above speculations, the practical realization of this support-optimization approach can be still hard to achieve, since

[3] A set A is said to be *starshaped with respect to a point* x if, for any $a \in A$, the segment from a to x is contained in A.

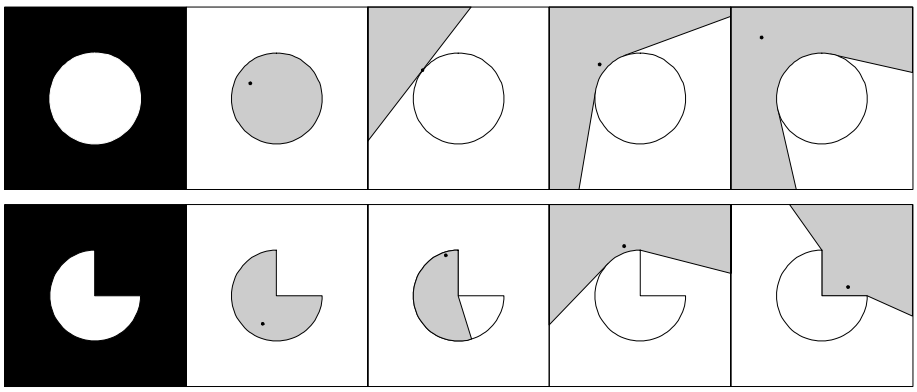


Figure 4.1: Examples of the ideal starshaped neighborhoods $\tilde{U}_x^*$ resulting from $U_x^* = \max \mathfrak{U}_x$. On each row, the first subimage to the left is the true signal y, followed by the illustrations of four different ideal neighborhoods $\tilde{U}_x^*$ corresponding to four different points $x \in X$.

the function y is usually unknown and only its noisy observation z is available. Concerning the topological formalization, it should be pointed out that the major difficulty is not the maximization – which is easy, since it suffices to choose the whole topological space to have a maximal element – but the actual construction of the topology itself.

Moreover, unless y is known to belong to some very specific class, ensuring unbiasedness is not possible, and biased estimates have to be considered.

4.1.2 Estimates with kernel-scale optimization

Another way to adapt to the signal's varying local features, following the majority of multiscale techniques, is to use kernels equipped with a scale parameter h (e.g. $g_h(\cdot) = g(\cdot/h)/h^2$). This estimate can be presented in the form

$$\hat{y}(x) = \int g_{h(x)}(x - v) z(v) dv.$$

The scale optimization can be formulated, similarly to (4.3), as

$$h^*(x) = \operatorname*{argmin}_h l_{\hat{y}}, \qquad l_{\hat{y}}(x) = m_{\hat{y}}^2(x) + \sigma_{\hat{y}}^2(x).$$

The bias and the variance are, respectively, $m_{\hat{y}}(x) = y(x) - \int g_{h(x)}(x - v) y(v) dv$, and $\sigma_{\hat{y}}^2(x) = \sigma^2 \int g_{h(x)}^2(v) dv$. This kind of optimization is known to be practical and can give good results through algorithms of reasonable complexity, such as thresholding (e.g. [66] and references therein), or the *ICI* rule.

When the support of the kernel g_h is bounded, the scale parameter $h(x)$ controls the size of the neighborhood for estimation at the point x. Thus, the support of the ideal scale kernel $g_{h^*(x)}$ can be thought as an approximation of the ideal U_x^* considered in equation (4.3). However, traditional kernels have supports of simple convex geometry (square, rectangle, circle, oval, etc.), whereas the ideal neighborhoods can be quite complex, especially near edges or corners. Thus, this approximation of U_x^* can be quite poor.

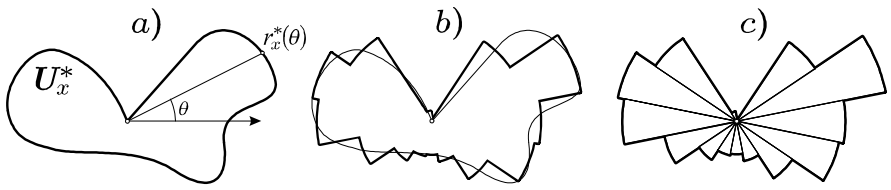


Figure 4.2: Piecewise constant approximation of $r_x^*(\theta)$ and its representation by adaptive-size sectors.

4.2 Anisotropic estimator based on directional adaptive-scale

It would be desirable to find a reasonable compromise between the geometrical approach discussed in Section 4.1.1 and the above kernel-based method.

The previous topological considerations shed some insight on how this sort of compromise is produced and clarify the geometrical properties of the estimator.

The starshapedness of U_x^* allows to describe this set using polar coordinates: there exists a function $r_x^*(\theta)$, $\theta \in [0, 2\pi)$ (see Figure 4.2*a*), such that

$$U_x^* = \{v \in X, v = (v_1, v_2) = (r_v \cos\theta_v, r_v \sin\theta_v) : r_v < r_x^*(\theta_v)\}.$$

Instinctively, one may assume some sort of continuity of r_x^* with respect to its argument θ. This regularity, however, fails in the vicinity of edges where, as in the examples shown in Figure 4.1, $r_x^*(\theta)$ presents sharp transitions. This irregular behaviour is a direct manifestation of the *anisotropy* of y or, roughly speaking, that the function's properties are different in different directions. The most natural model, allowing good approximation of such rapid transitions and also discontinuities, is to assume $r_x^*(\theta)$ as a piecewise constant function of its angular argument, i.e. assuming that the ideal neighborhood U_x^* has a sectorial structure, as shown in Figure 4.2(*b*-*c*).

4.2.1 Anisotropic *LPA-ICI* estimator

In our approach we exploit the above sectorial decomposition. A collection of directional-*LPA* kernels $\{g_{h_j,\theta_k}\}_{h_j \in H, k=1,\dots,K}$ supported on such sectors is designed. Each kernel is characterized by a direction θ_k and a scale parameter h. The corresponding estimate of y is given by the convolution

$$\hat{y}_{h_j,\theta_k} = g_{h_j,\theta_k} \circledast z. \tag{4.5}$$

For a fixed x, we obtain a collection $\left\{\hat{y}_{h_j,\theta_k}(x)\right\}_{h_j \in H, k=1,\dots,K}$ which is multi-scale and multi-directional. For each specified direction θ_k, the *ICI* rule is used to select a pointwise-adaptive scale $h^+(x,\theta_k) \simeq r_x^*(\theta_k)$ that approximates the radius of the ideal neighborhood U_x^*. Let $\hat{y}_{h^+(x,\theta_k),\theta_k}$ be the directional adaptive-scale estimate,

$$\hat{y}_{h^+(x,\theta_k),\theta_k}(x) \triangleq \left(g_{h^+(x,\theta_k),\theta_k} \circledast z\right)(x) \qquad \forall x \tag{4.6}$$

and let

$$\sigma_k^2(x) \triangleq \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)} = var\left\{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right\} \qquad \forall x \tag{4.7}$$

be its variance[4]. All these estimates can be fused in the final *anisotropic estimate* $\hat{y}$ as follows:

$$\hat{y}(x) = \sum_k \lambda(x,\theta_k)\hat{y}_{h^+(x,\theta_k),\theta_k}(x), \qquad \forall x\ . \tag{4.8}$$
$$\lambda(x,\theta_k) = \frac{\sigma_k^{-2}(x)}{\sum_i \sigma_i^{-2}(x),}$$

The weights $\lambda(x,\theta_k)$ in the above convex[5] combination are data-driven adaptive, as $\sigma_k^{-2}(x)$ depends on the adaptive $h^+(x,\theta_k)$. Formula (4.8) embeds and makes clear our basic intentions. We introduce the directional estimates $\hat{y}_{h_j,\theta_k}(x)$, optimize the scale parameter for each of the directions (sectors), and fuse the resulting directional adaptive estimates into the final one $\hat{y}(x)$ using the weights $\lambda(x,\theta_k)$. We call this approach *the anisotropic LPA-ICI technique.*

4.2.2 Adaptive anisotropic kernel and adaptive anisotropic neighborhood

The estimate (4.8) is exactly equivalent to the adaptive kernel estimate

$$\hat{y}(x) = \int g_x^+(x-v)\, z(v)\, dv, \tag{4.9}$$
$$g_x^+ \triangleq \sum\nolimits_k \lambda(x,\theta_k) g_{h^+(x,\theta_k),\theta_k}. \tag{4.10}$$

We call g_x^+ *the adaptive anisotropic kernel* (for the estimation of x).

Let us remark that, despite the appearence, neither (4.6) nor (4.9) is an estimate which can be obtained by a convolution: in both equation an adaptive kernel, which depends on x, is used in the integration against z. A similar issue has been discussed in a footnote on page 11.

We define *the adaptive anisotropic neighborhood* U_x^+ as the union of the supports of the kernels used for estimation,

$$U_x^+ = \bigcup_k \operatorname{supp} g_{h^+(x,\theta_k),\theta_k}. \tag{4.11}$$

Obviously, $U_x^+ \supseteq \operatorname{supp} g_x^+$.[6]

With a notation similar to the one introduced in Section 4.1.1, the $\sim$ decoration is used to denote the translated and mirrored adaptive anisotropic neighborhood, $\tilde{U}_x^+(\cdot) = U_x^+(x-\cdot)$.

Figure 4.3 and Figure 4.4 show some of the adaptive anisotropic neighborhoods resulting from the proposed anisotropic *LPA-ICI* approach for noisy images[7]. A comparison with the lower row from Figure 4.1 shows the similarity between the previous ideal example and this concrete case.

A few other examples of such adaptive anisotropic neighborhoods were given in the Introduction (Figure 1).

[4] In the equation (4.7), we treat h^+ as a purely deterministic variable. This simplification is however quite reasonable, as in practice the adaptive h^+ does not exhibit a significant variability (see Section 4.9).

[5] The combination is convex because, by definition, $\sum_k \lambda(x,\theta_k) = 1$.

[6] It may happen, but only using particularly-exotic kernels, that $\bigcup_k \operatorname{supp} g_{h^+(x,\theta_k),\theta_k} \supsetneq \operatorname{supp} g_x^+$. It means that, during the fusing, adaptive-scale kernels corresponding to different directions cancel-out on a set whose measure is larger than 0. This can happen only if the kernels corresponding to different directions overlap. In the coming sections we will consider

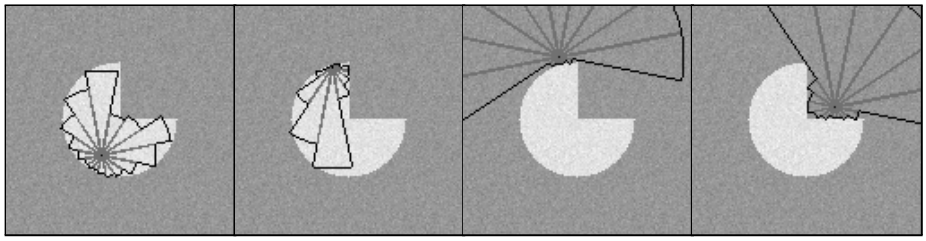


Figure 4.3: *Cheese*: adaptive anisotropic neighborhoods $\tilde{U}_x^+$ obtained through *ICI* using sectorial kernels. Compare with the ideal example shown in the bottom row of Figure 4.1.

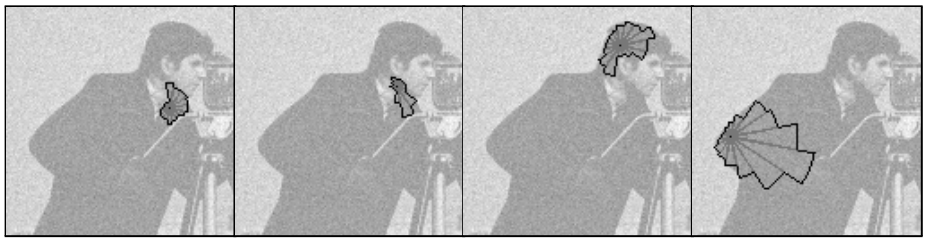


Figure 4.4: *Cameraman* (detail): adaptive anisotropic neighborhoods $\tilde{U}_x^+$ obtained through *ICI* using sectorial kernels

4.2.3 Anisotropic estimation: formal modelling

In this section we introduce a formal model of the anisotropy. Such a model is, to some extent, functional to the design of appropriate families of directional-*LPA* kernels to be used for the anisotropic *LPA-ICI* estimator (4.8). Let us first introduce some notation.

Notation

Given a point $x = (x_1, \ldots, x_d)$ in the d-dimensional Euclidean space $\mathbb{R}^d$, we denote as $u(x, \theta)$ the rotation of x by the spherical angle $\theta = (\theta_1, \ldots, \theta_{d-1})$ about the origin. The standard representation of x in spherical coordinates is written as $(\rho, \theta)_{\text{sph}}$. Here, $(\rho, \theta)_{\text{sph}}$ means the cartesian coordinates of x, given its radius $\rho = \|x\|$ and its spherical angular component θ. For example, for $d = 2$, $x = (\rho, \theta)_{\text{sph}} = (\rho \cos\theta, \rho \sin\theta) = (x_1, x_2)$. The rotated cartesian coordinates are denoted as $\{u_i\}_{i=1,\ldots,d}$. In this notation the directional derivative ∂_θ is nothing but the partial derivative ∂_{u_1} with respect to the first component u_1 of the rotated cartesian axes. Partial derivatives with respect to u_i have the form $\partial_{u_i} y(x) = \lim_{\delta \to 0} \left(y\left(x + u\left(\delta e_i, \theta\right)\right) - y(x) \right) / \delta$, where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the i-th standard basis vector. Higher-order derivatives are defined as the iteration of the corresponding first order derivatives, $\partial_{u_i}^{n+1} = \partial_{u_i} \left(\partial_{u_i}^n \right)$.

The kernel design procedure considered later is quite general and will be

---

the possibility of such overlap. However, even when the kernels overlap, the above strict inclusion is not the generic situation. Thus $U_x^+ = \operatorname{supp} g_x^+$ might be safely considered, instead of (4.11), as an equivalent definition for the adaptive anisotropic estimation neighborhood.

[7] The observations, contaminated by noise with standard deviation $\sigma = 0.1$, are whitened in the Figures for a better visual contrast.

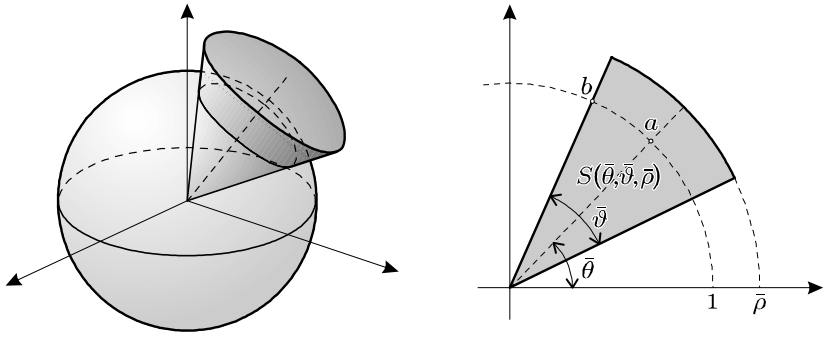


Figure 4.5: Illustration of the sector $S\left(\bar{\theta},\bar{\vartheta},\bar{\rho}\right)$.

given for d-dimensional signals and specified in the rotated variables u_i. It is then convenient to use a compact multi-index notation, as it was defined in Section 1.5.1.

Sectorial modelling of anisotropy

Let $S\left(\bar{\theta},\bar{\vartheta},\bar{\rho}\right)=\left\{x\in\mathbb{R}^d:\|x\|\leq\bar{\rho},\quad\left\|x/\|x\|-(1,\bar{\theta})_{\text{sph}}\right\|\leq 2\sin\left(\bar{\vartheta}/4\right)\right\}$ be the spherical sector of radius $\bar{\rho}\in\mathbb{R}^+$ and aperture angle $\bar{\vartheta}\in[0,2\pi]$ along the direction $\bar{\theta}\in\mathbb{R}^{d-1}$ having its vertex in the origin. The function $2\sin\left(\bar{\vartheta}/4\right)$ simply returns the length of the chord between two points on the unit sphere (a and b in figure 4.5) given the arc $\bar{\vartheta}/2$ that separates them. It is clear that any neighborhood of the estimation point can be covered by a collection of such directional sectors.

For a function to be anisotropic means that the function's properties are different in different directions, i.e. in different sectors around the estimation point. Let us formalize the ideas discussed in the previous section by introducing the following class of locally anisotropic functions:

$$F_\alpha^\Theta=\{y\in L^1_{\text{loc}}\left(\mathbb{R}^d\right):\forall x\in\mathbb{R}^d\ \ \exists\,\bar{\theta}\in\Theta,\ \bar{\vartheta}>0,\ \bar{\rho}>0,\ \overline{L}_\alpha\in\mathbb{R}^{+d}\ \text{ such that }\sup_{v\in S\left(\bar{\theta},\bar{\vartheta},\bar{\rho}\right)}|\partial_u^\alpha y(x-v)|\leq\overline{L}_\alpha\},\quad(4.12)$$

where α is a d-multi-index defining the derivative order, $\Theta\subseteq\mathbb{R}^{d-1}$ is a fixed set of directions and u are the rotated coordinates by the angle $\bar{\theta}$. At any point x, functions $y\in F_\alpha^\Theta$ might have different regularity depending on the direction, but for every point there is always at least one non-trivial sector $S\left(\bar{\theta},\bar{\vartheta},\bar{\rho}\right)$ (directed along one of the directions specified by the set Θ) in which the function satisfies some regularity conditions. Both the sector and the regularity bound depend on the point x.

Traditional models for anisotropy are formulated in cartesian axes (e.g. [51]). The proposed class allows more geometrical freedom. First, because Θ is not restricted to the cartesian directions, and second, because the directionality of the sector $S\left(\bar{\theta},\bar{\vartheta},\bar{\rho}\right)$, which is asymmetrical with respect to its vertex, tolerates discontinuities at x. This is of crucial importance since it enables accurate edge adaptation. The fact that F_α^Θ is a subset of the space of locally absolutely

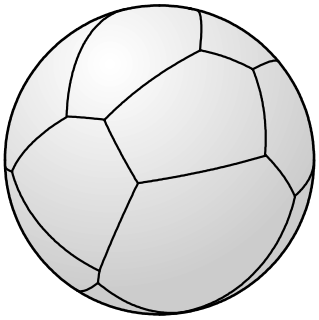


Figure 4.6: Example of a Voronoi tiling of the surface of the sphere. Each patch is the base of a cone which has its vertex at the center of the sphere. These cones constitute a Voronoi tiling for the whole ball.

integrable functions L^1_{loc} is purely technical requirement in order to guarantee the existence of the kernel estimates.

The above definition of anisotropy (4.12) can be a useful model for many typical images as the class F^{Θ}_{α} includes the local polynomial functions (the polynomial smoothness depends on the multi-index α) as well as the spline functions and, more generally, the piecewise-smooth functions whose discontinuities are rectifiable curves.

It is worth observing that whenever $\bar{L}_\alpha = \bar{L}_\alpha(x) = 0$ for $\alpha = m+1$, all derivatives $\partial^\alpha_u y(x-\cdot) = 0$. It means that $y(x-\cdot)$ is a polynomial of the degrees $m_1, \ldots, m_d$ with respect to $u_1, \ldots, u_d$ in the area restricted by $v \in S\left(\bar{\theta}, \bar{\vartheta}, \bar{\rho}\right)$. Note also that polynomials in $u(x,\theta)$ coordinates are also polynomials in $u(x,0) = x$ and, since they are infinitely differentiable, relations such as (3.3) hold for all derivatives.

Keeping this model of anisotropy in mind, we introduce a family of varying-scale kernels whose directionality and regularity are designed for accurate anisotropic estimation.

Covering of the sphere and conical sectors

Let $\{D_{\theta_k}\}_{k=1,\ldots,K}$ be a covering of the unit sphere $\partial B^d = \left\{x \in \mathbb{R}^d : \|x\| = 1\right\}$ with a finite family of non-overlapping[8] contractible bodies[9] (in the sphere topology) $D_{\theta_k} \subset \partial B^d$ whose baricenters have spherical angular components θ_k. For any given $h \in R^+$, $S^h_{\theta_k} = \bigcup_{0\le\alpha\le h} \alpha D_{\theta_k}$ are then the corresponding positive cones constituting an alike covering of the ball $hB^d = \{x \in \mathbb{R}^d : \|x\| \le h\}$ with angular sectors having their vertex in the origin and oriented as θ_k. A classical example of such a family of sectors is the Voronoi tiling of the ball, constructed given a set of points on the sphere, illustrated in Figure 4.6.

[8] Given a family of sets $\{C_k\}_k$, they are said to be non-overlapping if, for any two sets $C_{k'}, C_{k''}$, the intersection of their interiors is empty, $\mathring{C}_{k'} \cap \mathring{C}_{k''} = \varnothing$. However, for our practical considerations, it is enough to assume that sets are non-overlapping if the measure of their intersection is zero, $\mu(C_{k'} \cap C_{k''}) = 0$.

[9] A set C is said to be a body if and only if it is equal to the closure of its non-empty interior, $C = \overline{\mathring{C}} \neq \varnothing$.

The $\hat{y}_{h,\theta_k}(x)$ in (4.5) is the estimate of $y(x)$ using the observations from the sector $S^h_{\theta_k}$. Optimization of h for each of the directions, following Section 4.1.2, gives the adaptive-scales estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}$ depending on θ_k.

It should be noted that the majority of scale-optimization techniques are based on the thresholding in some orthogonal-transform domain. The design of multiscale orthogonal (or bi-orthogonal) decompositions on arbitrarly-shaped regions or with oriented atoms is a rather challenging task. This task becomes particularly hard if one tries to design a domain where "natural" signals have a sparse representation. It is well-known, that latter is a necessary condition for the thresholding to be effective.

On the contrary, narrow and oriented domains, such as the conical sectors $S^h_{\theta_k}$, pose absolutely no complication for the design of directional-*LPA* kernels. Thus, the problem of how to encompass the overall behaviour of the function y within a sector can be dealt efficiently by means of specifically designed directional-*LPA* kernels *supported on the sector*. Endowing these kernels with a scale parameter h allows to formulate the originally geometrical problem into an adaptive-scale selection problem that can be solved by the *ICI* rule.

Kernel design

Let w_{h,θ_k} be a compactly supported window such that $\operatorname{supp} w_{h,\theta_k} = S^h_{\theta_k}$ for all values of the scale parameter h. For example, a possible – and simplest – choice for such windows is $w_{h,\theta_k} = \chi_{S^h_{\theta_k}}$.

According to Section 2.3.2 – at least from a theoretical stand-point – depending on α, one can select an appropriate polynomial order m for the directional-*LPA* kernel design. The set of polynomial generators (3.4) $\left\{\phi_{n,\theta_k}\right\}_{n=1}^{N} = \left\{u^o_{\theta_k}/o!\right\}_{o\in O_m}$ – which is constructed with respect to a rotated coordinate system u_{θ_k} in such a way that u_1 coincides with the main axis of the conical sector $S^h_{\theta_k}$ – is used for the local approximation.

Again according to Section 2.3.2, the Lipschitz bound $\overline{L}_\alpha$ determines a non-zero ideal-scale, which may be approximated by using the *ICI* rule.

Finally, when the sectors $S^h_{\theta_i}$ are all equal up to a rotation, then the windows w_{h,θ_i} can simply be obtained as rotated[10] copies of one *basic window* $w_h = w_{h,0}$. The directional-*LPA* in the rotated variables is then applied independently, for each of the rotated windows, as described in Section 3.3.

Anisotropic estimator

The union U^+_x of the supports of $g_{h^+(x,\theta_k),\theta_k}$, $U^+_x = \bigcup_k \operatorname{supp} g_{h^+(x,\theta_k),\theta_k}$, can be regarded as an approximation of the best local vicinity of x in which the estimation model corresponding to the class F^Θ_α fits the data. Figure 4.7 illustrates this concept and shows sequentially: a local best estimation neighborhood U^*_x, a sectorial segmentation of the unit ball, and the sectorial approximation of U^*_x using the adaptive scales $h^+(x,\theta_k)$ defining the length of the corresponding

[10] In the discrete domain, the rotation method should be chosen accordingly with the smoothness of $w_{h,0}$. If $w_{h,0} = \chi_{S^h_0}$, then the use of linear or higher-order interpolation to perform the rotation is – in practice – not recommendable, as it produces rotated windows $w_{h,\theta}$ that are not characteristic functions. In the majority of our implementations, the simpler nearest-neighbor interpolation has been used to compute the rotated windows.

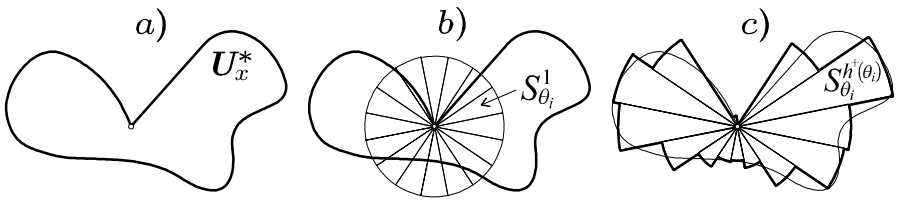


Figure 4.7: a) an ideal estimation neighborhood U_x^*, b) the unit ball segmentation, c) sectorial approximation of U_x^*.

sectors. Varying-scale sectors enable one to get a good approximation of any neighborhood provided that it is starshaped body.

Two points are of importance here.

First, the method is composed by a number of independent scale-optimizations, thus the overall complexity is proportional to the number of sectors, K.

Second, we are able to find good approximations of estimation supports which can be of a complex form; the accuracy of these approximations largely depends on the "directional resolution" K. It would be advisable that the supports $S_{\theta_k}^h$ closely match the sectors $S\left(\bar{\theta}, \bar{\varrho}, \bar{\rho}\right)$ where the Lipschitz bounds $\overline{L}_\alpha$ hold, the ideal case being $\{\theta_i\}_{i=1}^K = \Theta$. However, the sectors $S\left(\bar{\theta}, \bar{\varrho}, \bar{\rho}\right)$, as well as the corresponding Lipschitz bounds are to be considered as *unknown*, and in practice the choice of the directional resolution K depends more on implementation issues (computational and memory limitations, cost-benefit analysis, etc.), rather than on a precise modelling of the function space F_α^Θ.

For the overall algorithm to be effective, we do not even need a covering of the whole sphere, and the considered partition can be restricted to a few narrow cones pointing at different directions and covering only a part of the ideal neighborhood.

The presented construction allows to obtain, through simple-to-implement scalar optimizations, *application specific* anisotropic kernels that adapt to the local image features.

4.3 Anisotropic *LPA-ICI* pseudo-code

What follows is a conventional pseudo-code that describes in detail the different steps that are present in a practical implementation of the anisotropic *LPA-ICI* denoising algorithm.

Some noisy data z is given, and a collection of directional-*LPA* varying-scale kernels $\left\{g_{h_j,\theta_k}\right\}_{\substack{j=1,\dots,J \\ k=1,\dots,K}}$ is provided. It is assumed that z is affected by additive white Gaussian noise of unknown variance. Table 4.1 shows the pseudo-code, following the same notational formalism used in Section 2.4.3 (page 27) for the pseudo-code of the *ICI* algorithm (Table 2.1).

$\sigma = MAD\,(z)$ / estimate noise standard deviation (e.g using robust MAD estimator)

/ begin anisotropic *LPA-ICI* algorithm

$\hat{y} \equiv 0$ / initialize fused estimate buffer

$\varsigma \equiv 0$ / initialize fusing weights buffer

`for` $\theta_k = \theta_1, \ldots, \theta_K$ / loop on θ (directions)

varying-scale *LPA* filtering:

- `for` $h_j = h_1, \ldots, h_J$ / loop on h (scales)
- $\hat{y}_{h_j,\theta_k} = z \circledast g_{h_j,\theta_k}$ / compute dir.-*LPA* estimate
- $\sigma_{\hat{y}_{h_j,\theta_k}} = \sigma \left\| g_{h_j,\theta_k} \right\|_2$ / and its standard deviation
- `end` / end loop on h

ICI:

- $h^+_{\theta_k} \equiv h_1$
- $\hat{y}_{h^+_{\theta_k}} = \hat{y}_{h_1,\theta_k}$
- $\sigma^2_{\hat{y}_{h^+_{\theta_k}}} = \sigma^2_{\hat{y}_{h_1,\theta_k}}$

/ initialization of adaptive scale and of corresponding estimate and variance

- $U = \hat{y}_{h_1,\theta_k} + \Gamma\sigma_{\hat{y}_{h_1,\theta_k}}$
- $L = \hat{y}_{h_1,\theta_k} - \Gamma\sigma_{\hat{y}_{h_1,\theta_k}}$

/ initialization of upper and lower bounds of intersection

- `for` $j = 2, \ldots, J$ / loop on j (scale index)
- $U = \min\{U, \hat{y}_{h_j,\theta_k} + \Gamma\sigma_{\hat{y}_{h_j,\theta_k}}\}$
- $L = \max\{L, \hat{y}_{h_j,\theta_k} - \Gamma\sigma_{\hat{y}_{h_j,\theta_k}}\}$

/ update bounds of intersection

- $T = U \geq L$ / test for non-empty intersection
- $h^+_{\theta_k} = h_j T + h^+_{\theta_k} \mathrm{NOT}(T)$
- $\hat{y}_{h^+_{\theta_k}} = \hat{y}_{h_j,\theta_k} T + \hat{y}_{h^+_{\theta_k}} \mathrm{NOT}(T)$
- $\sigma^2_{\hat{y}_{h^+_{\theta_k}}} = \sigma^2_{\hat{y}_{h_j,\theta_k}} T + \sigma^2_{\hat{y}_{h^+_{\theta_k}}} \mathrm{NOT}(T)$

/ update adaptive scale, estimate and variance

- `end` / end loop on j (scale index)

Fusing:

- $\hat{y} = \hat{y} + \sigma^{-2}_{\hat{y}_{h^+_{\theta_k}}}\ \hat{y}_{h^+_{\theta_k}}$ / fuse adaptive estimates using inverse variances as weights
- $\varsigma = \varsigma + \sigma^{-2}_{\hat{y}_{h^+_{\theta_k}}}$ / sum the fusing weights

`end` / end of loop on θ

$\hat{y} = \dfrac{\hat{y}}{\varsigma}$ / normalization of fused estimate (to achieve convex fusing)

/ end of anisotropic *LPA-ICI* algorithm

Table 4.1: Pseudo-code of the basic version of the anisotropic *LPA-ICI* denoising algorithm.

4.4 Illustrations

The figures and tables in this section present the results of a denoising experiment.

The true signal y is the *Cameraman* grayscale image, and the observation z is given as $z = y + \sigma\eta$, where η is a zero-mean Gaussian noise of unitary variance and $\sigma = 0.1$. Figure 8.1, on page 110, shows this noisy observation.

The discrete kernels $\left\{g_{h_j,\theta_k}\right\}_{j=1,\dots,J,\ k=1,\dots,K}$ used for the experiment are uniform zero-order kernels, $m = (0,0)$, designed on a conical sector of radius h_j, thus they are rather simple. The set of scales is $H = \{1,2,3,5,7,11\}$, where the scale value corresponds to the length – in pixels –of the conical sector, and K=8 directions $\{\theta_k\}_{k=1}^{K} = \{(k-1)\,\pi/4 : k = 1,2,3,4,5,6,7,8\}$. Figure 4.9 shows the sectorial support of the kernels $\left\{g_{h_j,\pi/2}\right\}_{j=1}^{J}$.

Figure 4.8 shows the noisy observation and the adaptive scales $h^+(\cdot,\theta_k)$ that have been selected by the *ICI* algorithm, with a threshold parameter $\Gamma = 1$. The corresponding adaptive-scale estimates $\hat{y}_{h^+(\cdot,\theta_k),\theta_k}$ are shown, together with the final anisotropic estimate $\hat{y}$, in Figure 4.10. Numerical results, with respect to some standard criteria, are given in Table 4.2[11].

A comparison between the directional adaptive scales $\{h^+(\cdot,\theta_k)\}_{k=1}^{K}$ of Figure 4.8 with the non-directional h^+ Figure 2.4 (page 30) highlights the improved efficiency of the directional approach. The most significant difference consists in being able to have, at least for some direction θ_k, an adaptive scale $h^+(x,\theta_k)$ that is large enough, exactly as in the definition (4.12) of the class F_α^Θ. Instead, in the non-directional approach, when approaching an edge, the scale gets inevitably smaller.

Observe also that, when approaching the boundary of the image, the directional adaptive scales decrease only in the corresponding direction. Thus, also for the pixels at the boundary, we are able to find large adaptive scales.

4.5 Fusing: why σ^{-2}?

Indeed, there can be many choices for the definition of the adaptive fusing weights $\lambda(x,\theta_k)$ used in the convex combination (4.8). In this section we justify the choice to use, up to a convexification factor, the inverse of the variances of the adaptive estimates as fusing coefficients $\lambda(x,\theta_k)$. Curiously, this particular convex combination of adaptive estimates has been proposed originally in [28] for averaging any collection of different estimates one-dimensional estimates of

[11] Results analogous to those of Table 4.2, but obtained using with only four sectors (with a wider aperture) and $\Gamma = 1.2$ are given in the table below:

θ_k	$\pi/4$	$3\pi/4$	$5\pi/4$	$7\pi/4$	$\hat{y}$
ISNR (dB)	4.55	4.58	4.55	4.47	7.25
SNR (dB)	18.94	18.97	18.94	18.86	21.64
MAE (ℓ^1)	9.49	9.42	9.44	9.52	6.66
RMSE (ℓ^2)	15.14	15.10	15.14	15.29	11.10
MAX (ℓ^∞)	133.0	120.6	119.7	112.3	98.9

Although, compared to Table 4.2, the criteria values for the directional adaptive-scale estimates are better, the anisotropic fused estimate achieves a lower quality of reconstruction. This shows the advantage of the fusing of a larger number of estimates obtained using "narrower" sectors. Further comments on this results are given in Section 4.6.

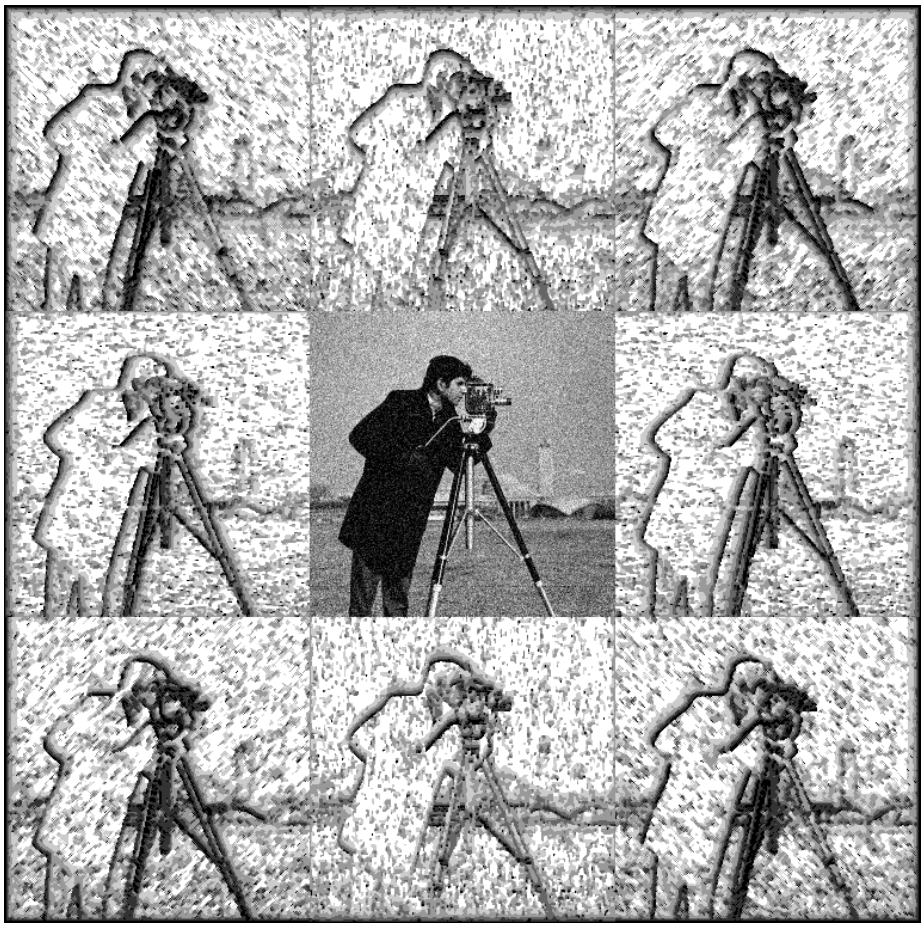


Figure 4.8: Clockwise from top-left, the adaptive scales $h^{+}(\cdot,\theta_k)$, $\theta_k = \frac{7\pi}{4}, \frac{3\pi}{2}, \frac{5\pi}{4}, \pi, \frac{3\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4}, 0$, and, in the center, the noisy observation z ($\sigma = 0.1$). Smaller scales are represented using a darker shade of gray. Observe how the adaptive scales reveal the structures in the image.

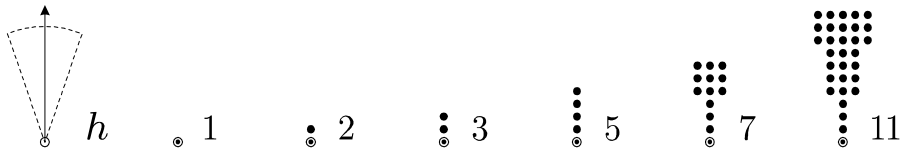


Figure 4.9: The supports of the discrete kernels $g_{h_j,\pi/2}$, $h_j = 1, 2, 3, 5, 7, 11$. The origin pixel is marked with a circle.

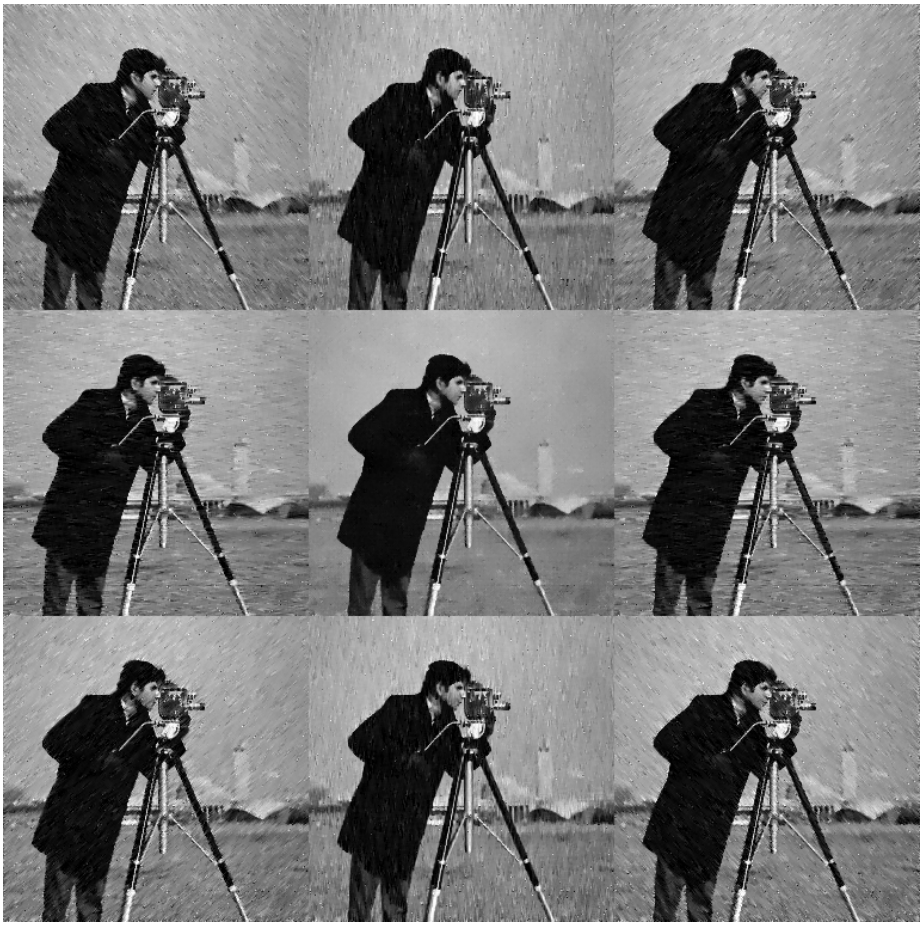


Figure 4.10: Clockwise from top-left, the adaptive-scale estimates $\hat{y}_{h^+(x,\theta_k)}(x)\ \forall x$, $\theta_k = \frac{7\pi}{4}, \frac{3\pi}{2}, \frac{5\pi}{4}, \pi, \frac{3\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4}, 0$, and, in the center, the fused anisotropic estimate $\hat{y}$.

θ_k	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	$\hat{y}$
ISNR (dB)	4.13	3.57	4.08	3.56	4.11	3.44	4.07	3.55	8.07
SNR (dB)	18.52	17.96	18.47	17.95	18.50	17.83	18.46	17.95	22.46
MAE (ℓ^1)	10.67	11.55	10.80	11.59	10.69	11.70	10.82	11.58	6.44
RMSE (ℓ^2)	15.90	16.95	16.00	16.98	15.93	17.21	16.01	16.98	10.10
MAX (ℓ^∞)	131.6	114.7	124.2	117.0	112.6	142.5	114.4	125.9	85.3

Table 4.2: Criteria values for the denoising of the *Cameraman* image using 8 directional adaptive estimates.

different order. We propose two different considerations, the first of statistical flavour, the second more geometrical. In particular, we show:

1. (Section 4.5.1) assuming that the directional adaptive estimates are independent and unbiased, such fusing is the maximum-likelihood estimate of $y(x)$ given $\left\{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right\}_{k=1}^{K}$;
2. (Section 4.5.2) for the simplest choice of kernels, the fused estimate (4.8) is exactly the average of the signal over the anisotropic adaptive neighborhood $\tilde{U}_x^+$, and, in this sense, the anisotropic estimator is just an estimator of the form (4.4), where the ideal neighborhood U_x^* is substituted by its approximation U_x^+.

4.5.1 Fusing unbiased estimates

Let us assume that the directional adaptive estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$, $k = 1,\dots,K$, are independent and unbiased. This latter condition is generally never satisfied[12], nevertheless, some analysis from the coming Section 4.6.3 (in particular a footnote on page 59) shows that in practice they are much less biased than one would expect and that, compared to their overall MSE, the component due to systematic error is negligible. To achieve independency it is enough to assume that the kernels $\left\{g_{h^+(x,\theta_k)}\right\}_{k=1}^{K}$ are non-overlapping[13], so that the estimates are obtained using observations from non-overlapping subsets. When $\operatorname{supp} w_{h,\theta_k} = S^h_{\theta_k}$ the condition is clearly satisfied.

Under these two assumptions, we have that the estimates are independent, normally-distributed random variables with mean $y(x)$ and variance $\sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$,

$$\hat{y}_{h^+(x,\theta_k),\theta_k}(x) \sim \mathcal{N}\left(y(x), \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}\right).$$

The corresponding log-likelihood, is

$$\begin{aligned} L &= \ln \prod_k \left(2\pi\sigma_k^2\right)^{-\frac{1}{2}} e^{-\frac{1}{2\sigma_k^2}\left(\hat{y}_{h^+(x,\theta_k),\theta_k}(x)-y(x)\right)^2} = \\ &= \sum_k -\frac{1}{2}\sigma_k^{-2}\left(\hat{y}_{h^+(x,\theta_k),\theta_k}(x) - y(x)\right)^2 + \ln\left((2\pi)^{-1/2}\sigma_k^{-1}\right) \end{aligned}$$

where $\sigma_k^2(x) \triangleq \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$. Differentiating with respect to y, we obtain

$$\frac{\partial L}{\partial y} = \sum_k \sigma_k^{-2}\left(\hat{y}_{h^+(x,\theta_k),\theta_k}(x) - y(x)\right).$$

By solving $\frac{\partial L}{\partial y} = 0$, we come to the fusing formula (4.8),

$$y(x)\sum_k \sigma_k^{-2} = \sum_k \sigma_k^{-2}\hat{y}_{h^+(x,\theta_k),\theta_k}(x),$$

$$y(x) = \frac{\sum_k \sigma_k^{-2}\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}{\sum_k \sigma_k^{-2}} = \sum_k \frac{\sigma_k^{-2}}{\sum_j \sigma_j^{-2}}\hat{y}_{h^+(x,\theta_k),\theta_k}(x).$$

[12] According to Section 2.3, unless y is accurately polynomial, or the adaptive kernel is a Dirac-delta, the adaptive estimates are always biased estimates.

[13] By "non-overlapping kernels" we mean that the intersection of the supports of any two kernels has measure equal to zero.

4.5.2 Uniform kernels for a uniform anisotropic kernel

In what follows, we revisit the support-optimization approach of Section 4.1.1 in terms of adaptive fusing of directional uniform-kernel estimators. By "uniform" we mean that the kernels are the normalized indicators of their support. For the *LPA*, this is achieved using a uniform[14] window and zero-order polynomials, $m = (0, 0)$.

Let us consider the simplest case, where $g_{h,\theta_k} = 1_{S^h_{\theta_k}} \equiv 1/\int_{S^h_{\theta_k}} = 1/\mu\left(S^h_{\theta_k}\right)$, i.e. where all kernels are constant on their non-overlapping sectorial support. Then, according to Section 4.1.1, $\sigma^2_{\hat{y}_{h,\theta_k}}(x) = \sigma^2/\int_{S^h_{\theta_k}} dv$.

According to (4.8), with $\sigma_k^{-2}(x) \triangleq \sigma^{-2}_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$,

$$\begin{aligned}\lambda(x,\theta_k) = \sigma_k^{-2}(x)/\sum_j \sigma_j^{-2}(x) = \\ = \sigma^{-2}\mu\left(S^{h^+(x,\theta_k)}_{\theta_k}\right)/\sum_j \sigma^{-2}\mu\left(S^{h^+(x,\theta_j)}_{\theta_j}\right) = \\ = \mu\left(S^{h^+(x,\theta_i)}_{\theta_i}\right)/\sum_j \mu\left(S^{h^+(x,\theta_j)}_{\theta_j}\right) = \mu\left(S^{h^+(x,\theta_k)}_{\theta_k}\right)/\mu\left(U_x^+\right)\end{aligned}$$

(the last equality holds because the sectors $S^{h^+(x,\theta_j)}_{\theta_j}$ are non-overlapping), then

$$\begin{aligned}\hat{y}(x) = \sum\nolimits_k \lambda(x,\theta_i)\hat{y}_{h^+(x,\theta_k),\theta_k}(x) = \sum\nolimits_k \frac{\mu\left(S^{h^+(x,\theta_k)}_{\theta_k}\right)}{\mu\left(U_x^+\right)} \frac{\int_{S^{h^+(x,\theta_i)}_{\theta_k}} z(x-v)dv}{\mu\left(S^{h^+(x,\theta_k)}_{\theta_k}\right)} = \\ = \frac{1}{\mu\left(U_x^+\right)} \sum\nolimits_k \int_{S^{h^+(x,\theta_k)}_{\theta_k}} z(x-v)dv = \frac{1}{\mu\left(U_x^+\right)} \int_{\cup_k S^{h^+(x,\theta_k)}_{\theta_k}} z(x-v)dv = \\ = \frac{\int_{U_x^+} z(x-v)dv}{\mu\left(U_x^+\right)} = \int 1_{U_x^+}(x-v)\, z(v)dv \approx \int 1_{U_x^*}(x-v)\, z(v)dv.\end{aligned}$$

Therefore, under these hypotheses, the fused anisotropic estimate $\hat{y}$ (4.8) is exactly identical to the estimate that would have been obtained by using the normalized indicator of the adaptive anisotropic neighborhood U_x^+ as the estimation kernel in formula (4.4)[15]. Figure 4.11 illustrates the described feature of the fusing (4.8), in the case of four sectorial adaptive estimates.

[14] For the window function it is enough to assume that it is constant on its support, because – as it is pointed out in Sections 1.2.2 and 1.3 – the resulting kernel is not affected by the multiplication of the window function by a constant factor.

[15] The difference between the ideal estimate (4.4) and the adaptive estimate (4.8) is

$$\begin{aligned}\int 1_{U_x^+}(x-v)\, z(v)dv - \int 1_{U_x^*}(x-v)\, z(v)dv = \int \left(1_{\tilde{U}_x^+}(v) - 1_{\tilde{U}_x^*}(v)\right) z(v)dv = \\ = \int_{\tilde{U}_x^+ \cap \tilde{U}_x^*} \left(\frac{1}{\mu\left(U_x^+\right)} - \frac{1}{\mu(U_x^*)}\right) z(v)\, dv + \int_{\tilde{U}_x^+ \setminus \tilde{U}_x^*} \frac{z(v)}{\mu\left(U_x^+\right)} dv - \int_{\tilde{U}_x^* \setminus \tilde{U}_x^+} \frac{z(v)}{\mu\left(\tilde{U}_x^*\right)} dv\end{aligned}$$

and can be bounded as follows,

$$\left|\int \left(1_{\tilde{U}_x^+}(v) - 1_{\tilde{U}_x^*}(v)\right) z(v)dv\right| \le \left(\frac{\mu\left(\tilde{U}_x^+ \cap \tilde{U}_x^*\right)|\mu(U_x^*) - \mu(U_x^+)|}{\mu\left(U_x^+\right)\mu(U_x^*)} + \frac{\mu\left(\tilde{U}_x^+ \setminus \tilde{U}_x^*\right)}{\mu\left(U_x^+\right)} + \frac{\mu\left(\tilde{U}_x^* \setminus \tilde{U}_x^+\right)}{\mu\left(\tilde{U}_x^*\right)}\right) \operatorname*{esssup}_{\tilde{U}_x^+ \cup \tilde{U}_x^*} |z|\,.$$

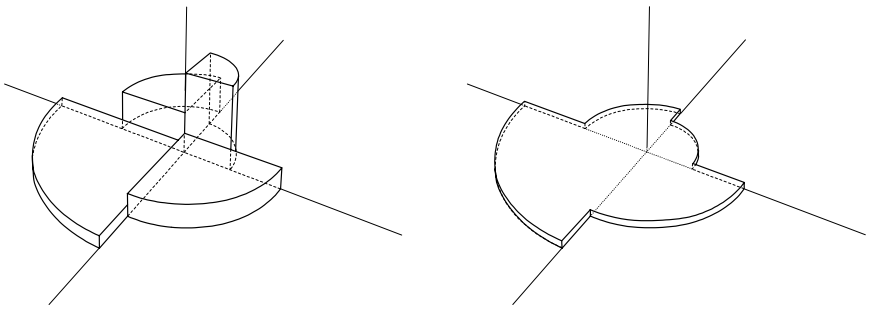


Figure 4.11: Fusing procedure in the case of uniform, non-overlapping kernels. Four sectorial kernels $g_{h^+(x,\theta_k),\theta_k}$ are fused together. The resulting anisotropic kernel g_x^+ (4.10) is exactly the normalized indicator $1_{U_x^+}$ of the adaptive anisotropic neighborhood, which, for non-overlapping kernels, is always equal to the union of the supports of $S_{\theta_k}^{h^+}$ of the individual kernels. Observe that the normalization of the kernels, $\int g_{h^+(x,\theta_k),\theta_k} = \int g_x^+ = 1$, means that the volume of each cylinder in the figure is equal to one.

4.6 Ideal scale h^* and the use of *ICI* for fused estimates

The derivation of the *ICI* rule presented in Section 2.4 is based on the accuracy analysis from Section 2.3. That analysis considered a single estimator. Yet, in the previous sections, we proposed the use of the *ICI* for the optimization of a number of estimates, that we then fuse together. It is quite natural to ask ourselves to which extent the analysis from Section 2.3 can be considered valid for the optimization of multiple estimates, what can be the meaning of "ideal scale" in the multi-directional context, and how the two main aspects of the anisotropic estimator, namely the scale-adaptation and the fusing, interact.

In this section we address these issues by means of analytical derivations based on asymptotics. The result of this analysis shed some insight on the statistical characteristics of the adaptive directional estimates, and – more importantly – highlight the dependancy between the dimension d and Γ. Because of its asymptotical nature, this analysis is subject to the same criticism that was pointed out in Section 2.4.4. It is therefore important to anticipate that the results of this analysis are found experimentally to be not only qualitatively exact, but also quantitatively quite accurate. Thus, they serve as a useful pilot in the algorithm's practical design and optimization.

If the considerations of Section 4.5 can be described as "a look into the fusing from the *ICI*-rule's point-of-view", then this section is sort of "a look into the *ICI* rule from the fusing's point-of-view".

4.6.1 Practical impact of the Γ parameter

The *ICI* algorithm is actually an optimization for the bias-variance tradeoff of varying scale kernel estimates. The ideal scale h^* that minimizes the risk $\bar{l}$ is found, according to the analysis from Section 2.3.3, as the one for which the

ratio between bias and standard deviation reaches a certain value γ (equation (2.8) and (2.10)).

The threshold parameter of the *ICI* algorithm $\Gamma = \gamma + \chi_{1-\lambda/2}$ plays a relevant role in the selection of the adaptive scale. In Section 2.4.4 we observed that a too large Γ makes the intersection of the confidence intervals more likely to be non-empty, and that as a consequence the chosen adaptive scale may be rather large. Analogously, with a small Γ, the empty intersection is likely to happen at a smaller scale. From a purely theoretical point of view, the correct tuning of the Γ parameter consists of choosing the quantile $\chi_{1-\lambda/2}$ in such a way that the adaptive scale h^+ satisfies, with a small margin of error, an equation like (2.8):

$$\frac{\bar{m}_{\hat{y}_{h^+}(x)}}{\sigma_{\hat{y}_{h^+}(x)}} \simeq \gamma.$$

Let us assume that such a tuning of Γ has been performed, and that an "algorithmically" correct value of $\chi_{1-\lambda/2}$ has been found.

4.6.2 MSE of the anisotropic "fused" estimate

If, instead of one individual estimate, we consider a fused estimate such as $\hat{y}(x) = \sum_k \lambda(x,\theta_k)\, \hat{y}_{h^+(x,\theta_k),\theta_k}(x)$, then the mean squared error that one should try to minimize is not $E\left\{\left(y - \hat{y}_{h(x,\theta_k),\theta_k}(x)\right)^2\right\}$, but rather

$$E\left\{\left(y - \sum\nolimits_k \lambda(x,\theta_k)\, \hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right)^2\right\}.$$

Let us assume that the adaptive scales $h^+(x,\theta_k)$ have already been selected. Given the variances $\sigma_k^2(x) = \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$ and bias terms $\bar{m}_k(x) = \bar{m}_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$ of the individual directional adaptive estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$, the variance of the fused estimate $\hat{y}(x)$ (4.8) is (assuming the independence of the directional estimates, i.e. assuming that the kernels $g_{h^+(x,\theta_k),\theta_k}$ don't overlap)

$$\sigma^2_{\hat{y}(x)} = \sum_k \lambda^2(x,\theta_k)\sigma_k^2(x), \tag{4.13}$$

and an upper bound of the bias is [16]

$$\bar{m}_{\hat{y}(x)} = \sum_k \lambda(x,\theta_k)\bar{m}_k(x). \tag{4.14}$$

An upper bound of the anisotropic estimate's mean squared error is thus given by $\bar{l}_{\hat{y}(x)} = \sigma^2_{\hat{y}(x)} + \bar{m}^2_{\hat{y}(x)}$.

[16] The upper bound (4.14) follows easily from

$$\left|(\textstyle\sum_k \lambda_k \hat{y}_k) - y\right| = \left|\textstyle\sum_k \lambda_k(\hat{y}_k - y)\right| \leq \textstyle\sum_k \lambda_k \left|\hat{y}_k - y\right|,$$

where the equality between the first and the second term is a consequence of $\sum_k \lambda_k = 1$. This upper bound can be interpreted as the "worst case" where all the systematic errors have the same sign, preventing thus any compensation between the bias terms to take place while fusing.

4.6.3 A simplified analysis

For convenience, in what follows we assume an even simplier model of the anisotropic neighborhood. We require that, depending on θ_k, $h^+(x,\theta_k)$ is either 0 or some constant h^+. It means that the anisotropic neighborhood is made of a number, say, $\check{K} \leq K$, of sectors and each one of them has the same length h^+. Despite its simplicity, such a model can be still quite reasonable in the vicinity of edges. It also means that $\lambda(x,\theta_k)$ is either 0 or some constant λ^+. The convexity constraint $\sum_k \lambda(x,\theta_k) = 1$ becomes $\check{K}\lambda^+ = 1$, i.e. $\lambda^+ = \check{K}^{-1}$. Without loss of generality, we can further assume that $h^+(x,\theta_k) > 0 \iff k \leq \check{K}$. Therefore, the fused estimate, its variance, the upper bound of its bias, and the upper bound of its MSE, have the form

$$\begin{aligned}
\hat{y}(x) &= \sum\nolimits_{k=1}^{\check{K}} \lambda^+ \hat{y}_{h^+,\theta_k}(x), \\
\sigma^2_{\hat{y}(x)} &= \sum\nolimits_{k=1}^{\check{K}} \left(\lambda^+\right)^2 \sigma^2_{\hat{y}_{h^+,\theta_k}(x)} = \check{K}^{-1}\sigma^2_{\hat{y}_{h^+,\theta_1}(x)}, \\
\bar{m}_{\hat{y}(x)} &= \sum\nolimits_{k=1}^{\check{K}} \left(\lambda^+\right) \bar{m}_{\hat{y}_{h^+,\theta_k}(x)} = \bar{m}_{\hat{y}_{h^+,\theta_1}(x)}, \\
\bar{l}_{\hat{y}(x)} &= \sigma^2_{\hat{y}(x)} + \bar{m}^2_{\hat{y}(x)} = \check{K}^{-1}\sigma^2_{\hat{y}_{h^+,\theta_1}(x)} + \bar{m}^2_{\hat{y}_{h^+,\theta_1}(x)}.
\end{aligned} \tag{4.15}$$

Let us now look for the ideal value of the constant h^+. By recalling the asymptotic expressions (2.5)-(2.6) from Section 2.3.2, and mimicking the analysis of Section 2.3.3, we derive

$$\partial_{h^+}\bar{l}_{\hat{y}} = 0 \iff 2a^2\alpha h^{2\alpha-1} - 2\check{K}^{-1}b^2\beta h^{-2\beta-1} = 0$$

and thus this ideal scale $h^*_{\check{K}}$ is found as

$$h^*_{\check{K}} = \left(\check{K}^{-1}\frac{\beta b^2}{\alpha a^2}\right)^{\frac{1}{2\alpha+2\beta}} = \check{K}^{\frac{-1}{2\alpha+2\beta}} h^*_1, \tag{4.16}$$

where $h^*_1 = h^*$ is the ideal scale found by minimizing the ideal risk of an individual estimate (exactly as in equation (2.7) of Section 2.3.3).

By replacing $h^*_{\check{K}}$ into the bias and variance expressions (2.5) and (2.6) and by considering the ratio between these ideal values of the bias squared and the variance we obtain

$$\begin{aligned}
\bar{m}^2_{\hat{y}_{h^*_{\check{K}},\theta_1}(x)} = a^2\left(\check{K}^{-1}\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{\alpha}{\alpha+\beta}}, \quad \sigma^2_{\hat{y}_{h^*_{\check{K}},\theta_1}(x)} = b^2\left(\check{K}^{-1}\tfrac{\beta b^2}{\alpha a^2}\right)^{\frac{-\beta}{\alpha+\beta}}, \\
\frac{\bar{m}^2_{\hat{y}_{h^*_{\check{K}},\theta_1}(x)}}{\sigma^2_{\hat{y}_{h^*_{\check{K}},\theta_1}(x)}} = \check{K}^{-1}\tfrac{\beta}{\alpha} = \gamma^2_{\check{K}} = \check{K}^{-1}\gamma^2_1 = \check{K}^{-1}\gamma^2.
\end{aligned} \tag{4.17}$$

Directly from (4.16), but also indirectly from the inequalities (2.10), we conclude that $h^*_{\check{K}} \leq h^*_1 = h^*$, with the strict inequality when $\check{K} > 1$. The risk $\bar{l}_{\hat{y}_{h^*_{\check{K}},\theta_1}(x)}$ is then inevitably larger than the ideal risk (2.9) $\bar{l}_{\hat{y}_{h^*,\theta_1}(x)}$ whenever $\check{K} > 1$. In this sense, we say that $\hat{y}_{h^*_{\check{K}},\theta_k}(x)$, $k = 1,\ldots,\check{K}$, are *sub-optimal estimates*.

Such a result is of fundamental importance. Roughly speaking, it suggests two facts:

- (a) the *optimal* fused anisotropic estimate should be composed by a number of *sub-optimal* directional estimates;
- (b) the fusing of *optimal* directional estimates produces a *sub-optimal* anisotropic estimate.

In (a) the sub-optimality of the directional estimates is due to a bias-variance ratio unbalanced towards the variance, whereas in (b) the sub-optimality of the anisotropic estimate is due to a ratio unbalanced towards the bias.

The nature of (a) can be intuited by observing Figure 4.10, where it can be seen that the directional adaptive estimates are rather noisy. The numerical results from Table 4.2, in which the directional estimates present – compared to the fused estimate – surprisingly low values for the objective criteria, confirm the suboptimality of the individual estimates[17]. This suboptimality is necessary to achieve an optimal fused estimate: if the criteria values of the individual estimates were higher (and they could have been higher if a larger value of Γ would have been used), then the results for the final fused estimate would have been lower (too large bias).

4.6.4 Speculations and results on the value of Γ

From equations (4.17) and (2.11), and under the same assumption given at the end of Section 4.6.1, we can derive the following relations between the ideal value of the *ICI* threshold parameter Γ and the number of fused estimates $\breve{K}$:

$$\begin{aligned} \Gamma_1 &= \Gamma = \gamma + \chi_{1-\lambda/2}, \\ \Gamma_{\breve{K}} &= \gamma_{\breve{K}} + \chi_{1-\lambda/2} = \breve{K}^{-\frac{1}{2}}\gamma + \chi_{1-\lambda/2}. \end{aligned} \tag{4.18}$$

It means that the "ideal" (according to the above much simplified analysis) value of the threshold parameter Γ should be adjusted depending on the number of fused sectors and – more generally – depending on the shape of the ideal neighborhood. Qualitatively, the more isotropic the shape, the lower the Γ.

In practice however, one does not know in advance the shape of this neighborhood and unless a costly, iterative procedure – in which the values of Γ are updated recursevely as the shape of the neighborhood is estimated – is implemented, the only practical choice, which we follow in all our programs, is to assume some sort of isotropy (the most natural initial guess on the shape of the neighborhood) in order to use the *ICI* rule. Extensive simulations, as well as the examples provided in this thesis, show that this choice is actually efficient, and that, without turning to the use of further iterations, an accurate enough

[17] From the same table, it is possible, to roughly extrapolate quantitatively the bias-variance ratio for the directional estimates and the fused one. Let us use the approximation (4.15) with an average *MSE* value among directions,$\simeq 280$, and the *MSE* of the fused estimate, $\simeq 100$. We can derive $280 - m^2 \simeq (100 - m^2)\,\breve{K}$, which gives $m^2 \simeq \frac{100\breve{K}-280}{\breve{K}-1}$. For $\breve{K} = 8, 7, 6, 5$, it gives, respectively, $m^2 \simeq 74, 70, 64, 55$. This suggests that the estimation error for the anisotropic fused estimate is – in terms of bias and variance – quite balanced, whereas the bias component of the adaptive directional estimates is nearly negligible when compared to the stochastic error. Thus, one might regard the adaptive directional estimates as "practically unbiased".

estimate of the shape of the ideal neighborhood is obtained even when this ideal shape is substantially anisotropic.

In connection with this observations, it is important to point out that the optimal value of Γ depends – even assuming isotropy – on the total number of directions K.

In the video-denoising simulations presented in the coming Section 10.1, a value of $\Gamma = 0.7$ is used for the 3D denoising (with a total number of directions $K = 26$), whereas the 2D frame-by-frame denoising ($K = 8$) is performed using a larger $\Gamma = 0.9$. Analogously, the image denoising examples of Section 4.4, also obtained for $K = 8$ sectors, use about the same[18] threshold, $\Gamma = 1$, while those corresponding to 4 sectors (given in the footnote of page 51) use $\Gamma = 1.2$. All these results are obtained using uniform zero-order kernels, therefore γ can be assumed to be the same for all experiments.

The fact that the preferred value of Γ gets smaller as K increases confirms the *qualitative* correctness of the above analysis. On the other hand, the mentioned empirical relations between K and Γ can be used to assess *quantitatively* its precision.

To do so, let us consider first the extreme cases for $K = 4$ and $K = 26$. They give, by solving a linear system based on (4.18),

$$\begin{cases} 1.2 = 4^{-\frac{1}{2}}\gamma + \chi_{1-\lambda/2} \\ 0.7 = 26^{-\frac{1}{2}}\gamma + \chi_{1-\lambda/2} \end{cases} \quad \Longrightarrow \quad \begin{array}{l} \gamma = 1.645 \\ \chi = 0.377 \end{array} \quad .$$

Some basic validation can be obtained by inserting these numbers again in (4.18) for $K = 8$. It yields $\Gamma_8 = 8^{-\frac{1}{2}} \times 1.645 + 0.377 = 0.959$, indeed very close to both values of Γ used in the other two experiments.

This accuracy can be verified also for the single estimate ($K = 1$) from the preliminary examples of Section 2.4.5. The best-found "oracle" value for Γ corresponding to Figure 2.5, $\Gamma = 2$, is nearly identical to the value that can be "predicted" by the above analytical formula, $\Gamma_1 = \gamma + \chi_{1-\lambda/2} = 2.022$.

All these results show how (4.18) can be used to predict – quite accurately – the value of Γ for a different number of dimensions d, or for a different "directional resolution" K.

4.7 Uniform fusing for overlapping discrete kernels

Let us consider again the simplest case of zero-order kernels designed on uniform window functions. The resulting directional-*LPA* kernels are simple averaging kernels on their support. If the underlying signal y is a piecewise constant function, then the ideal adaptive kernel for estimation is a uniform averaging kernel. It means that the anisotropic kernel g_x^+ (4.10) should be a uniform kernel. It is shown in Section 4.5.2 that, if the kernels are uniform and non-overlapping, then g_x^+ is also uniform, and in particular

$$g_x^+ = 1_{U_x^+}, \quad \text{and} \quad \hat{y}(x) = \frac{1}{\mu\left(U_x^+\right)} \int_{\tilde{U}_x^+} z(v)\, dv. \tag{4.19}$$

[18] The 0.1 difference between the choices of Γ is negligible, as shown in Section 2.4.5.

If the kernels are overlapping, then g_x^+, as obtained using the fusing formula (4.8), is not necessarily constant on U_x^*.

In the discrete case, the directional kernels *are inevitably overlapping*, since the origin – which belongs to the support of each one of them – is a set of measure one.

However, by exploiting a different fusing formula, it is possible to obtain, using the same directional estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$, a final estimate $\hat{y}(x)$ that exactly coincides with the average of the observation z on the adaptive neighborhood U_x^+ of x.

Let $\Lambda(x) = \sum_j \sigma_j^{-2}(x)$. Clearly, $\lambda(x,\theta_k) = \sigma_k^{-2}(x)/\Lambda(x)$. Then, recalling that for uniform kernels $\sigma_k^{-2}(x) = \sigma^{-2}\mu\left(\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\right)$, we obtain

$$\begin{aligned}\sigma_k^{-2}(x)\hat{y}_{h^+(x,\theta_k),\theta_k}(x) = \\ = \sigma^{-2}\mu\left(\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\right)\int g_{h^+(x,\theta_k),\theta_k}(x-v)\,z(v)dv = \\ = \int_{\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}} \sigma^{-2} z(x-v)dv.\end{aligned}$$

That is, normalization of the sectorial estimates by their corresponding variances is equivalent to the integration on the adaptive sectorial kernel support against the constant weight σ^{-2}. This is true for any (continuous or discrete) uniform kernel.

Consider now the discrete case, and assume that the kernels are all overlapping in and only in the origin. Such a configuration is typical when dealing with discrete data[19]. The fused estimate defined as

$$\hat{y}(x) = \frac{\sigma^{-2}z(x) + \sum_k \left(\sigma_k^{-2}(x)\hat{y}_{h^+(x,\theta_k),\theta_k}(x) - \sigma^{-2}z(x)\right)}{\Lambda(x) + \sigma^{-2}(1-K)} \tag{4.20}$$

is thus exactly equal to $\int_{U_x^+}\sigma^{-2}z(x-v)\,dv / \int_{U_x^+}\sigma^{-2}dv$, which is obviously equal to (4.19). Formula (4.20) allows to obtain a final estimate in the form (4.10) where the anisotropic kernel g_x^+ is uniform over the adaptive anisotropic neighborhood U_x^+.

Let us remark that the above formula makes sense *only* in the discrete case, where the origin (one pixel, in the case of images) is a set of measure one; in the continuous case the origin is a set of measure zero and is therefore negligible with respect to the Lebesgue integration. Observe also that the denominator of (4.20) is indeed always positive: in fact, since we consider the discrete case, $\mu\left(\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\right) \geq 1$, hence $\Lambda(x) \geq \sigma^{-2}K$.

Example

Figure 4.12 illustrates the different filtering ability resulting from the use of the different fusing formula (4.20) compared to the standard fusing (4.8). The surface used in this example is taken from [80]. Observe that the underlying original signal, with the exception of a large step discontinuity, is very smooth

[19] For images, when $K \leq 8$ it is always possible to design the directional window supports in such a way that they don't overlap in pixels other than the origin. For three-dimensional data (e.g. video), such a configuration can be achieved with $K \leq 26$.

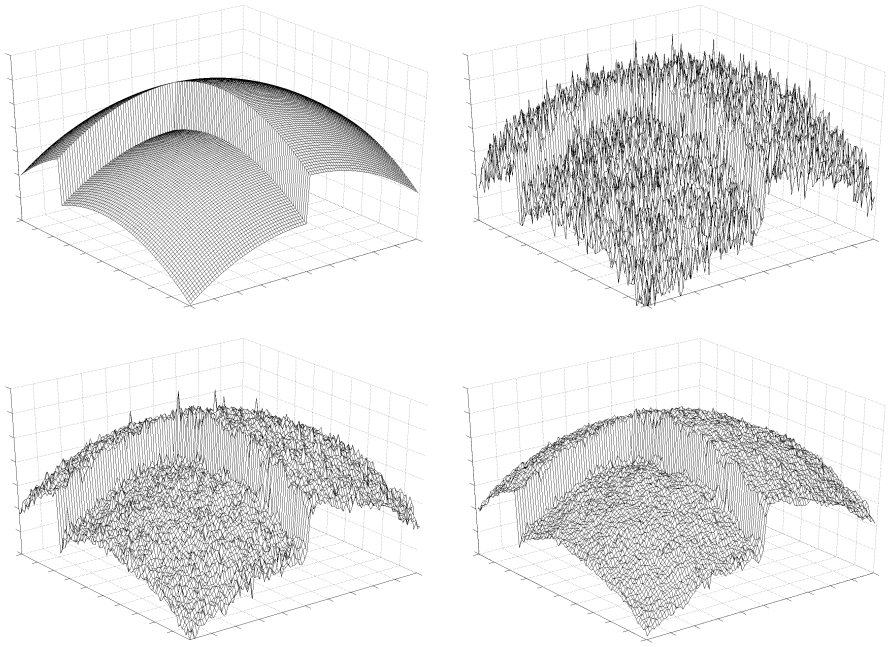


Figure 4.12: The original surface (top left), its noisy observation (top right) and the anisotropic *LPA-ICI* estimates obtained using two different fusing rules, according to formula (4.8) (bottom left) and (4.20) (bottom right).

and it is locally nearly flat, thus enabling (with very good approximation) the assumption that signal is constant on the anisotropic adaptive neighborhood.

Unless the signal presents this type of uniformity, there is no clear advantage in using the more involved formula (4.20), and for the filtering of natural images we always use the much simpler (4.8).

4.8 Variance of the fused estimate

In this section we compute the variance of the final fused estimate $\hat{y}(x)$. It plays an important role for the practical implementation (Section 8.2) of the recursive *LPA-ICI* filtering algorithm (Chapter 6).

The general form of the variance of the final estimate $\hat{y}(x)$ is

$$\sigma^2_{\hat{y}(x)} = \sigma^2 \left\| g^+_x \right\|^2_2 . \tag{4.21}$$

Depending whether the kernels $g_{h^+(x,\theta_k),\theta_k}$, $k = 1, \dots K$, are overlapping or not, and depending on the fusing formula (4.8) or (4.20) which is used to combine these kernels, the actual explicit form of (4.21) is obviously different.

To simplify the notation, we omit here the letter x, and define $\lambda_k = \lambda(x, \theta_k)$, and $\sigma^2_k = \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}$.

4.8.1 Non-overlapping kernels

In the case of non-overlapping kernel supports, formula (4.21) can be easily rewritten in terms of the variances of the individual directional estimates or, equivalently, in term of the individual directional kernels. Indeed, since the supports are non-overlapping, the estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}$ are independent random variables (as k varies). Thus, the variance of $\hat{y}(x)$ has the simple form

$$\sigma^2_{\hat{y}(x)} = \sum_k \lambda_k^2 \sigma_k^2$$

and from the expression of the coefficients $\lambda_k = \lambda(x, \theta_k)$ (4.8), follows that

$$\begin{aligned} \sigma^2_{\hat{y}(x)} &= \sum_k \left(\frac{\sigma_k^{-2}}{\sum_j \sigma_j^{-2}} \right)^2 \sigma_k^2 = \sum_k \frac{\sigma_k^{-2}}{\left(\sum_j \sigma_j^{-2}\right)^2} = \\ &= \left(\sum_k \sigma_k^{-2}\right)^{-1} = \sigma^2 \left(\sum_k 1/\left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2\right)^{-1}. \end{aligned} \quad (4.22)$$

Thus the variance of the fused estimate (4.8) is equal to the inverse of the sum of the variances of the directional estimates.

4.8.2 Origin-overlapping kernels

If the kernels $g_{h^+(x,\theta_k),\theta_k}$ are overlapping, then the directional estimates are no longer independent, and their respective overlap areas should be taken into account. We consider the situation where the directional kernels always have the origin as the only common point. In this case the variance of the fused estimate becomes

$$\sigma^2_{\hat{y}(x)} = \sigma^2 \sum_k \lambda_k^2 \left\| g_{h^+(x,\theta_k),\theta_k}\big|_{x\neq 0} \right\|_2^2 + \sigma^2 \left(\sum_k \lambda_k g_{h^+(x,\theta_k),\theta_k}(0)\right)^2 =$$

$$= \sigma^2 \sum_k \left(\lambda_k^2 \left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2 - \lambda_k^2 g^2_{h^+(x,\theta_k),\theta_k}(0)\right) + \sigma^2 \left(\sum_k \lambda_k g_{h^+(x,\theta_k),\theta_k}(0)\right)^2.$$

By substituting the coefficients λ_k with their expressions in terms of the variance of the directional estimates, we obtain $\lambda_k = \sigma_k^{-2}/\sum_j \sigma_j^{-2} = \sigma_k^{-2}/\Lambda$, where $\Lambda = \sum_j \sigma_j^{-2}$. More precisely, $\lambda_k = \Lambda^{-1}\sigma^{-2}/\left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2$, and $\Lambda = \sum_j 1/\sigma^2 \left\|g_{h^+(x,\theta_j),\theta_j}\right\|_2^2$.

$$\begin{aligned} \sigma^2_{\hat{y}(x)} &= \sum_k \left(\frac{\Lambda^{-2}\sigma^{-2}}{\left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2} - \frac{\Lambda^{-2}\sigma^{-2} g^2_{h^+(x,\theta_k),\theta_k}(0)}{\left(\left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2\right)^2} \right) + \frac{\sigma^{-2}}{\Lambda^2} \left(\sum_k \frac{g_{h^+(x,\theta_k),\theta_k}(0)}{\left\|g_{h^+(x,\theta_k),\theta_k}\right\|_2^2} \right)^2 = \\ &= \Lambda^{-2} \sum_k \sigma_k^{-2} - \frac{\sigma^2}{\Lambda^2} \sum_k \frac{g^2_{h^+(x,\theta_k),\theta_k}(0)}{\left(\sigma_k^2\right)^2} + \frac{\sigma^2}{\Lambda^2} \left(\sum_k \frac{g_{h^+(x,\theta_k),\theta_k}(0)}{\sigma_k^2} \right)^2 = \\ &= \frac{\sum_k \sigma_k^{-2} - \sigma^2 \sum_k \left(\frac{g_{h^+(x,\theta_k),\theta_k}(0)}{\sigma_k^2} \right)^2 + \left(\sigma \sum_k \frac{g_{h^+(x,\theta_k),\theta_k}(0)}{\sigma_k^2} \right)^2}{\left(\sum_j \sigma_j^{-2}\right)^2}. \end{aligned} \quad (4.23)$$

Observe that when only one kernel is non-zero in the origin, i.e. $g_{h^+(x,\theta_k),\theta_k}(0) = 0 \; \forall k \neq \bar{k}$, the last formula reduces exactly to equation (4.22).

4.8.3 Uniform fusing

The unelegant (4.23) reflects the "non-uniformity" of the fusing (4.8) for kernels that are overlapping in the origin.

For such kernels, the variance of the fused estimate takes a much simpler expression if the corresponding adaptive estimates are fused according to the uniform fusing (4.20).

Using this formula the resulting anisotropic kernel g_x^+ is uniform on the anisotropic neighborhood U_x^+, where it is equal to $1/\mu\left(U_x^+\right)$. In particular, it is rather easy to realize that

$$\mu\left(U_x^+\right) = \sum\nolimits_{k=1}^{K} \mu\left(\operatorname{supp} g_{h^+(x,\theta_k),\theta_k}\right) - K + 1 = \sigma^2 \Lambda\left(x\right) - K + 1.$$

Hence, we can calculate the variance of $\hat{y}\left(x\right)$ as,

$$\sigma^2_{\hat{y}(x)} = \sigma^2 \left\|g_x^+\right\|_2^2 = \sigma^2 \frac{\mu\left(U_x^+\right)}{\left(\mu\left(U_x^+\right)\right)^2} = \frac{\sigma^2}{\sigma^2 \Lambda\left(x\right) - K + 1}. \tag{4.24}$$

4.9 Robust *ICI* for anisotropic estimation

We propose a special type of weighted order-statistics (WOS) filters to be used *within* the *ICI* algorithm, so to reduce the impact of the randomness of the noise on the adaptive scales, and thus to improve the efficiency of the *ICI* rule. The action of this nonlinear filter is to correct some of the mistakes in the selection of the adaptive scale. These mistakes are a kind of "false alarm" triggered by the detection of the empty intersection of confidence intervals.

They are clearly visible in the adaptive-scales diagram for the *Cheese* image, shown in the first two subimages to the left in Figure 4.13. From these images, it is easy to recognize the impulsive nature of the errors[20]. On the right part of the same figure are shown the "corrected" adaptive scales obtained exploiting the proposed directionally-weighted WOS filters.

This filtering has an obvious beneficial impact on the quality of the corresponding adaptive-scale estimate y_{h^+}, and since its computational cost is – compared to the entire anisotropic *LPA-ICI* algorithm – marginal, it is exploited in most of the restoration experiments presented in this thesis. Moreover, since the proposed filters are based *explicitly* on the directional nature of the directional-*LPA* kernels, they can be rightly considered as an intrinsic part of the overall anisotropic *LPA-ICI* approach.

Before describing how this filtering is performed, let us make a few remarks.

Preliminaries

It should be clear by now, that choosing a higher threshold parameter Γ effectively helps to reduce the number of these false alarms[21]. However – as it has been discussed in the previous sections – this also leads to oversmoothing of

[20] A rather complete statistical and experimental study of the adaptive-scale selection errors of the *ICI* algorithm is presented in [89].

[21] Also for this reason, in the convergence-rate analysis in [28], it is proposed to use an increasing $\Gamma = \mathcal{O}(\sqrt{\ln n}) \to \infty$.

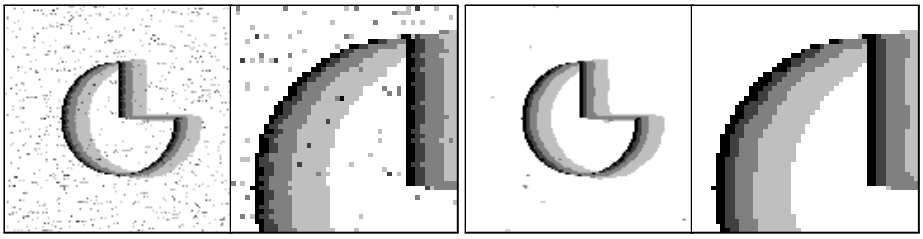


Figure 4.13: Adaptive scales $h^+(\cdot, 0)$ for the *Cheese* image resulting from standard *ICI* (left) and from the proposed *directionally* WOS-*filtered ICI* (right). Smaller scales are pictured with a darker shade of gray. For both cases an enlarged portion is also shown.

details and small features in the signal since the algorithm would allow more bias to "leak" into the estimate.

In order to suppress the impulsive noise present in the adaptive scales, it has been originally suggested in [44] to take advantage of the correlation between the ideal scales relative to nearby pixels by using median filters. Such filtering, which is performed on the adaptive scales h^+ *after* the *ICI* algorithm, has however the drawback of suppressing not only the outliers originated from an undesired empty intersection of confidence intervals, but also some small scales that are related to actual details in the picture.

Let us say that such a median filtering is too simple as it does not take into account the structure of the *ICI* rule, which is based on a recursive intersection of confidence intervals. More precisely, the simple filtering attempts to correct the erroneous adaptive-scales, and not the cause of the error. The cause lies in an "unwanted" empty intersection of confidence intervals at some point during the algorithm recursions.

Our first claim is that the filtering should not be done after the *ICI* algorithm, but rather *within* the algorithm. The intersection tests (denoted by T in the algorithm pseudo-code) need to be filtered.

Second, if we are using a directional kernel – which has an asymmetric support – it is quite unnatural to use a symmetric median filter to perform the filtering. Instead, the nonlinear filter to be used for correcting the impulsive errors should also be directional.

From these two observations follows the proposed filtering strategy: to use oriented (weighted) non-linear filters to clean the intersection of confidence intervals at each iteration of the *ICI* algorithm.

4.9.1 WOS filters

Weighted order statistics filters (WOS) [101] are a generalization of the weighted median filters (and thus also of the classical median filter). For a given mask of weights $\mathbf{W} = [W_1, \ldots, W_N]$, $W_n \in \mathbb{N}$, and an offset constant[22] $O \in \mathbb{N}$, $1 \leq O \leq \sum W_N$, the output of the WOS filter on a set of observations $\mathbf{Z} = [Z_1, \ldots, Z_N]$

[22] O is often called *the threshold* of the WOS filter.

is

$$WOS_{\mathbf{W},O}(\mathbf{Z}) = O\text{-th smallest of } \{W_1 \Diamond Z_1, \ldots, W_N \Diamond Z_N\},$$
$$\text{where} \quad W_i \Diamond Z_i = \underbrace{Z_i, \ldots, Z_i}_{i \text{ times}}$$

The filtering of the whole signal is performed by applying the $WOS_{\mathbf{W},O}$ operation in a sliding fashion.

The classical 3×3 median, maximum and minimum filters correspond to $\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ and, respectively, $O = 5$, $O = 9$, and $O = 1$.

4.9.2 Anisotropic *LPA-ICI* WOS filters

We restrict our attention to the case where $K = 8$ and

$$\{\theta_k\}_{k=1}^{K} = \{(k-1)\,\pi/4 : k = 1, 2, 3, 4, 5, 6, 7, 8\}.$$

It is the most frequently used directional decomposition in our implementations of the anistropic *LPA-ICI* technique.

After experimental optimization, the following set of WOS masks $\mathbf{W}_{\theta_k}$ and have been found to give good results with an offset $O_{\theta_k} = 5$, $k = 1, \ldots, 8$:

$$\mathbf{W}_0 = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{W}_{\pi/4} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 0 \\ 2 & 1 & 1 \end{bmatrix}, \quad \mathbf{W}_{\pi/2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix},$$

$\mathbf{W}_{\theta_k}$, $k = 4, \ldots, 8$ are obtained by counter-clockwise rotation of these masks.

These WOS are in fact weighted medians, since $\sum_i W_i = 9$ and the offset is equal to 5.

The masks $\mathbf{W}_\theta$ rotate counter-clockwise, with a phase difference of π with respect to the kernels $g_{h,\theta}$. The weights W are actually matching the supports of the kernels $\tilde{g}_{h,\theta}$, and $\mathbf{W}_\theta$ and $\tilde{g}_{h,\theta}$ rotate in phase.

Such WOS filters operate on the test of intersections, thus in the pseudo-code of Table 2.1 (Section 4.3) the following modification could be made:

$$T = U \geq L \quad \rightsquigarrow \quad T = WOS_{\mathbf{W}_{\theta_k},O}(U \geq L) \quad .$$

They are enabled in the algorithm starting from the second iteration (i.e. when testing the intersection of three confidence intervals), leaving unfiltered the first intersection test, which is very delicate and should not be compromised. On the other hand, when the scales are large enough, the WOS filtering can be applied more than once, in order to exploit the correlation on a wider support[23].

We apply the same filtering strategy also on the resulting adaptive scales $h^+(\cdot, \theta_k)$, using the same offset and masks $\mathbf{W}'_{\theta_k}$ with larger weights for the central pixel and for the two pixels on its side[24]:

$$\mathbf{W}'_0 = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 4 & 1 \\ 1 & 2 & 0 \end{bmatrix}, \quad \mathbf{W}'_{\pi/4} = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 4 & 0 \\ 2 & 1 & 2 \end{bmatrix}, \quad \mathbf{W}'_{\pi/2} = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

[23] Repeating twice a WOS filter with a mask of size 3×3 can be interpreted as a special WOS filter with a mask of size 5×5.

[24] Since $\sum W'_i = 13$ and $O = 5 \neq (13+1)/2 = 7$, $WOS_{\mathbf{W}'_{\theta_k},O}$ is not a weighted median but really a general WOS filter.

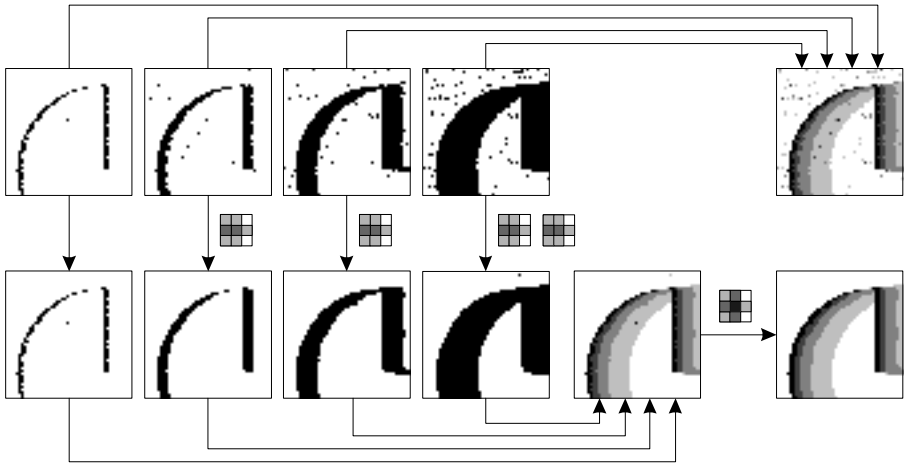


Figure 4.14: Weighted order-statistics filtering within the *ICI* algorithm. First row is standard (unfiltered) *ICI*. Single steps of the standard algorithm are filtered, then the combined adaptive scales are filtered again, yielding the final $h^+(\cdot, 0)$.

$\mathbf{W}'_{\theta_k}$, $k = 4, \ldots, 8$ are again obtained by obvious rotations.

Figure 4.14 illustrates this procedure for the adaptive scales of a detail of the *Cheese* image.

4.9.3 Binary WOS as thresholding of a linear filter

The action of a WOS filter on a binary 0-1 image is equivalent to a much faster operation, based on linear filtering. It makes the WOS-filtering of the tests of intersection a computationally attractive approach for the improvement of the *ICI* algorithm.

Let us use the same notation that was used for the definition of $WOS_{\mathbf{W},O}$. It is evident that, since $Z_i \in \{0,1\}\ \forall i$, $WOS_{\mathbf{W},O}(\mathbf{Z}) \in \{0,1\}$. In particular, after sorting from smallest to largest, the set $\{W_1 \Diamond Z_1, \ldots, W_N \Diamond Z_N\}$ is a vector $Z_{wsort} = [0, \ldots, 0, 1, \ldots, 1]$ of length $\sum_i W_i$. The vector Z_{wsort} contains $S_0 = \sum_{i:Z_i=0} W_i$ zeros and $S_1 = \sum_{i:Z_i=1} W_i = \sum_i W_i Z_i$. Obviously $S_0 = \sum_i W_i - S_1$. The O-th smallest element of $\{W_1 \Diamond Z_1, \ldots, W_N \Diamond Z_N\}$ is the O-th element of Z_{sort}. So $WOS_{\mathbf{W},O}(\mathbf{Z}) = 1$ if and only if $S_0 < O$, i.e. if and only if $S_1 > \sum_i W_i - O$.

Thus, the result of the $WOS_{\mathbf{W},O}$-filtering of a binary image B can be obtained as follows: first, filter B using $\mathbf{W}$ as a linear filter mask, e.g. by convolution against a π-rotated copy of $\mathbf{W}$, say $\mathbf{W}^{\mathrm{rot}\pi}$, and second, check the condition $B \circledast \mathbf{W}^{\mathrm{rot}\pi} > \sum_i W_i - O$. Where this condition is true, then $WOS_{\mathbf{W},O}(B) = 1$, otherwise it is 0.

4.10 Algorithm complexity and implementation issues

4.10.1 Complexity

We briefly discuss the computational complexity of the anisotropic *LPA-ICI* algorithm.

First, let us consider the cost of an individual direction θ_k.

The computational cost of calculating an entire estimate $y_{h_j,\theta_k}(\cdot)$ depends on the size of the observations[25], M, and on the size of the kernel, N_{h_j}. Observe that $N_{h_j} \propto h_j^d$. If convolutions are performed in the space domain, the complexity is $\mathcal{O}\left(MN_{h_j}\right)$. If N_{h_j} is particularly large (for example if $N_{h_j} \sim M$), then discrete convolution in the frequency domain based on the FFT can be used, and the cost of computing the estimate is $\mathcal{O}(M \log M)$.

Since we need to compute sets of varying-scale estimates, the overall cost for a single direction can be bounded by $\mathcal{O}(MJN_{h_J})$, where J is the number of scales and N_{h_J} is the size of the largest kernel. As the image size M increases, one should consider whether it is appropriate to use larger scales or not. If the noise level is assumed constant with respect to M, then N_{h_J} can be considered as fixed, thus the complexity is $\mathcal{O}(M)$. If, on the contrary, the randomness of the noise increases with M[26], then N_{h_J} should also increase, say, proportionally to M. There are two possibilities, either J is fixed, and the complexity is $\mathcal{O}(M \log M)$, or the number of scales is allowed to increase together with h_J. The typical choice is to have $J \sim \log h_J \sim \log \sqrt[d]{N_{h_J}} = \frac{1}{d} \log N_{h_J} \propto \log M$, thus the complexity is $\mathcal{O}\left(M \log^2 M\right)$. The *ICI*, even when the WOS filters are used, has a complexity $\mathcal{O}(MJ)$, thus the adaptive scale selection does not affect the complexity of the algorithm.

Since it is rather unlikely that the number of directions K would need to be increased together with M, and since we can certainly assume that all directions have a similar computational cost, the total complexity of the anisotropic *LPA-ICI* algorithm can be estimated as $\mathcal{O}(M)$, $\mathcal{O}(M \log M)$, or $\mathcal{O}\left(M \log^2 M\right)$, depending on the progression of H with respect to M, and depending on the restrictions we are ready to impose.

4.10.2 Implementation aspects

Although it may not be completely transparent from the theoretical description, not only the directional estimates $\hat{y}_{h_j,\theta_k}(\cdot)$ are computed "globally" as a convolution, but also the *ICI* is performed globally, simultaneously for the whole image, and the presented pseudo-codes are both written in matrix/array form. This enables a rather fast implementation also in environments with a higher level of abstraction, such as Matlab.

To our knowledge, all other non-parametric estimation techniques that exhibit some sort of anisotropic adaptation (e.g. [81],[78]), are based on costly iterative procedures repeated in a pointwise manner.

[25] For example, for a 128×128 image ($d = 2$), $M = 16384$.

[26] This is the typical behaviour that arises when the pixel-density of digital imaging sensors (CCD, CMOS) is increased.

The directional estimations are performed independently for each θ_k, and can thus be implemented as parallel processes. Also the varying-scale filtering can be parallelized, therefore, in a fully-parallel implementation, the total time required for the overall algorithm does not exceed the time of computing an estimate for the largest scale, plus the time required for one *ICI* algorithm execution.

Chapter 5

LPA-ICI for signal-dependant or space-variant noise

The standard *LPA-ICI* filtering algorithm, as it was proposed in [28], assumes that the observations $z(x)$ follow the additive white Gaussian noise (AWGN) model,

$$z(x) = y(x) + \sigma\eta(x),$$

where σ is a positive constant and η is standard Gaussian white noise, $\eta(x) \sim \mathcal{N}(0,1)$. In spite of its simplicity, this model can be adequate for many applications. However, there are a number of cases where more general noise models need to be considered.

The use of the *LPA-ICI* approach, for observations that do not follow the above model, is a rather delicate issue, which needs at least some basic analysis. Let us first introduce two classical observation models: the signal-dependant noise, and the space-variant noise; the required analysis will be then presented within the framework of the models.

5.1 Signal-dependant noise

In many applications the observed signal is corrupted by a signal-dependent noise. The most widely encountered models are Poisson, film-grain, multiplicative and speckle noise. Their common feature is that the variance of the noise is directly related to the signal. There are a number of adaptive approaches to this sort of observations based on local-statistics' calculation. In particular the filters by Lee [61, 62] and Kuan [60] are well known in the field.

Let us consider the observations $z(x)$, $x \in \mathbb{R}^d$, with the expectations

$$E\{z(x)\} = y(x),$$

where the errors $\eta(x) = z(x) - y(x)$ are independent and the variance of these observations is modeled as

$$\sigma_z^2(x) = var\{z(x)\} = var\{\eta(x)\} = \rho(y(x)), \tag{5.1}$$

ρ being a given positive function of y called the *variance function*. For example, we have $\rho(y) = (1-y)y$ and $\rho(y) = y$ for the Bernoulli and Poisson models respectively. As usual, the problem is to reconstruct the signal y from the noisy observations z.

The above observation model can be written also in the following, additive form:

$$z(x) = y(x) + \rho^{1/2}(y(x))\, n(x), \tag{5.2}$$

where ρ is a function that controls the gain of the noise component and n is some independent noise with variance equal to one and mean equal to zero. This noise is not necessarily Gaussian and, generally, $n(x)$ can have different distributions for different points x.

Observe that, in principle, with the exclusion of certain particular cases (for example when ρ is a constant), estimating $\sigma_z^2(x)$ is equivalent to estimating $y(x)$ itself, since if $y(x)$ is unknown also $\rho(y(x))$ is.

As an example of the above model, we consider, in the next section, the Poisson observations. However, the model (5.1) is more general. In particular, it is especially useful for those problems where the probability distribution of the observation error is unknown[1].

5.1.1 Poisson observations

A traditional example of signal dependant noise is the (direct) Poisson observations model. It is most frequently used for applications based on photon-counting. Many biomedical imaging techniques (x-ray scan, PET, etc.), but also the widespread and mass-produced digital imaging sensors (CCD, CMOS) are thus often modeled as Poisson observations.

Given a true signal y, its Poissonian observations are given as

$$z(x) \sim \mathcal{P}(y(x)), \qquad \forall x,$$

where $\mathcal{P}(\lambda)$ denotes the Poisson distribution of parameter λ. Here, $z(x)$ is the measured *integer* number (count) of received/detected photons at the location x. Its mean value $E\{z(x)\}$ is usually denoted in the literature as $\lambda(x)$ and it represents the signal/object mean intensity at the location x, with brighter objects emitting a higher number of photons during the observation period (exposure time). The (discrete) probability function, the mean and the variance of Poisson random variables $z(x)$ are, respectively,

$$\begin{aligned} P(z(x) = k) &= e^{-\lambda}\frac{\lambda^k}{k!}, \qquad k \in \mathbb{N}, \\ E\{z(x)\} &= \lambda(x), \\ var\{z(x)\} &= \sigma_z^2(x) = \lambda(x). \end{aligned}$$

Equation (5.2) takes then the form

$$z(x) = \lambda(x) + \sqrt{\lambda(x)}\eta(x).$$

[1] These problems are usually solved in the context of the quasi-likelihood [94]. In [22], we show how the recursive *LPA-ICI* filters can be interpreted as a special kind of quasi-likelihood estimates.

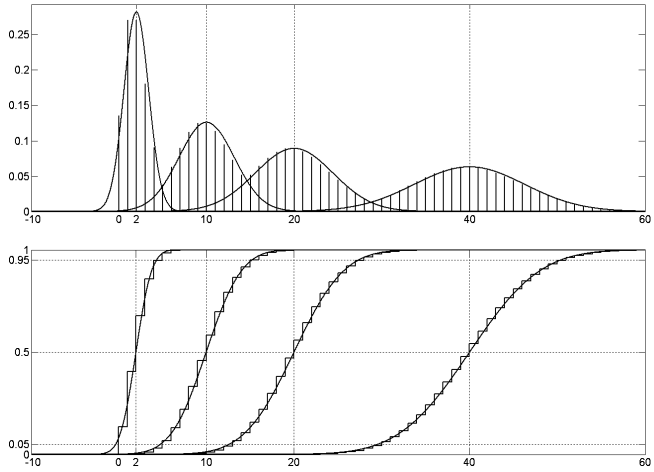


Figure 5.1: Probability densities (top) and distributions (bottom) for Poisson and Gaussian ($\mathcal{N}(\lambda, \lambda)$) models, $\lambda = 2, 10, 20, 40$. Since the Poisson distribution is discrete, in the top chart are shown scaled Dirac-impulses (discrete weights) and not the density itself (which is not defined as an ordinary function).

5.2 Space-variant noise

For the signal-dependant noise, the variances $\sigma_z^2(x)$ depend only on the expected value of the observations, i.e. on the underlying signal $y(x)$. However, in general, one may allow $\sigma_z^2(x)$ to be just a (known or unknown) generic function, independent of y.

Formula (5.2) is, thus, generalized to the following space-variant observation model,

$$z(x) = y(x) + \sigma_z(x)\,\eta(x). \tag{5.3}$$

Such stochastic processes are commonly called, especially in the financial literature, *heteroskedastic*, meaning that the standard deviation $\sigma_z(x)$ is not constant with respect to the spatial (or temporal) variable x.

Usually the actual distribution of $\eta(x)$ is unknown and only some of its moments are known with sufficient accuracy. In what follows, we assume that η is an independent zero-mean noise with variance equal to one. None of the higher moments is assumed to be known, and η may also be non-identically distributed.

Such generality may result in the potential sub-optimality not only of the linear *LPA* filters, but also of any linear filters (typically if the distributions are heavy-tailed). One way to improve the performance of kernel estimators is by reweighting using the inverse variance of the integrated samples. However, such a procedure sensibly increases the computational complexity, since filtering cannot be performed by convolutions (the resulting reweighted kernels become space-variant)[2].

Nevertheless, to some extent, such sub-optimality can be effectively ignored

[2] This particular type of estimators are also discussed in [22], in connection with the quasi-likelihood.

if the adaptive-scale selection is carefully managed. The extensive experiments presented in the second part of this thesis strongly support such a claim.

Two issues should be addressed before considering the use of the *ICI* rule with an observation model such as (5.3): the first concerns with the calculation of the variances $\sigma^2_{\hat{y}_h}$ of the varying-scale estimates $\hat{y}_h$, the second – more subtle – deals with the distribution of $\hat{y}_h$ (as a random variable with variance $\sigma^2_{\hat{y}_h}$).

The first issue is rather simple and it results in a generalization of equation (2.4).

5.2.1 Variance for heteroskedastic observations

Since the noise is independent, the variance $\sigma^2_{\hat{y}_h(x)}$ of the estimate $\hat{y}_h(x)$,

$$\hat{y}_h(x) = \int z(v)\, g_h(x-v)\, dv = (z \circledast g_h)(x),$$

can be calculated as the convolution of σ^2_z against g^2_h,

$$\sigma^2_{\hat{y}_h(x)} = \int \sigma^2_z(v)\, g^2_h(x-v)\, dv = \left(\sigma^2_z \circledast g^2_h\right)(x). \tag{5.4}$$

Remark: If σ^2_z is constant, or nearly constant, within the support of g_h, the approximation

$$\sigma^2_{\hat{y}_h(x)} = \int \sigma^2_z(v)\, g^2_h(x-v)\, dv \approx \int \sigma^2_z(x)\, g^2_h(x-v)\, dv = \sigma^2_z(x) \int g^2_h(v)\, dv \tag{5.5}$$

can be used to avoid the computation of the convolution (5.4), and thus contain the algorithm's computational complexity.

5.2.2 Variance's asymptotics

If we assume that σ^2_z is continuous at x, and that it is uniformly bounded, from above and from below, by two positive constants, $0 < c \leq \sigma^2_z(x) \leq C < \infty$, then the asymptotic expression for the variance (2.6) is essentially unchanged. Because of the bounds, the exponent β remains the same. From the continuity of σ^2_z follows that the factor b^2 is proportional to $\sigma^2_z(x)$.

Let us now consider the distribution of $\hat{y}_h(x)$.

5.2.3 Confidence intervals for non-Gaussian distributed estimates

The confidence intervals that are intersected in the *ICI* algorithm traditionally have the form

$$\mathcal{D}_j = \left[\hat{y}_{h_j}(x) - \Gamma\sigma_{\hat{y}_{h_j}}, \hat{y}_{h_j}(x) + \Gamma\sigma_{\hat{y}_{h_j}}\right]$$

as they are constructed assuming that the estimates $\hat{y}_{h_j}(x)$ have a Gaussian distribution. When the observations z have a distribution that is not Gaussian the confidence intervals may have, in general, a different expression.

However, unless also higher moments of the distribution are accurately known, it is anyway reasonable to use the Gaussian confidence intervals. This choice is justified by the following fact.

Regardless of the true distribution of $z(x)$, as h increases, the estimates $\hat{y}_h(x)$ are constructed by averaging an increasing number of observations $z(v)$ from a neighborhood of x. A central limit theorem argument can thus take place, and thus the probability distribution of $\hat{y}_h(x)$ gets progressively "*Gaussianised*".

In the particular case of the Poisson observations, when $y(x) = \lambda(x)$ is sufficiently large, the Poisson distribution are already well approximated by the corresponding Gaussian distribution with mean and variance equal to $\lambda(x)$, as shown in Figure 5.1.

5.2.4 Conclusions

We then come to the conclusion that the standard *ICI* algorithm (for Gaussian observations) may be used also for estimates coming from non-Gaussian distributed observations (e.g. Poisson photon counting, speckle, film-grain noise, etc.) in order to obtain a quite accurate selection of the adaptive scale $h^+(x)$. This conclusion is confirmed by many simulation results obtained for a wide range of non-Gaussian and heteroskedastic observations. These results are presented in the second part of the thesis.

However, we should remark that the variance of the estimates should be really calculated according to (5.4). The use of the formula (2.4), which is correct for the AWGN case, would obviously fail in taking into consideration the factor σ_z^2 that multiplies b^2 in the asymptotic expression (2.6). Observe that (5.4) and (5.5) are to be considered equivalent with respect to the asymptotic analysis. Nevertheless, we don't recommend, in general, to use the latter approximated formula in the algorithm implementations, unless σ_z^2 is really known to be quite regular.

Finally, let us note that all common types of signal-dependent noise models involve infinitely smooth variance functions. Therefore, the corresponding σ_z^2 is as smooth as y itself.

Chapter 6

Recursive anisotropic *LPA-ICI*

It was proposed in [20][1] to use the anisotropic *LPA-ICI* estimator iteratively. This recursion results in the anisotropic enlargement of the estimation neighborhood U_x^+, an effect that can be interpreted as a special diffusion process.

In this chapter we discuss mainly the theoretical aspects of this recursive procedure. We leave the implementation aspects for the second part of the thesis, in Section 8.2. Let us anticipate however, that the improvement in the restoration performance achieved by the recursive application of the anisotropic *LPA-ICI* is significant. Therefore, most of the denoising algorithms presented in Part II are actually based on the method that we are about to describe.

6.1 An iterative system

The idea behind the proposed procedure is to apply recursively the anisotropic *LPA-ICI* algorithm, filtering the final output $\hat{y}$ (4.8) once or many times over again.

Denoting by $\mathcal{LI}$ the overall anisotropic *LPA-ICI* filter, this recursion is expressed as follows:

$$\begin{cases} z^{[1]} = z, \\ \hat{y}^{[l]} = \mathcal{LI}(z^{[l]}), \qquad l = 1, 2, \dots \quad . \\ z^{[l+1]} = \hat{y}^{[l]}, \end{cases} \tag{6.1}$$

The square brackets [ ] indicate the iteration number.

Expanding (6.1), in order to explicitly write $\hat{y}^{[l]}$ with respect to the initial

[1] [20]: Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "A novel anisotropic local polynomial estimator based on directional multiscale optimizations", *Proc. 6th IMA Int. Conf. Math. in Signal Processing*, Cirencester (UK), pp. 79-82, December 2004.

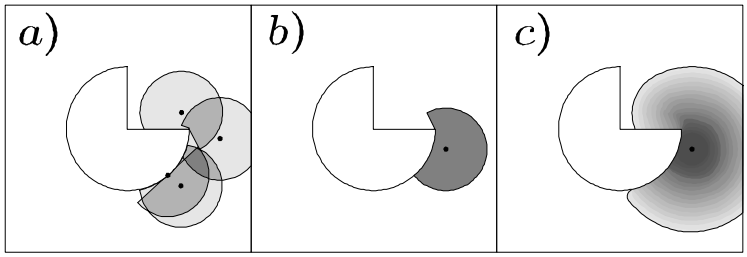


Figure 6.1: Some ideal starshaped neighborhoods $\tilde{U}_v^*$ (a) corresponding to points v belonging to the ideal neighborhood $\tilde{U}_x^*$ of the estimation point x (b) and the resulting *fattened* neighborhood of x, $\bigcup_{v\in\tilde{U}_x^*}\tilde{U}_v^*$ (c), obtained by the second iteration of the adaptive algorithm.

observations z, we obtain

$$\hat{y}^{(l)}(x) = \int g_x^{+[l]}(x-v)\hat{y}^{[l-1]}(v)dv =$$
$$= \int g_x^{+[l]}(x-v)\left(\int g_v^{+[l-1]}(v-w)\hat{y}^{[l-2]}(w)dw\right)dv =$$
$$= \int\left(\int g_x^{+[l]}(x-v)g_v^{+[l-1]}(v-w)dv\right)\hat{y}^{[l-2]}(w)dw = \cdots =$$
$$= \int\left(\int\cdots\int\left(\tilde{g}_x^{+[l]}(v^{[1]})\tilde{g}_{v^{\{1\}}}^{+[l-1]}(v^{[2]})\cdots\tilde{g}_{v^{\{l-1\}}}^{+[1]}(v^{\{l\}})\right)dv^{\{1\}}...dv^{\{l-1\}}\right)z(v^{\{l\}})dv^{\{l\}} \quad (6.2)$$

where $g_x^{+[l]}$ denotes the anisotropic kernel at the l-th iteration, $\tilde{g}_x^{+[l]}(\cdot) = g_x^{+[l]}(x-\cdot)$, and $v^{\{i\}}$ are auxiliary variables.

6.2 Estimation neighborhood's enlargement

Let us consider the simple settings discussed in Section 4.1.1, where the true image is binary and the ideal estimation neighborhood U_x^* corresponds to the best unbiased estimate. As it was observed, the shape and size of U_x^* do not depend on the observed signal z, but rather only on the (unknown) signal y. When a second iteration is performed in (6.1), the ideal neighborhood used for estimating $y(x)$ from $z^{[2]} = \hat{y}^{[1]}$ is again the same U_x^* from the first iteration. Since this applies to all iterations, the whole process is described by replacing all kernels $\tilde{g}_t^{+[\cdot]}$ with $1_{\tilde{U}_t^*}$ in (6.2). Despite the ideal neighborhood U_x^* is always the same for all l, the support of the resulting kernel that is used for integration against $z(v^{\{l\}})$ in the right hand side of (6.2) may grow at every iteration. For example, at the second iteration, the estimation support with respect to the initial observations z is $\operatorname{supp}\int 1_{\tilde{U}_x^*}(v)1_{\tilde{U}_v^*}(\cdot)dv = \bigcup_{v\in\tilde{U}_x^*}\tilde{U}_v^*$. This is illustrated in Figure 6.2(left), with (a) some ideal starshaped neighborhoods $\tilde{U}_v^*$ corresponding to points v belonging to, (b) the ideal neighborhood $\tilde{U}_x^*$ of the estimation point x, and (c) the resulting *enlarged* neighborhood of x, $\bigcup_{v\in\tilde{U}_x^*}\tilde{U}_v^*$, obtained by the second iteration of the adaptive algorithm. Such sets are not necessarily starshaped with respect to x.

If the ideal neighborhoods were translation-invariant, $U_x^* = U^* \; \forall x$, then (6.2) would take the simple convolutional form $\hat{y}^{[l]}(x) = \left(1_{U^*} \circledast \cdots \circledast 1_{U^*} \circledast z\right)(x)$, where convolution between kernels is repeated l-1 times. This resembles other iterative constructions, such as the Gaussian/Laplacian pyramids or wavelet-type projections (e.g. [66]), where multiscale filtering is obtained by recursively convolving the observations against the same filter.

In general, however, formula (6.2) cannot be written in a simple convolutional form, because the adaptive kernels are not translation-invariant. Nevertheless, the considerations previously given about the enlargement of ideal neighborhoods hold similarly for the supports of the fused kernels. This anisotropic propagation of the estimation neighborhoods realizes a diffusion flow similar to the non-linear anisotropic diffusion ([77]), but intrinsically robust to noise because of the *ICI*-based adaptive scale. Regardless of their linear appearance, (6.2), as well as (4.8), are also non-linear estimators. The non-linearity is introduced by the adaptive selection of the directional scale $h^+(x, \theta_k)$.

6.3 Variance of l-th iteration's directional estimates

Let $G^{[l]}_{x,h,\theta_k}(\cdot) = \int\cdots\int (g_{h,\theta_k}(x - v^{\{1\}}) \tilde{g}^{+[l-1]}_{v^{\{1\}}}(v^{\{2\}}) \cdots \tilde{g}^{+[1]}_{v^{\{l-1\}}}(\cdot)) dv^{\{1\}} \ldots dv^{\{l-1\}}$, then, the standard deviation of the estimate $\hat{y}^{[l]}_{h,\theta_k}(x)$, which is needed in order to use the *ICI* rule to select the adaptive scale $h^+(x, \theta_k)$ at the l-th iteration, according to (6.2) and (5.4), is

$$\sigma_{\hat{y}^{[l]}_{h,\theta_k}(x)} = \left(\int \sigma_z^2(v) \left(G^{[l]}_{x,h,\theta_k} \right)^2 dv \right)^{1/2} = \sigma \left\| G^{[l]}_{x,h,\theta_k} \right\|_2, \qquad (6.3)$$

where the equality to the right holds in the case of homoskedastic observations, i.e. when $\sigma_z^2 \equiv \sigma^2$.

Unlike the standard (non-recursive) *LPA-ICI*, even in the homoskedastic case, the variance of the recursive directional varying-scale estimate is space-variant (note the "x" in the right-hand side terms of (6.3)).

The calculation of the standard deviation (6.3) is computationally quite complex, requiring also a good deal of computer memory. These technical reasons limit the direct and accurate implementation of the recursive system (6.1). It would be appealing to use a simpler construction, where each step is performed without keeping track of the previous iterations, i.e. using the $\mathcal{LI}$ operator as a "black box", with a pair of inputs (observations and their standard deviations) and a pair of outputs (estimates and their standard deviations), as shown in Figure 6.2.

Would it be possible to use the variance of the previous fused estimate $\sigma^2_{\hat{y}^{[l-1]}}$, *directly* as an estimate of the variance for the next observation $\hat{\sigma}^2_{z^{[l-1]}}$?

The answer to this question is negative, because, in general, the residual noise in the fused estimate $\hat{y}$ is correlated. This is because the anisotropic estimation neighborhoods may overlap with each other.

In Section 8.2, we present a way to implement efficiently the recursive anisotropic *LPA-ICI*. However, this efficiency is achieved at the cost of a quite approximate computation of the variance. Nevertheless, it appeared that in

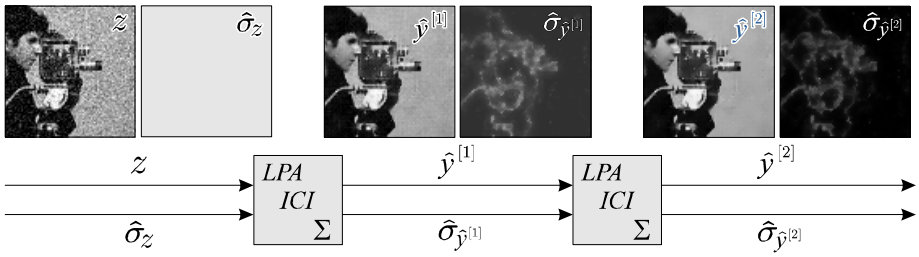


Figure 6.2: Layout of the recursive *LPA-ICI* procedure. At each iteration *LPA* filtering, *ICI* adaptive scale selection (for each direction independently) and fusing are performed. Input data are the pair formed by an observation and its variance map. Output is the pair formed by the estimate and its variance map. Recursion is obtained by feeding the next iteration using the previous estimate as an observation.

spite of the rough estimate of the variance, the *ICI* algorithm is able to perform the adaptive-scale selection.

Chapter 7

Directional derivative estimation and anisotropic gradient

7.1 Derivative estimation

In a number of applications one is interested not in the function itself but rather in the function's derivative. When the data is discrete, the classical way to compute the derivative is by *finite differencing*.

We recall three basic types of incremental ratios:

$$\textit{forward difference} \qquad \frac{\partial^{\text{finite}}\varphi}{\partial^{+}x}(x) = \varphi(x+1) - \varphi(x); \tag{7.1}$$

$$\textit{backward difference} \qquad \frac{\partial^{\text{finite}}\varphi}{\partial^{-}x}(x) = \varphi(x) - \varphi(x-1); \tag{7.2}$$

$$\textit{central difference} \qquad \frac{\partial^{\text{finite}}\varphi}{\partial^{c}x}(x) = \frac{1}{2}\varphi(x+1) - \varphi(x-1). \tag{7.3}$$

These finite differences can be computed also by means of the convolution of φ against a derivative-estimation filtering kernel g. In particular, g is equal to $[1, -1, 0]$, $[0, 1, -1]$, and $[\frac{1}{2}, 0, -\frac{1}{2}]$, for the central, forward, and backward difference, respectively. This is similar to the distributional approach, where the derivative operator is obtained as the convolution with the "derivative of the Dirac delta", that is, as the limit of the convolutions against the derivatives of a sequence of delta-approximating smooth fundamental kernels, so-called Delta-sequences. This interpretation allows a more general approach to the problem of derivative estimation, because it suggests that, in principle, an approximation of the derivative can be obtained by convolution against the derivative of any approximation of a Dirac delta. Gaussian approximating functions (as in the steerable filters [25]) are typically used; however, many other options are possible. For instance, one could restrict his attention to a specific function space, and develop derivative-estimation kernels that are accurate within this specific class of functions. The *LPA* technique allows to design convolutional kernels for an accurate estimation of the derivative for polynomials of any desirable

order m. In particular, as discussed in Section 1.5.2, convolution against such kernels yields the coefficients of the higher-order terms $\frac{-1^k}{k!}x^k$ of the windowed-least-square best fitting polynomial $p(x)$. These coefficients are, similarly as in Taylor expansions, equal to the k-th derivatives $p^{(k)}(0)$ of p evaluated at the center of the *LPA* window.

The forward, backward, and central finite difference kernels are a special case of the more general *LPA* derivative estimation kernels. They are produced for $m = 1$ and a support window with characteristic function $[1, 1, 0]$, $[0, 1, 1]$, and $[1, 1, 1]$, respectively.

7.1.1 Derivative estimation *LPA* kernels

Standard *LPA*

In the one-dimensional case, although there may be many ways to compute or estimate it, there is in fact only one first-order derivative, namely the derivative with respect to the function's argument x, $\frac{\partial}{\partial x}$. When the dimension is higher, $d > 1$, also other derivatives can be computed. As a direct extension of the 1D case, the partial derivatives with respect to the cartesian coordinates $x_1, \ldots, x_d$, $\frac{\partial}{\partial x_i}$, $i = 1, \ldots, d$, can be defined. The gradient ∇, which is the vector composed by all these partial derivatives, is the main subject of Section 7.2.

In the standard – non-directional – *LPA*, the partial derivatives $\frac{\partial}{\partial x_i}$ are estimated with kernels obtained for some $m_i \geq 1$. The estimated value is the coefficient of the x_i term of the best fitting polynomial. Note that, in the discrete domain, the length of the window along the x_i-axis h_i has to be greater than 1, to allow the design of these kernels.

Only partial derivatives with respect to the cartesian variables can be directly estimated using the standard-*LPA* kernels; the directional derivative along a non-cartesian direction is then computed, indirectly, as a linear combination of the partial derivatives. However, such estimated value is questionable unless the underlying unknown signal is differentiable in a large enough circular neighborhood (ball) of the estimation point.

Directional *LPA*

The directional *LPA* offers a different approach to the derivative estimation problem. Since we are concerned mostly in imaging, $d = 2$ is assumed for the sake of simplicity throughout the following sections. However, a generalization to higher dimensions is straightforward.

If $m \geq (1, 0)$ and $o = (1, 0)$ then $g^{(o)}_{h,\theta}$ are kernels that estimate the partial derivative with respect to the first cartesian coordinate u_1 of the rotated (moving) axes. The rotation is through an angle θ. It means that $g^{(1,0)}_{h,\theta}$ estimates the directional derivative ∂_θ with respect to the direction θ. By varying θ, directional derivatives can be estimated for *any* direction, provided that the corresponding kernels are constructed. In the presence of noise, even when the underlying signal is differentiable, the estimated directional derivative may differ from the usual combination of the partial derivatives,

$$\hat{\partial}_\theta \neq \cos\theta\hat{\partial}_0 + \sin\theta\hat{\partial}_{\frac{\pi}{2}} = \cos\theta\frac{\hat{\partial}}{\partial x_1} + \sin\theta\frac{\hat{\partial}}{\partial x_2}.$$

This difference goes to zero as $h \to \infty$ if the underlying signal is differentiable at the estimation point and is accurately polynomial with the required order on the support window $w_{h,\theta}$ for all θ. In practice, this condition is never satisfied. Instead, to different directions θ correspond different largest scales for which the polynomial assumption is valid within some tolerance factor of the variance. This justifies both the use of the adaptive scales $h^+ = h^+(\cdot, \theta)$, and the independent estimation of the directional derivatives using kernels $g^{(1,0)}_{h,\theta}$ instead of the pair $\left\{g^{(1,0)}_{h,0}, g^{(1,0)}_{h,\frac{\pi}{2}}\right\}$. Indeed, to design first derivative estimation kernels it is enough to have a support of two pixels. However, just as for function estimation kernels, there is a number of reasons why larger supports may be needed. The variance/bias tradeoff plays again a central role. Also the directionality of the support window is an important factor when directional derivatives are estimated.

Illustration

As an illustration of the effect of directional derivative filters, we show the horizontal ($\theta = 0$), diagonal ($\theta = \frac{\pi}{4}$), and vertical ($\theta = \frac{\pi}{2}$) derivatives estimated for two different images using two different sets of kernels.

The first set is formed by symmetric-window kernels of length 3 and width 1. For the horizontal direction they are equivalent to the central finite difference. The second set is constructed with a larger support of length 11 and width 3. The weights of the supporting windows are uniform for both sets.

Their behaviour can be consider as characteristic of small-scale and large-scale kernels, respectively. In particular, it can be seen clearly from the pictures that much of the detail information is lost when large kernels are used.

One-handed derivatives

Typically, the windows that are used for the directional-*LPA* kernel design are asymmetric with respect to the estimation point. More precisely, they are elongated along the positive semiaxis of direction θ. However, due to the rotation of the kernel support in the convolution, the kernels $g^{(1,0)}_{h,\theta}$ estimate the directional left-hand derivative $\partial_{-\theta}$, accordingly to the definition (3.2). Roughly speaking, they are a generalization of the backward finite difference.

Once more, we stress that the use of asymmetric supports is important since it allows to achieve a good edge adaptation. This is consistent with the continuous case, where one-sided derivatives may exist also at points of discontinuity, such as edges.

Right-hand derivatives $\partial_{+\theta}$ are obtained, after a change of sign, from the kernel $g^{(1,0)}_{h,\theta+\pi}$, which is specular to $g^{(1,0)}_{h,\theta}$.

Let us consider again the finite differences (7.1-7.3). It is easy to see that the central finite difference can be obtained as the average of forward and backward finite differences:

$$\frac{\partial^{\text{finite}}\varphi}{\partial^{c}x}(x) = \frac{\partial^{\text{finite}}\varphi}{\partial^{+}x}(x) + \frac{\partial^{\text{finite}}\varphi}{\partial^{-}x}(x).$$

When the more general, adaptive-scale directional-*LPA* derivative estimates are exploited, a more robust, weighted average can be used. To be precise, we

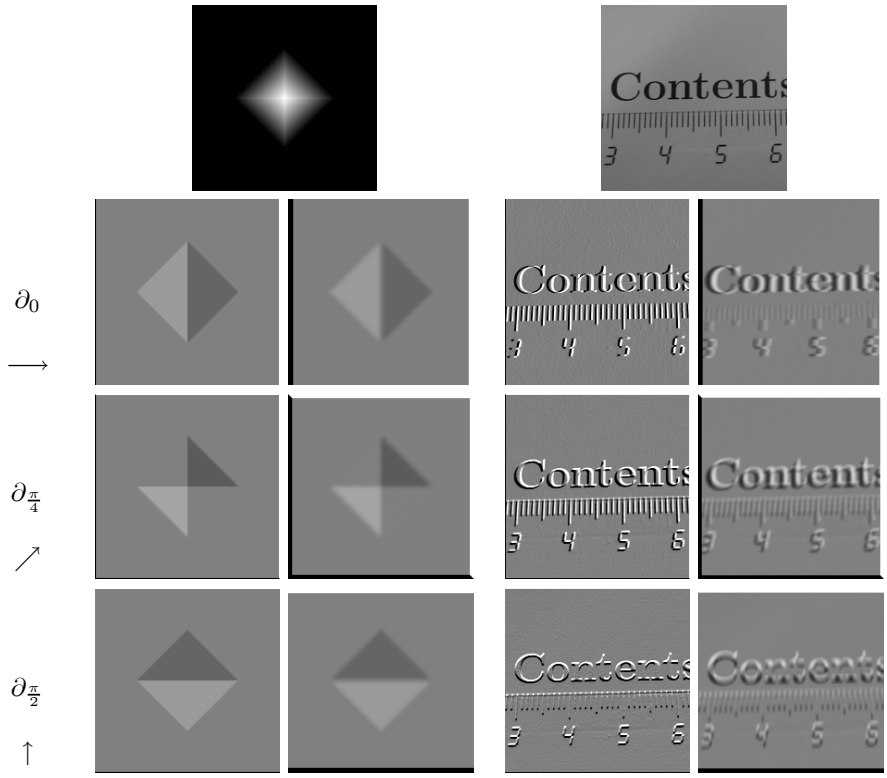


Figure 7.1: Filtering with directional derivative kernels: horizontal, diagonal and vertical derivatives are estimated using kernels of small scale (first and third columns) and large scale (second and fourth columns).

construct an estimate of the directional derivative $\hat{\partial}_\theta$ using the convex combination of one-handed derivatives

$$\hat{\partial}_\theta \triangleq \lambda_+ \hat{\partial}_{+\theta} + \lambda_- \hat{\partial}_{-\theta}, \tag{7.4}$$

where $\hat{\partial}_{+\theta} = -\hat{\partial}_{-(\theta+\pi)}$, and the weights λ_+ and λ_- are defined – similarly to (4.8) – as the normalized (so that $\lambda_+ + \lambda_- = 1$) inverses of the variances of the corresponding estimates,

$$\lambda_- = \frac{\sigma_\theta^{-2}}{\sigma_\theta^{-2} + \sigma_{\theta+\pi}^{-2}}, \qquad \lambda_+ = \frac{\sigma_{\theta+\pi}^{-2}}{\sigma_\theta^{-2} + \sigma_{\theta+\pi}^{-2}}.$$

Fusing of adaptive one-handed derivative estimates: an example

Figures 7.2-7.5 illustrate the clear advantage arising by the combined use of adaptive-scale asymmetrical derivative estimators. A two-dimensional "pyramidal" signal is given (see Figure 7.2 top-left) and some Gaussian white noise of relatively low variance is added to it (top-right). Despite the weakness of the noise, the estimate of the signal's partial derivative $\frac{\partial}{\partial x_1}$obtained by a kernel of fixed-length equal to 3 (i.e. central differencing) is visibly compromised. Increasing the kernel length helps in the reduction of the influence of the noise but

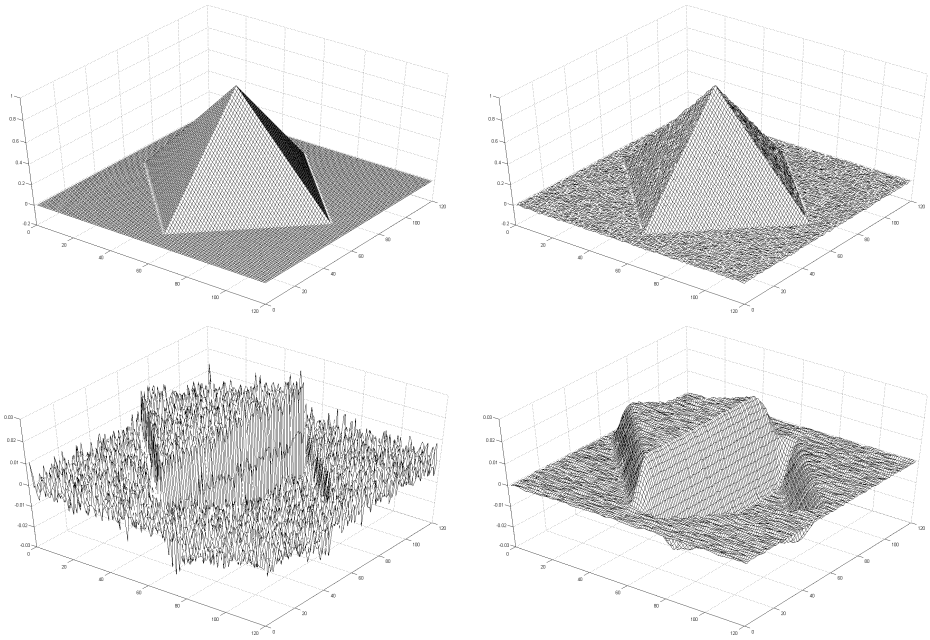


Figure 7.2: Non-adaptive derivative estimation. The original pyramidal signal (top left) is contaminated by some weak additive noise (top right). If a short derivative estimation kernel of length 3 is used to compute the horizontal partial derivative $\frac{\partial}{\partial x_1}$ the result is extremely noisy (bottom left). A larger kernel support helps reduce the noise, however it significatly blurs the sharp transitions in derivative values (bottom right).

at the same time the estimate loses its sharpness. In other words, the variance of the estimate is decreased but the estimate became sensibly biased. These two fixed-length estimates are shown in the bottom row of Figure 7.2.

Figure 7.3 shows the one-handed derivative estimates (top row) obtained using adaptive-length kernels and the corresponding adaptive lenghts (bottom row). Adaptivity is achieved by the *ICI* rule. The adaptive scales are quite noisy, nevertheless their qualitative behaviour is clearly showing: large scales (up to a fixed maximum value) are adaptively selected far from the change-points of the derivative, and the scale decreases when moving – in the direction of estimation – towards the change point. The qualitative behaviour of the adaptive scales that correspond to the left-hand derivative is specular to the one of those corresponding to the right-hand derivative. It is interesting to observe that the two estimates of the one-handed derivatives are still rather noisy, and their quality degradates towards the change-points of the derivative value.

The fused estimate of the derivative, according to formula (7.4), is shown in Figure 7.4. It is quite clean from noise, and present sharp, well-defined transitions at the change-points. The cross-sections in Figure 7.5 provide a visibly-clear comparison of the quality of the estimates.

In this example, one-dimensional "line-wise" kernels have been used, for the fixed-length (non-adaptive) as well as for the adaptive method. It means that

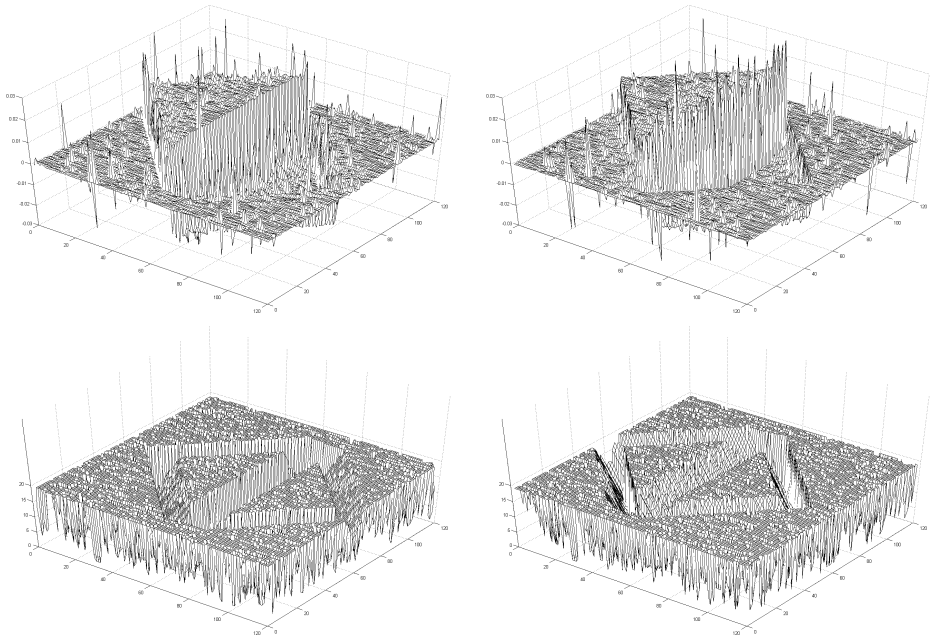


Figure 7.3: Adaptive one-handed derivative estimation: $\frac{\partial}{\partial x_1^-}$ and $\frac{\partial}{\partial x_1^+}$ (top); the corresponding adaptive scales h^+ are shown in the bottom.

the correlation of the underlying data across the orientation of estimation has *not* been exploited. The widely-used 3×3 Prewitt and Sobel kernels (e.g. [37]) and the steerable filters [25] use this possible correlation as the key element to achieve an improved estimate. However, in the vicinity of change-points, such correlation may fail.

In what follows, we propose a novel approach where a robust estimate of any directional derivative is obtained exploiting the data-correlation only when it is present. The estimation of the derivatives, and the estimation of the region of estimation are performed *simultaneously*.

7.2 Anisotropic gradient estimation

Although the adaptivity of the asymmetric estimates and the fusing allow to achieve a remarkable improvement, one should observe that so far, no speculations of geometrical kind (i.e. anisotropic neighborhood) have been exploited. Indeed, without any differentiability assumption, the directional derivatives are, in general, unrelated on to the other.

Traditionally, such differentiability assumption is used to allow a representation of the generic directional derivative ∂_θ in the form

$$\partial_\theta = \cos\theta \partial_{x_1} + \sin\theta \partial_{x_2},$$

where only the two partial derivatives, i.e. the gradient $\nabla = (\partial_{x_1}, \partial_{x_2}) = \left(\partial_0, \partial_{\pi/2}\right)$, are actually estimated from convolution against the data.

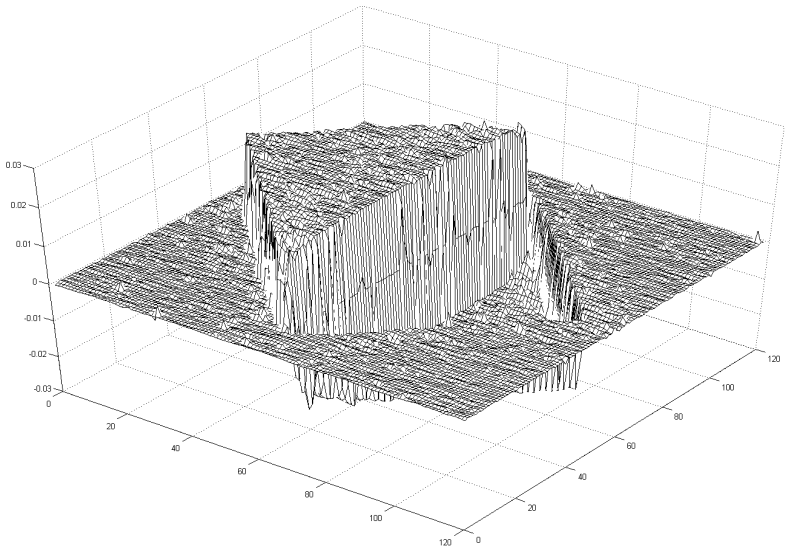


Figure 7.4: Estimate of the derivative obtained by fusing, with adaptive weights, the left- and right-hand estimates shown in Figure 7.3.

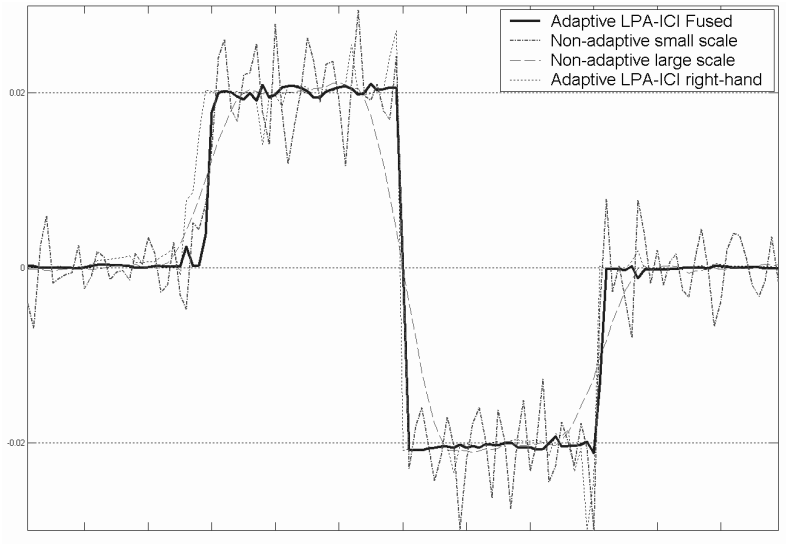


Figure 7.5: Cross-section of one line of the horizontal derivative $\frac{\partial}{\partial x}$ estimated by convolution with non-adaptive (small scale and large scale) and adaptive (single and fused) kernels.

In this section we introduce a generalization of the concept of differentiability. In contrast with the above formula, where the estimated gradient is used in order to obtain, by projection, any directional derivative, we will exploit many independently-estimated directional derivatives to achieve a more robust and more general estimate of the gradient itself.

7.2.1 Motivation

In the continuous domain, a function f is said to be differentiable at a point x if there exist a linear functional $\mathcal{L}_x$ such that

$$f(x+v) - f(x) - \mathcal{L}_x(v) = o(|v|). \tag{7.5}$$

If such functional $\mathcal{L}_x$ exists, then it has the following expression

$$\mathcal{L}_x(v) = \nabla f(x)\, v^T, \tag{7.6}$$

where $\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)$ is the gradient vector of the function f.

When the function is differentiable, a lot of information about its local behaviour in a vicinity of x is expressed by this approximating linear functional $\mathcal{L}_x$. Most notably the direction of steepest ascent for f at x is equal the direction of the gradient vector $\nabla f(x)$ and the rate of ascent is given by the modulus $|\nabla f(x)|$. If the graph of the function f is thought as a surface in the 3D space, then $\nu_f(x) = \left(-\frac{\partial f}{\partial x_1}(x), -\frac{\partial f}{\partial x_2}(x), 1\right)$ is the normal vector to the surface (i.e. the normal vector to the tangent plane).

Such information is often of critical importance in many theoretical problems (ordinary and partial differential equations, optimization theory, algorithm convergence) and practical applications (geophysics, imaging, computer graphics). The aforementioned properties of the gradient *do not* hold, in general, when the function is not differentiable and no functional $\mathcal{L}_x$ satisfying (7.5) exists. Indeed, checking the differentiability hypothesis can be itself an extremely difficult - if not impossible - task, especially when the function f is known only partially (sampling) or with limited precision (quantization, noise). Nevertheless in practical applications the differentiability hypothesis is often given implicitly for granted and formula (7.6) is used to derive a functional that may not satisfy (7.5). Although in some applications this incongruence can be the key to successful results (the most striking example is probably is the edge detection problem: the gradient is used to detect edges, which are clearly points of non-differentiability), in others it can be a limitation since the information obtained from the partial derivatives $\frac{\partial f}{\partial x_1}$, $\frac{\partial f}{\partial x_2}$ can be not representative at all of the local behaviour of f. An example of such applications is the Z-buffer shading, in which the angle between the normal vector ν and the illuminant vector is used to evaluate the correct intensity value for realistic 3D rendering. A few simulation results relative to this shading application are presented in Section 10.2.

Traditional approaches for the estimation of the gradient are based on some surface-fitting of the observation. After the estimated surface is fitted, its normal vector at x is taken from the analytic expression of the surface. Although this technique produces very good results when the underlying data is smooth, particular care must be taken in the vicinity of edges, where surface fitting may

be unsuitable. More sofisticated approaches (such as [100]) first segment the image into regions of smoothness, and then evaluate the normal within these regions using conventional surface-fitting or kernel-estimates. Other authors (e.g. [5], [85]) estimate the gradient by using both forward and backward differences. Edges are treated in a similar spirit as for edge detection, exploiting thresholding schemes on the magnitude of the gradient and/or on the discrepancy between forward and backward estimates. Based on this thresholding, edges or change-points are identified, and are then excluded from the further calculations. Such schemes are quite complicated and their robustness to noise is not clear.

In the present section we propose a different notion of gradient. We call it the *anisotropic gradient*.

A key aspect of the proposed approach is that the differentiability hypothesis is not assumed. This novel gradient can be defined, even in its analytical form, even at points where the function is not differentiable or discontinuous.

Nevertheless, it can be thought as an extension of the concept of gradient, in the sense of equation (7.6). That is because the traditional gradient and the proposed anisotropic gradient coincide whenever the function is differentiable. However, when the function is not differentiable the anisotropic gradient provides information which is more faithful to the local behaviour of the function in the sense of equation (7.5), where *local* means that the function is considered as restricted to an anisotropic neighborhood of regularity. The method is also instrinsically *multiscale*, since for different directions an ideal scale is selected. In practice this will be accomplished by means of the *ICI* rule.

It allows unlimited directional and scale resolution, as well as any order of polynomial approximation.

To facilitate the reader in understanding the main ideas, we begin from an illustratory example, which is followed in detail throughout the section, so to highlight the various peculiarities of the proposed method.

The exposition is organized as follows:

- an example is given and discussed, first in its analytical form in the continuous domain, and then in the discrete domain in the presence of noise;
- the anisotropic gradient is defined in the continuous domain;
- the anisotropic function estimation based on the directional *LPA-ICI* approach is reviewed, and then generalized, leading to the notion of discrete anisotropic gradient;
- further examples of anisotropic gradient estimation are shown.

7.2.2 An illustrative example in the continuous domain

Let us consider the real function $\varphi : \left[-\frac{1}{2}, \frac{1}{2}\right] \times \left[-\frac{1}{2}, \frac{1}{2}\right] \to \mathbb{R}$, defined as

$$\varphi(x_1, x_2) = \begin{cases} 0 & \text{for } r \leq \frac{1}{10} \\ \text{angle}(x_1 + ix_2) & \text{for } r > \frac{1}{10} \end{cases}, \tag{7.7}$$

where, i is the imaginary unity, "angle" is the function returning the angular component of a complex number, (i.e. $\text{angle}\left(\rho e^{i\theta}\right) = \theta$ for $\theta \in (-\pi, \pi]$, $\rho > 0$),

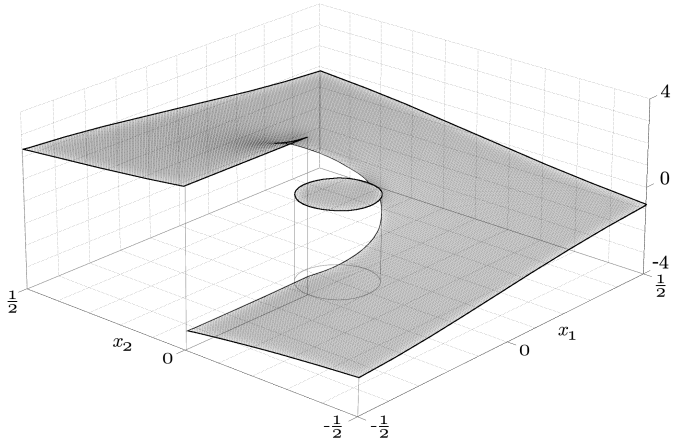


Figure 7.6: The function $\varphi(x_1, x_2)$.

and $r = r(x_1, x_2) = \sqrt{x_1^2 + x_2^2}$. An illustration of this function is given in Figure 7.6.

The function φ is smooth at and only at all points (x_1, x_2) such that $r < \frac{1}{10}$ or $r > \frac{1}{10}$ and $x_2 \neq 0$. Hence, the set of non-differentiability is

$$E = \left\{(x_1, x_2) : r = \tfrac{1}{10}\right\} \cup \left\{(x_1, 0) : x_1 < -\tfrac{1}{10}\right\}.$$

E, shown in Figure 7.7(left), is a nowhere dense set of zero Lebesgue measure and its complementary (the set where the function is smooth) is a dense subset of $\left[-\frac{1}{2}, \frac{1}{2}\right] \times \left[-\frac{1}{2}, \frac{1}{2}\right]$. The point $\left(\frac{1}{10}, 0\right)$ is somehow special, since although it belongs to E, the function is there nevertheless continuous and both partial derivatives, $\frac{\partial \varphi}{\partial x_1}$ and $\frac{\partial \varphi}{\partial x_2}$, exist.

Let us consider now the gradient $\nabla\varphi = \left(\frac{\partial \varphi}{\partial x_1}, \frac{\partial \varphi}{\partial x_2}\right)$. It has the form

$$\nabla\varphi = \begin{cases} (\,0\,,\,0\,) & \text{for } r < \frac{1}{10}, \\ \left(-\frac{x_2}{x_1^2+x_2^2}, \frac{x_1}{x_1^2+x_2^2}\right) & \text{for } r > \frac{1}{10} \text{ and } x_2 \neq 0, \text{ or } x_2 = 0 \text{ and } x_1 \geq \frac{1}{10}. \end{cases} \tag{7.8}$$

The gradient is not defined on most of the set of non-differentiability E. Nevertheless, for every point $x = (x_1, x_2) \in E$ there exist an *anisotropic neighborhood* U_x such that $\varphi_{|U_x}$ is smooth at x. In other words, by restricting to a particular neighborhood of the point x, we are able to find an approximating plane to the surface of the (restricted) function. We can define an "extended gradient" $\overline{\nabla}\varphi$ defined on the whole domain of φ. Such gradient $\overline{\nabla}\varphi$, which we call the *anisotropic gradient of* φ, has the form

$$\overline{\nabla}\varphi = \begin{cases} (\,0\,,\,0\,) & \text{for } r \leq \frac{1}{10} \\ \left(-\frac{x_2}{x_1^2+x_2^2}, \frac{x_1}{x_1^2+x_2^2}\right) & \text{for } r > \frac{1}{10} \end{cases}. \tag{7.9}$$

The precise definition of $\overline{\nabla}$ is given later on. However, we anticipate that its main property is that $\overline{\nabla}\varphi(x)\, v^T = \mathcal{L}_x^U(v)$, where $\mathcal{L}_x^U(v)$ is a linear functional that, similarly to the one in equation (7.5), approximates $\varphi_{|U_x}$ (i.e. φ restricted to U_x) with $o(|v|)$ precision.

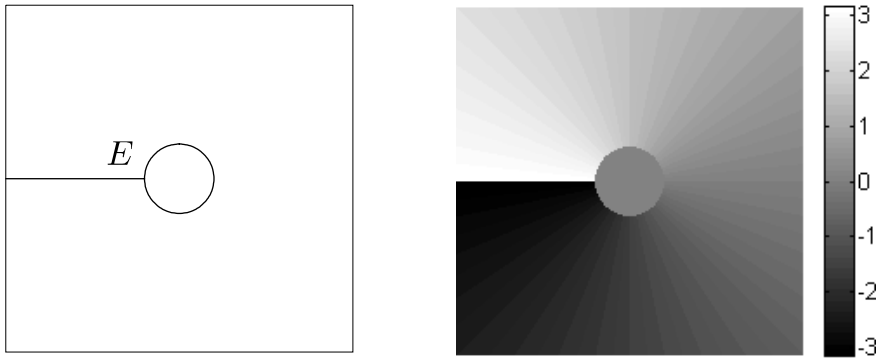


Figure 7.7: The set of non-differentiability E (left) and the function φ sampled on a discrete 200×200 grid (right).

For this particular example, it is interesting to note that the original gradient $\nabla\varphi$ itself can be smoothly extended to a smooth vector function $\widetilde{\nabla\varphi}$, defined for $r \neq \frac{1}{10}$, which satisfies

$$\widetilde{\nabla\varphi} = \overline{\nabla}\varphi \text{ for all } (x_1, x_2) \text{ such that } r \neq \tfrac{1}{10}.$$

This vector function $\widetilde{\nabla\varphi}$ can be found in most mathematical analysis textbooks as $\omega = \widetilde{\nabla\varphi}\,(dx_1, dx_2)^T$. It is the classical example of a closed differential form that it is not exact, i.e. that does not admit a potential. In fact, the function φ can be interpreted as some real counterpart of a complex Riemann surface of logarithmic type (whose domain is not a simply connected set).

Note that $\overline{\nabla}\varphi$ is not defined univocally, since different choices of $U_{(x_1,x_2)}$ may lead to a different $\overline{\nabla}\varphi\,(x_1, x_2)$.

However, as shown in the following sections, $\overline{\nabla}\varphi$ is uniquely defined for all points but $\left(\frac{1}{10}, 0\right)$. The value indicated in (7.9) for this point is not uniquely determined.

Although there are some differences between the "gradients" $\nabla\varphi$, $\overline{\nabla}\varphi$, and $\widetilde{\nabla\varphi}$, these differences are not too essential because E is a nowhere dense set of measure zero. Nevertheless, when we move our attention to the discrete case, differences get more evident. In particular, we will see that the discrete anisotropic gradient will be much closer to the analytical (continuous domain) gradients than the traditional gradient estimated with standard derivative estimation kernels.

7.2.3 The same example in the discrete domain

We consider now the discrete version of the above example, where function φ is sampled on a 200×200 grid (see Figure 7.7(right)). As we consider the sampling rate to be equal to one, all derivative values (shown in the figures) should be multiplied by 200 to obtain results comparable to the above continuous domain example. A weak Gaussian additive noise ($\sigma = 0.01$) was added to the observations.

Figure 7.8 shows the true *ideal* gradient $\nabla\varphi$ obtained analytically from the analytical expression of φ, as in formula (7.8) of the previous section, sampled on

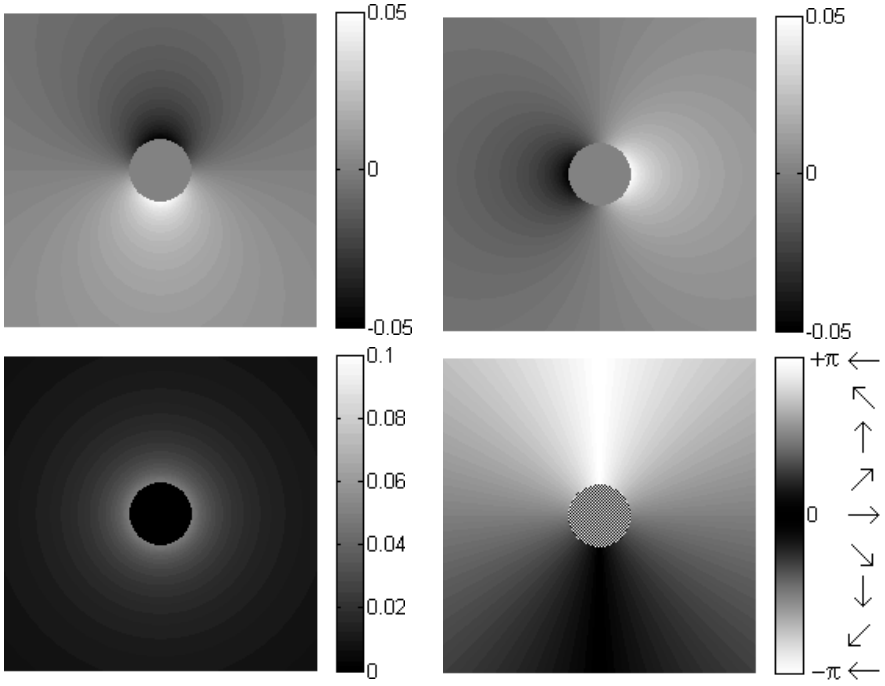


Figure 7.8: True *ideal* gradient: $\frac{\partial \varphi}{\partial x_1}$, $\frac{\partial \varphi}{\partial x_2}$ (above), $|\nabla\varphi|$ and angle $\left(\nabla\varphi\,(1,i)^T\right)$ (bottom). Note that the latter is not defined when $|\nabla\varphi| = 0$.

the same discrete grid. The top two subimages show, respectively, $\frac{\partial \varphi}{\partial x_1}$ and $\frac{\partial \varphi}{\partial x_2}$, whereas the two bottom subimages show $|\nabla\varphi|$, and angle $\left(\nabla\varphi\,(1,i)^T\right)$. Note that the angular component of gradient cannot be defined where $|\nabla\varphi| = 0$.

In Figure 7.9 and 7.10 we present the gradient estimated by standard convolution against derivative kernels. For this standard approach two separate sets of kernels were respectively used: fine scale (length=3) and large scale (length=7). It can be seen clearly that the discontinuities in the image affect sensibly the estimated gradient, in particular, as the scale increases, such unwanted features become more marked.

The gradient estimated by the proposed discrete anisotropic gradient method is shown in Figure 7.11. An adaptive-scale varying between 2 and 6 is used. The similarity between the discrete anisotropic gradient and the (sampled) *analytically computed ideal* gradient is evident.

Symmetric derivative kernels of length 3 and 7 were used for the small-scale, and for the large-scale "traditional" gradient estimation examples, respectively. For the anisotropic case we used asymmetric kernels of length varying between 2 and 6. Figure 7.12 shows the largest scale kernels used for the estimation of $\frac{\partial \varphi}{\partial x_1}$ and $\frac{\partial \varphi}{\partial^+ x_1}$ in the standard, and anisotropic approach, respectively. Because the central value of symmetric kernels is always zero, the number of non-zero taps in the two kernels is the same.

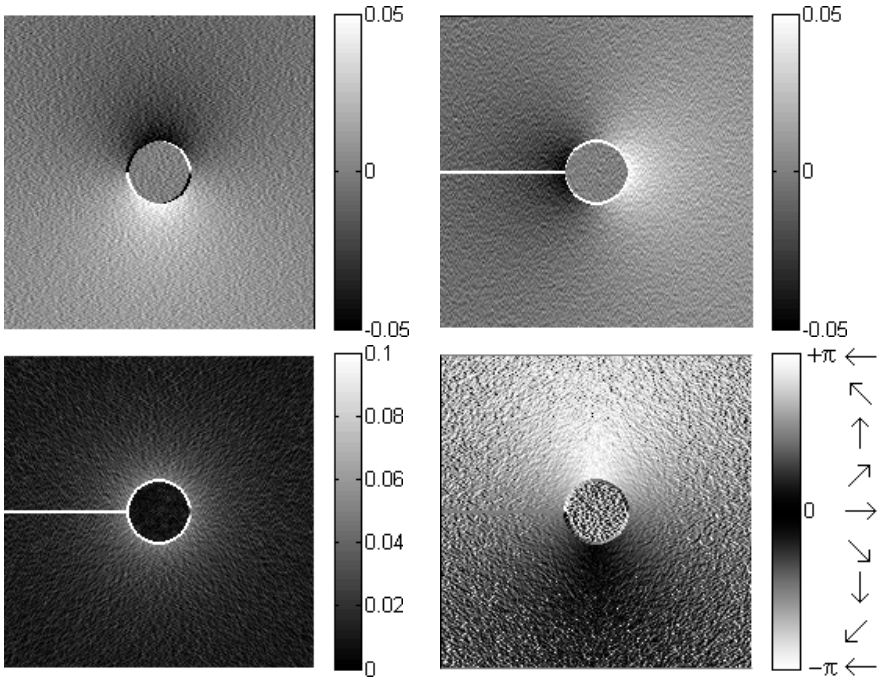


Figure 7.9: Gradient estimated using standard derivative estimation kernels of lenght 3: $\frac{\partial \varphi}{\partial x_1}$, $\frac{\partial \varphi}{\partial x_2}$ (above), $|\nabla \varphi|$ and angle $\left(\nabla \varphi\,(1,i)^T\right)$ (bottom).

Comments

People from the signal processing community could find this example quite unusual and maybe surprising. In fact, we present a technique for gradient/derivative estimation that is, in some way, *not sensible* to the edges in the image. This – at least from an heuristic point of view – contradicts the common practice of using derivatives to actually find the edges. Two facts should be clarified.

First, the very notion of derivative across an edge is objectable. We refuse it, as the edge introduces a discontinuity which prevents the derivability of the function. We choose instead one-handed derivatives, which are intrinsically compatible with edges.

Second, within our approach, edges are implicitly detected by the *ICI* algorithm. Similarly to standard techniques, *LPA*-derivative values across them are large, as shown in Figure 7.13. However, these values are not considered for the estimation of the overall gradient, because at that particular point the function is identified as non-derivable in any direction, along which the edge is crossed. We stress that the method is *not* based on a threshold on the derivative magnitude, but instead on a notion of coherence of varying-scale derivative estimates. This aspect is closely connected with the analytical continuous-domain notion of derivability.

No prior knowledge of the signal structure is assumed, as the method implicitly performs a local analysis of the neighborhood of the estimation point,

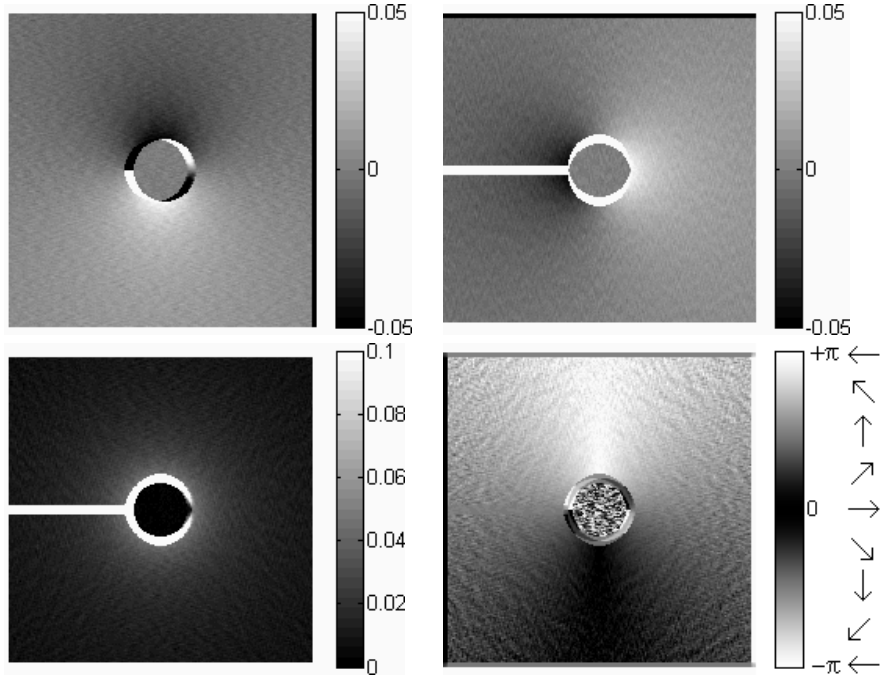


Figure 7.10: Gradient estimated using standard derivative estimation kernels of length 7: $\frac{\partial\varphi}{\partial x_1}$, $\frac{\partial\varphi}{\partial x_2}$ (above), $|\nabla\varphi|$ and angle $\left(\nabla\varphi\,(1,i)^T\right)$ (bottom).

selecting the directions where the differentiability assumptions can hold[1]. Consequently, the estimation is naturally robust to noise.

7.2.4 Continuous domain anisotropic gradient

Let f be a bivariate real function, $\Theta = \{\boldsymbol{\theta}_k\}_k \subset S_1$ a family of versors and h a positive scalar. For every θ_k, we can approximate f along the direction θ_k with a first order expression and write

$$f\,(x + h\theta_k) = f\,(x) + hL_{\theta_k} + e_{\theta_k}\,(h) \qquad h \geq 0$$

where $L_{\theta_k} \in \mathbb{R}$ is a constant, and e_{θ_i} is the approximation error. Note that

$$e_{\theta_k}\,(h) = o\,(h) \iff \exists \partial_{+\theta_k} f\,(x) = L_{\theta_k}.$$

If $\exists \frac{\partial f}{\partial^+ \theta_k}\,(x)$ and $\frac{\partial f}{\partial^+ \theta_k}\,(x) \neq L_{\theta_k}$, then $e_{\theta_k}\,(h) \sim h$. In the following, we always assume that $L_{\theta_k} = \frac{\partial f}{\partial^+ \theta_k}\,(x)$ whenever the right-hand directional derivative $\partial_{+\theta_k} f\,(x)$ exists. We remark that if f is differentiable at x, then $\partial_{+\theta_k} f\,(x)$ exists for all θ_k, coincides with $\partial_{\theta_k} f\,(x)$, and

$$\nabla f\,(x) \begin{pmatrix} \cos\,(\theta_k) \\ \sin\,(\theta_k) \end{pmatrix} = \cos\,(\theta_k)\,\frac{\partial f}{\partial x_k}\,(x) + \sin\,(\theta_k)\,\frac{\partial f}{\partial x_2}\,(x) = \partial_{\theta_k} f\,(x) \qquad (= L_{\theta_k})\,.$$

[1] Observe that, more visibly when the scale gets larger, the traditional derivatives present significant artifacts close to the border of the image. This is due to the imposed boundary conditions, set to $-\frac{1}{\varepsilon}$ (compare with the footnote on page 11).

The discrete anisotropic gradient does not show any artifacts, thanks to its anisotropic adaptation.

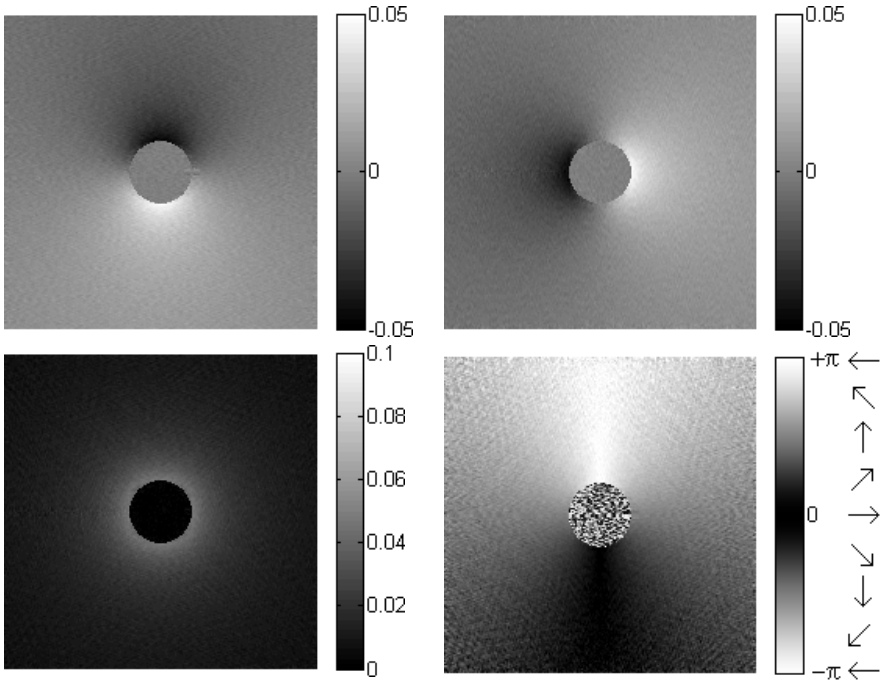


Figure 7.11: Estimated discrete anisotropic gradient: $\frac{\partial \varphi}{\partial x_1}$, $\frac{\partial \varphi}{\partial x_2}$ (above), $|\nabla \varphi|$ and angle $\left(\nabla \varphi \left(1, i\right)^T\right)$ (bottom).

Let w be a continuous increasing function, $w\left(0\right) = 0$, and $h^{\max}$ a positive constant. For the sake of simplicity one may take w the identity function and $h^{\max} = 1$.

For any given $\varepsilon > 0$ let $h_{\theta_k}\left(\varepsilon\right)$ be defined as

$$h_{\theta_k}\left(\varepsilon\right) \triangleq \begin{cases} \min\left(h^{\max}, \sup\left\{h : \left|e_{\theta_k}\right| < \varepsilon\right\}\right) & \text{if } \exists \partial_{+\theta_k} f\left(x\right) \\ 0 & \text{if } \nexists \frac{\partial f}{\partial^+ \theta_k}\left(x\right) \end{cases} . \tag{7.10}$$

By definition, h_{θ_k} is a bounded increasing function and, if $\exists \frac{\partial f}{\partial^+ \theta_k}\left(x\right)$, definitevely as $\varepsilon \to 0$, $h_{\theta_k}\left(\varepsilon\right) > \min\left(h^{\max}, \varepsilon\right) > 0$.

We define the *fixed-scale anisotropic gradient* of f as

$$\nabla_\varepsilon f \triangleq \underset{(M,N)}{\operatorname{argmin}} \sum_k w\left(h_{\theta_k}\left(\varepsilon\right)\right) \left(\frac{\partial f}{\partial^+ \theta_k}\left(x\right) - \cos\left(\theta_k\right) M - \sin\left(\theta_k\right) N\right)^2 . \tag{7.11}$$

This solution is well-posed if $\frac{\partial f}{\partial^+ \theta_k}$ exists for at least two linearly independent θ_k.

In other words, $\nabla_\varepsilon f$ is the solution of a weighted least-square minimization problem where the residuals are the differences between the projections of a "candidate gradient" along the direction θ_k and the corresponding directional derivative.

Observe that if f is differentiable at x, then these residuals can be all put to zero by choosing $(M, N) = \left(\frac{\partial f}{\partial x_1}\left(x\right), \frac{\partial f}{\partial x_2}\left(x\right)\right) = \nabla f\left(x\right)$, and since all $h_{\theta_k}\left(\varepsilon\right)$

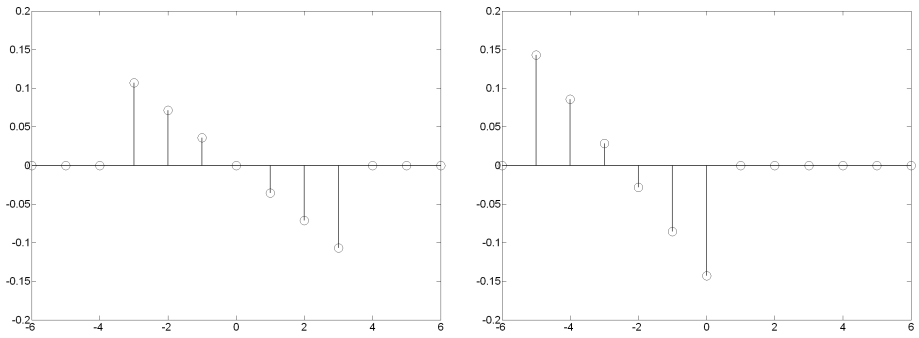


Figure 7.12: "Standard" symmetric derivative kernel of length 7 (left) and "anisotropic" asymmetric right-hand derivative kernel of length 6 (right).

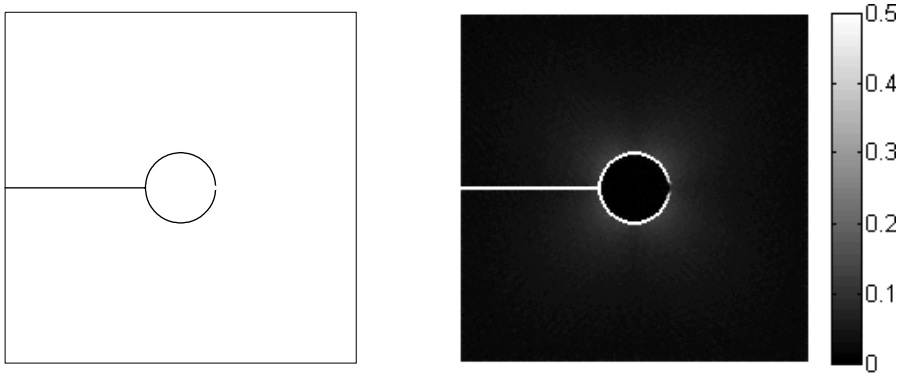


Figure 7.13: Edges of the function φ (left) and the sum $\sum_i \left|\frac{\partial}{\partial^+\theta_i, h^+}\varphi\right|$, where these edges are detected (right).

are strictly positive and bounded, it follows that

$$\nabla_\varepsilon f(x) = \nabla f(x). \tag{7.12}$$

Provided that the following limit exists, we define the *(asymptotic) anisotropic gradient* of f as

$$\overline{\nabla} f(x) \triangleq \lim_{\varepsilon \to 0} \nabla_\varepsilon f(x). \tag{7.13}$$

It follows from (7.12) that, if f is differentiable at x, then $\overline{\nabla} f(x) = \nabla f(x)$. In other words, differentiability implies that the usual gradient, the fixed-scale anisotropic gradient, and the (asymptotic) anisotropic gradient, coincide.

When f is not differentiable, then the anisotropic gradient may be not uniquely defined, and its value depends on the set Θ.

However it is uniquely defined in the following case.

Proposition: Let U_x be an (anisotropic) sectorial neighborhood of x and let $f|_{U_x}$ be differentiable at x. Assume that the following two conditions are satisfied:

- at least two[2] linearly independent versors $\theta_k \in \Theta$, $k \in \left\{\bar{k}_i\right\}_{i=1,2,\ldots}$, lie in

[2] For $d > 2$, at least d linearly independent versors.

the positive linear span of U_x;

- f in *not* derivable along all other versors $\theta_k \in \Theta$, $k \notin \left\{\bar{k}_i\right\}_{i=1,2,\ldots}$.

Then the anisotropic gradient $\overline{\nabla} f(x)$ of f at x is well (uniquely) defined and coincides with the standard gradient $\nabla f|_{U_x}(x)$ of f restricted to the sectorial neighborhood U_x.

The proof is straightforward. First, we observe that the differentiability in U_x implies that $\frac{\partial f}{\partial^+ \theta_k}(x)$ exist for all $k \in \left\{\bar{k}_i\right\}_{i=1,2,\ldots}$. Moreover, it implies also that these directional derivatives are "coplanar", i.e. there exists a pair $M, N \in \mathbb{R}$ such that they all can be obtained as

$$\frac{\partial f}{\partial^+ \theta_k}(x) = \cos(\theta_k) M + \sin(\theta_k) N \qquad \forall k \in \left\{\bar{k}_i\right\}_{i=1,2,\ldots}.$$

Such M and N are the x_1 and x_2 coefficients of the approximating linear functional $\mathcal{L}_x$ from (7.5). They are uniquely defined, because linear independence of the versors implies that the rank of the cosine-sine projection matrix is greater or equal than two. Since f is derivable with respect to θ_i if and only if $i \in \{\bar{\imath}_k\}_{k=1,2,\ldots}$ the weights $w(h_{\theta_i}(\varepsilon))$ are non-zero only for $i \in \{\bar{\imath}_k\}_{k=1,2,\ldots}$. As a consequence M and N are solutions of (7.11), and since they are constant as ε varies, the limit in equation (7.13) exists and $(M, N) = \overline{\nabla} f(x)$.

The typical case when the above proposition plays a role is when f is piecewise smooth function.

7.2.5 Discrete domain anisotropic gradient

Another look at the anisotropic *LPA-ICI* estimation strategy

Let us summarize again, quite informally, the key ideas behind our function estimation strategy. We try to recover (from its noisy observations) a function that lacks traditional regularity hypotheses. However, we assume that there exists a neighborhood $\tilde{U}_x^*$ where the function has some degree of regularity. This regularity can be used to estimate the true value of $y(x)$ using a kernel supported on $U_x^* = \{v \in \mathbb{R}^d : x - v \in \tilde{U}_x^*\}$. We find an approximation of the set U_x^* (which is generally unknown) by using a finite number of directional families of varying scale kernels. The *ICI* algorithm is used to choose from them an adaptive-scale estimate $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$ for each direction. Such scales $h^+(x, \theta_k)$ define the boundary of U_x^+. The set U_x^+ itself is defined as the union of the supports of the adaptive-scale kernels $g_{h^+(x,\theta_k),\theta_k}$. At this point, one could design a kernel $g_x^\times$ supported on U_x^+ and estimate *again* the value of $y(x)$ as $\int_{U_x^+} g_x^\times(v)\, z(x - v)\, dv$. However, this is not computationally efficient as it requires one independent integration for each estimation point. We follow a more efficient approach, and instead use the *already computed* estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$ and a simple fusing procedure (4.8) to obtain the final *anisotropic estimate* $\hat{y}(x)$. This is equivalent to performing the integration $\int_{U_x^+} g_x^+(v)\, z(x - v)\, dv$, where $g_x^+ = \sum_k \lambda_k g_{h^+(x,\theta_k),\theta_k}$ is a kernel supported on U_x^+.

We offer another "less geometrical" interpretation of this approach, substantially similar to the considerations of Section 4.5.1.

The adaptive estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$ are different estimates of the same (unknown) value $y(x)$. Each one of them has its own variance σ_k^2. It is quite natural to try "fitting" one estimate $\hat{y}(x)$ of $y(x)$ in such a way that its residual

differences with the already known estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$ are minimized. A popular way to do such fitting, is to minimize the following weighted sum of squares,

$$\hat{y}(x) = \underset{\xi}{\operatorname{argmin}} \sum\nolimits_k \sigma_k^{-2} \left(\xi - \hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right)^2. \tag{7.14}$$

The fused estimate (4.8) $\hat{y}(x) = \sum_k \sigma_k^{-2} \left(\hat{y}(x) - \hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right) / \sum_j \sigma_j^{-2}$ is in fact exactly the minimizer of (7.14).

Discrete domain anisotropic gradient: a least-squares fitting

For a given direction θ_k let $\hat{\partial}_{\theta_k}$ be the directional (left-hand) derivative estimated with the adaptive scale $h^+(x,\theta_k)$ selected by the *ICI* rule. If the estimation is correct and the function is differentiable, then this directional derivative is also obtained from the gradient as $\hat{\partial}_{\theta_k} = (\cos\theta_k \partial_{x_1}, \sin\theta_k \partial_{x_2})$. We use these different directional derivatives to estimate the gradient by looking for the pair $(\partial_{x_1}, \partial_{x_2})$ that minimizes the differences between the found $\hat{\partial}_{\theta_k}$ and the computed $(\cos\theta_k \partial_{x_1}, \sin\theta_k \partial_{x_2})$. Since $\hat{\partial}_{\theta_k}(x)$ have different variance depending on $h^+(x,\theta_k)$, we perform the minimization in a weighted fashion, we introduce weights λ_k that depend on the variance $\sigma_k^2(x)$ of $\hat{\partial}_{\theta_k}(x)$

$$\hat{\nabla}^T = \left(\hat{\partial}_{x_1}, \hat{\partial}_{x_2}\right)^T = \underset{\partial_{x_1},\partial_{x_2}}{\operatorname{argmin}} \sum_k \lambda_k \left(\hat{\partial}_{\theta_k} - (\cos\theta_k, \sin\theta_k)\left(\partial_{x_1}, \partial_{x_2}\right)^T\right)^2. \tag{7.15}$$

Using the weighted least squares method, such solution is obtained as

$$\hat{\nabla} = \left[D^T \Lambda D\right]^{-1} D^T \Lambda \hat{D}, \tag{7.16}$$

where $D = [(\cos\theta_k, \sin\theta_k)]$, $\hat{D} = \left[\hat{\partial}_{\theta_k}\right]$ and $\Lambda = diag(\lambda_k)$ are respectively a $2 \times K$ matrix, a column vector of length K and a diagonal $K \times K$ diagonal weight matrix.

How exactly λ_k should depend on $\sigma_k^2(x)$ is the central issue in the definition of the discrete anisotropic gradient.

Enforcing a derivability condition in the discrete domain

Just as in the convex fusing (4.8), we may set $\lambda_k = \sigma_k^{-2}(x)$. If this is done, (7.16) is exactly a two-dimensional generalization of (4.8), as the latter is also the solution of the analogous minimization (7.14) of the weighted squared-error between the final function estimate $\hat{y}(x)$ and the adaptive directional estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}$.

Recall from the definition of the continuous domain anisotropic gradient that the weight function $w(h_{\theta_k})$ was zero whenever the directional derivative $\frac{\partial f}{\partial \theta_k}$ didn't exist. Since derivation is a limit process, requiring infinitely many samples, the concept of *derivability* cannot be found in the discrete domain as a straightforward modification of the original continuous domain definition.

Derivability can be interpreted as coherence across the finest scales h of the incremental ratio (by definition, an approximation which in its limiting form converge to the derivative), precisely, this coherence means that when $h \to 0$ the limit exists.

We reformulate this concept in the discrete domain as follows: f is derivable[3] if the varying-scale estimates of the derivative $\hat{y}^{(1,0)}_{h,\theta_k}(x)$ are coherent across the finest scales. The *ICI* rule checks such a coherence. Therefore we say that f is derivable if and only if $h^+ > \min H$. Observe that if H is a continuous set of scales including 0, then the inverse of the variance $\sigma_k^{-2}(x)$ of non-derivable estimates would be 0, since – as a limit – $\sigma^2_{\hat{y}^{(1,0)}_{h,\theta_k}} = +\infty$ when $h = 0$. For the discrete case, if $m = 1$ the smallest scale $\min H$ can be considered to be equal to 2. If higher-order polynomial smoothness is used, $\min H \geq m_1 + 1$.

We enforce the derivability condition by defining $\lambda_k = 0$ for all k such that $h^+(x, \theta_k) = \min H$. If $h^+(x, \theta_k) > \min H$, then $\lambda_k = \sigma_k^{-2}(x)$. It means that (7.16) is not influenced by the estimates corresponding to directions where the derivability condition has not been satisfied[4]. This enforcement is exactly as the condition (7.10), used in the definition of the continuous-domain anisotropic gradient.

7.2.6 More examples

Figure 7.14 shows the anisotropic gradient $\hat{\nabla}$ of the pyramidal function from Figure 7.2 (page 85). It is represented by the two partial derivatives found by solving (7.16). The sum of the squared-residuals of the minimization (7.15) are shown in Figure 7.15. The residuals are large in correspondence of points where the anisotropic gradient is not-well defined. In practice this "ill-definedness" consists in the existence of two (or more) anisotropic neighborhoods whose corresponding anisotropic gradients cannot be matched together. These facts are well related to the Proposition on page 96.

Quite evidently, the anisotropic gradient can be used to synthesize any directional derivative as $\frac{\partial f}{\partial \theta}(x) = \left(\hat{\nabla} f\right)(\cos\theta_k, \sin\theta_k)^T$. These derivatives are automatically left-hand, right-hand, or bilateral depending on the anisotropy of U_x^+, as shown in Figure 7.16.

Some additional illustrations of the anisotropic gradient for natural images are found in Figures 7.17, 7.18, 7.20, and 7.21. The adaptive scales $h^+(\cdot, \theta_k)$ clarify the adaptivity of the method. Figure 7.19 and 7.21 present, for comparison, also some gradient estimation results obtained using the classical Sobel filters.

An application example of the anisotropic gradient is presented in Section 10.2.

[3] at x, along θ, and – if the support is directional – from the left-hand side.

[4] To allow the definition of the anisotropic gradient for all points (even those that are not strictly considered of differentiability by the above rule) we impose $\lambda_k = 1$ for all k, whenever λ_k is found to be zero for all k. However, a more refined approach would be to regularize all weights for those directions whose non-orthogonal λ_k are also zero.

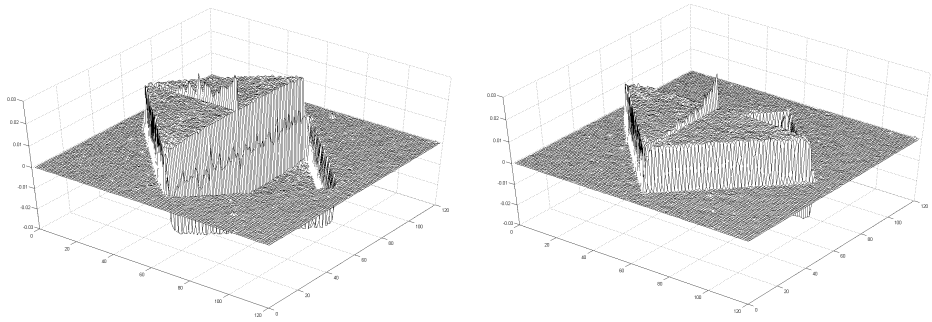


Figure 7.14: Anisotropic gradient: $\frac{\partial f}{\partial x_1}$ (left) and $\frac{\partial f}{\partial x_2}$ (right).

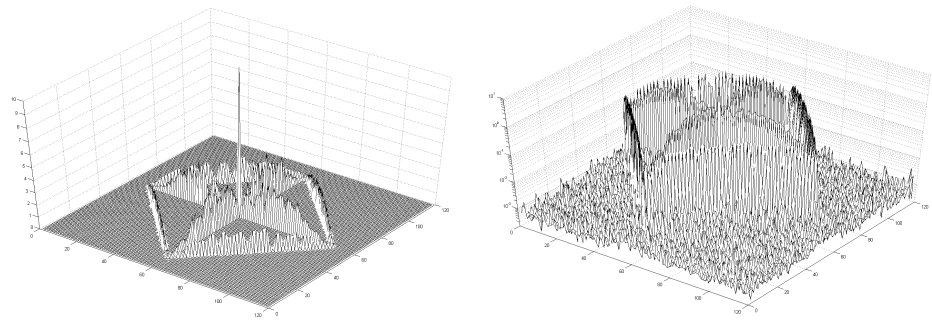


Figure 7.15: Sum of squared residuals, indicating points were the anisotropic gradient is not well-defined. Logarithmic scale is used on the right.

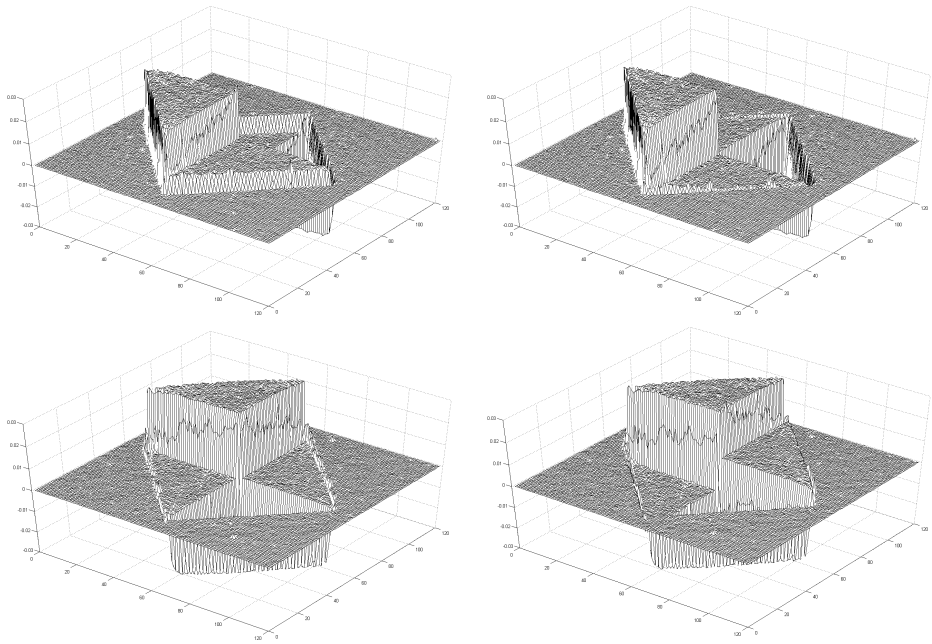


Figure 7.16: Other directional derivatives obtained as linear combination of $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$.

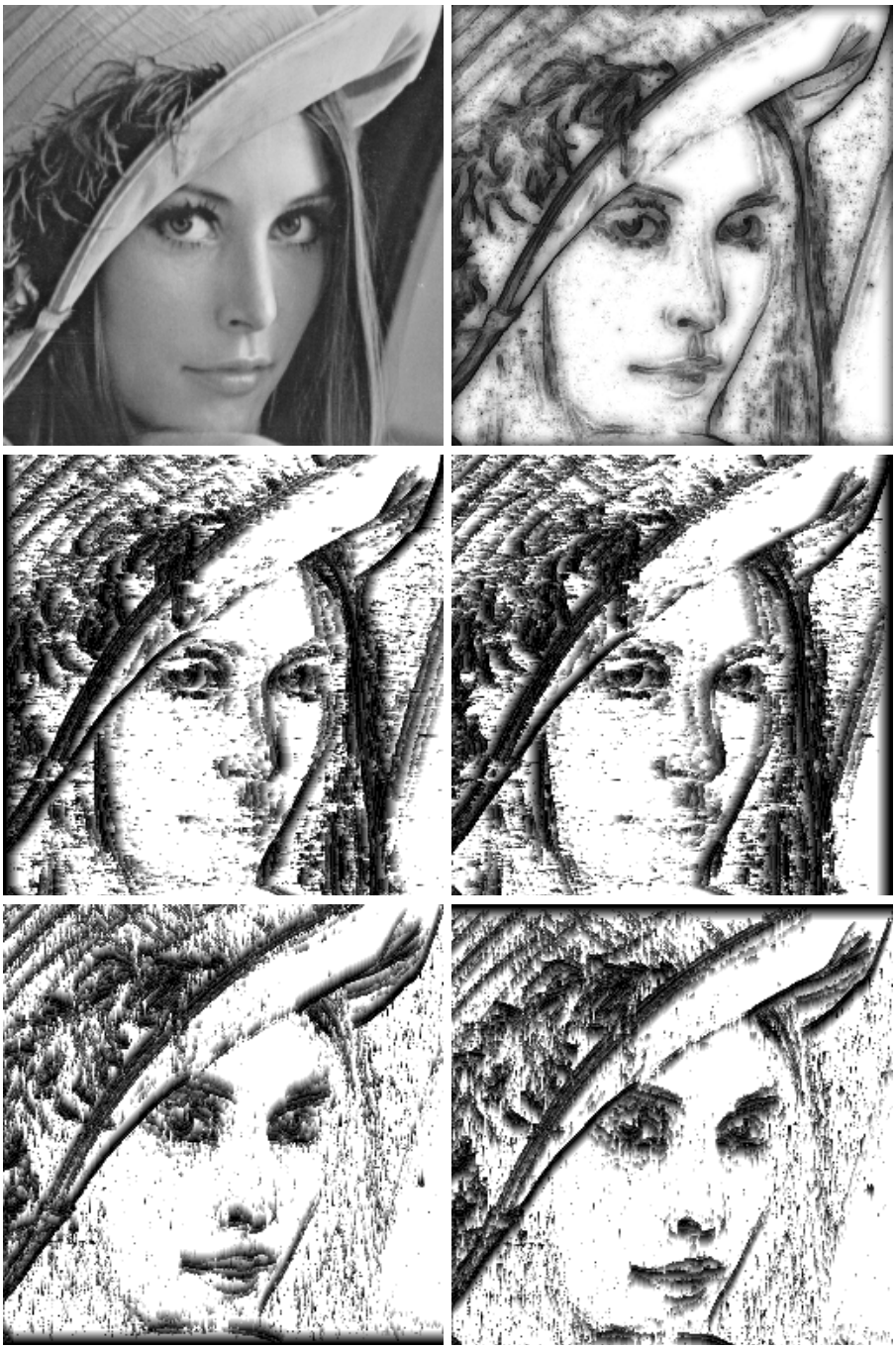


Figure 7.17: A detail of the *Lena* image (top left), adaptive scales for the horizontal (middle) and vertical (bottom) left derivative (left) and right derivative (right). The mean value of the adaptive scales computed among all directions is shown in the top right subimage. Observe how the adaptive scales reveal the edges and contours of the image.

Figure 7.18: Anisotropic gradient: $\frac{\partial f}{\partial x_1}$ (left column) and $\frac{\partial f}{\partial x_2}$ (right column). Same results are shown in different rows with different contrast to enhance visualization.

Figure 7.19: Sobel derivative filters: $\frac{\partial f}{\partial x_1}$ (left column) and $\frac{\partial f}{\partial x_2}$ (right column). Same results are shown in different rows with different contrast to enhance visualization.

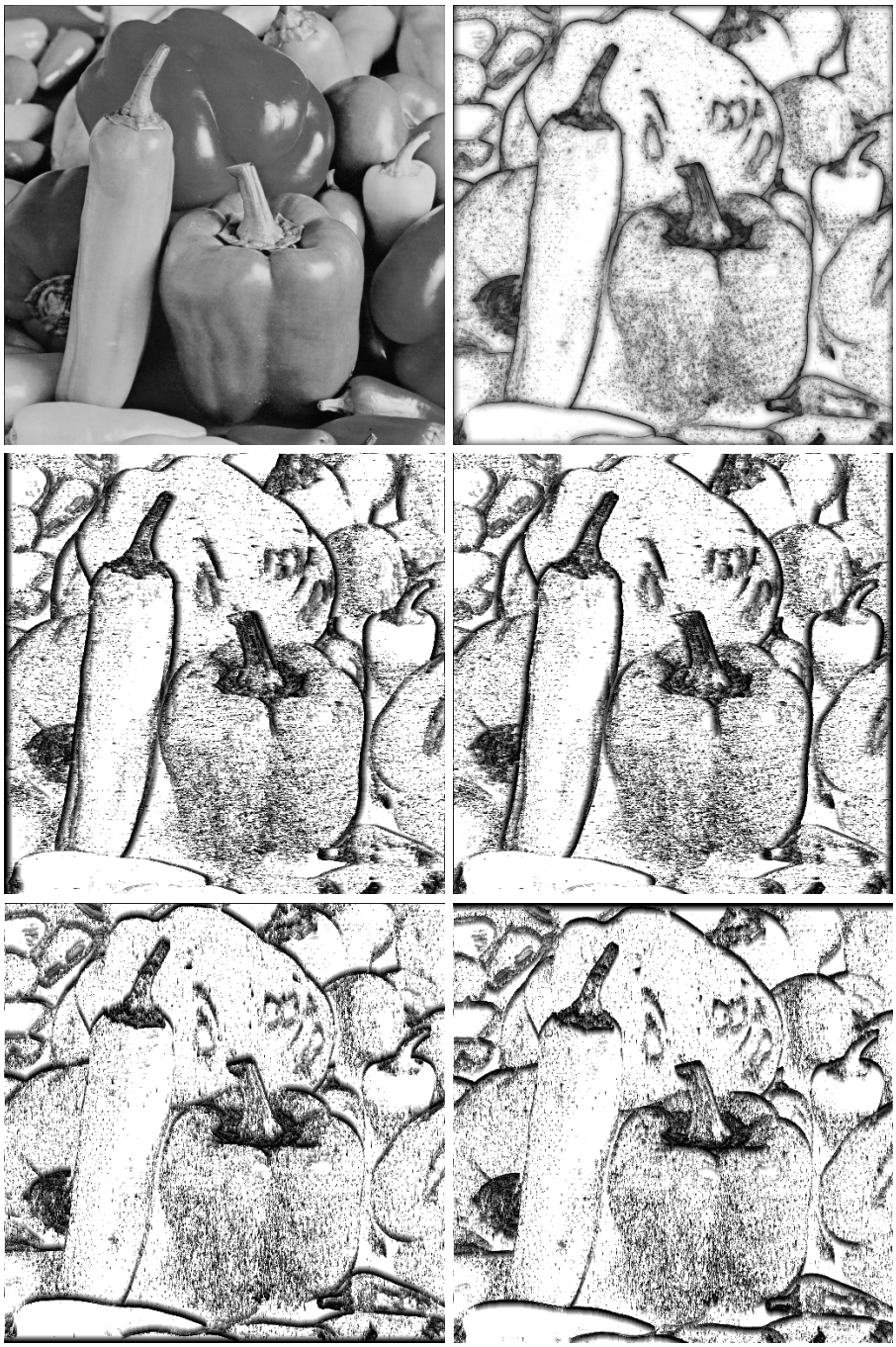


Figure 7.20: *Peppers* image (top left), adaptive scales for the horizontal (middle) and vertical (bottom) left derivative (left) and right derivative (right). The mean value of the adaptive scales computed among all directions is shown in the top right subimage. Observe how the adaptive scales reveal the edges and contours of the image.

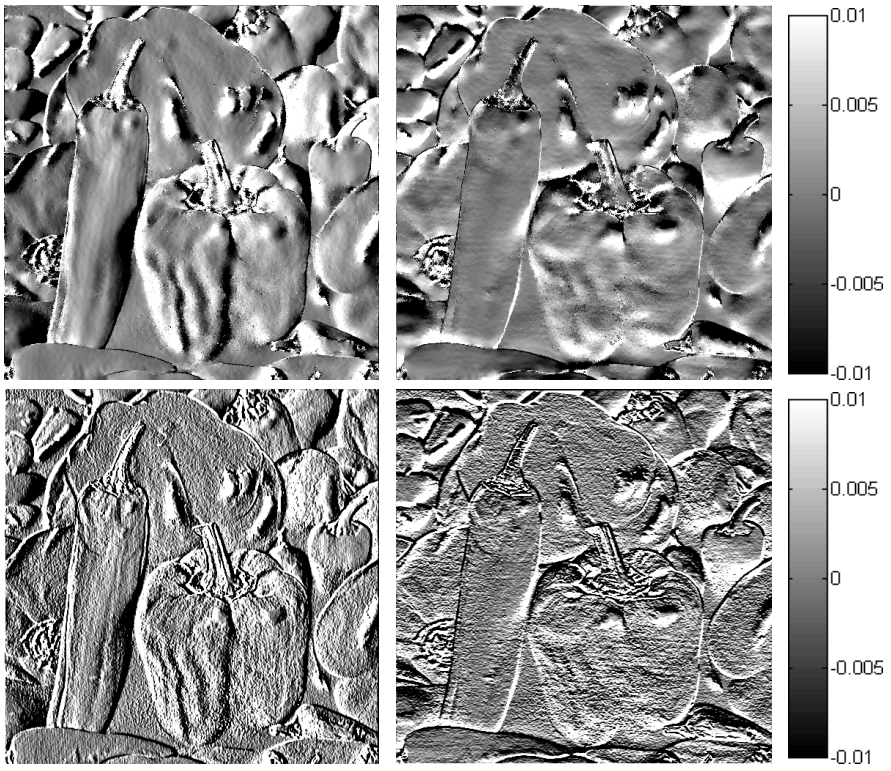


Figure 7.21: Anisotropic gradient: $\frac{\partial f}{\partial x_1}$ (top left) and $\frac{\partial f}{\partial x_2}$ (top right). The same derivatives estimated by the Sobel derivative filters are shown in the bottom row.

Part II

Algorithms, applications and further examples

Chapter 8

Denoising

8.1 Additive white Gaussian noise

The anisotropic *LPA-ICI* algorithm for the additive Gaussian white noise model has been described in full detail in Chapter 4. In this section we only add a few more experimental results, and some comparison with other techniques.

Figure 8.1 shows the noisy observation of the *Cameraman* image, ($\sigma = 0.1$). Figure 8.2 presents fragments of the original (for comparison) and of three restored images. One is obtained using the anisotropic *LPA-ICI*, the other two are the results of translation-invariant wavelet thresholding [9]. Wavelet thresholding is performed using the best-found (oracle) value of the threshold parameter. Although the *ISNR* values are not too different, the image reconstructed by the anisotropic algorithm is visually much better, presenting well defined edges, faithfully reconstructed details, and no noticeable artifacts (such as the unpleasant ringing visible in the Daubechies-wavelets estimate).

8.2 Recursive *LPA-ICI* implementation

In this section we present an efficient, although not exact, implementation of the recursive *LPA-ICI* algorithm.

The residual noise in the anisotropic estimate $\hat{y}$ is no longer uncorrelated (estimation neighborhoods may overlap one with each other) nor its standard deviation is a constant (estimation neighborhoods are adaptive), as shown in Figure 6.2 on page 80. The expression for the variance of the anisotropic estimate is given, depending on the formula which is used for the fusing, by the corresponding formula from Section 4.8: (4.22), (4.23)[1], or (4.24).

[1] In the general case the variance of the noise is not constant. The expression of formula (4.23) for heteroskedastic observations is

$$\sigma^2_{\hat{y}(x)} = \frac{\sum_k \sigma_k^{-2} - \sum_k \left(\frac{\sigma_z(x) g_{h_+(x,\theta_k),\theta_k}(0)}{\sigma_k^2} \right)^2 + \left(\sum_k \frac{\sigma_z(x) g_{h_+(x,\theta_k),\theta_k}(0)}{\sigma_k^2} \right)^2}{\left(\sum_j \sigma_j^{-2} \right)^2}.$$

Figure 8.1: The noisy observation of the *Cameraman* image, $\sigma = 0.1$. SNR=14.39dB, $PSNR$=19.97dB, $RMSE$=25.57, MAE=20.38, MAX=106.73.

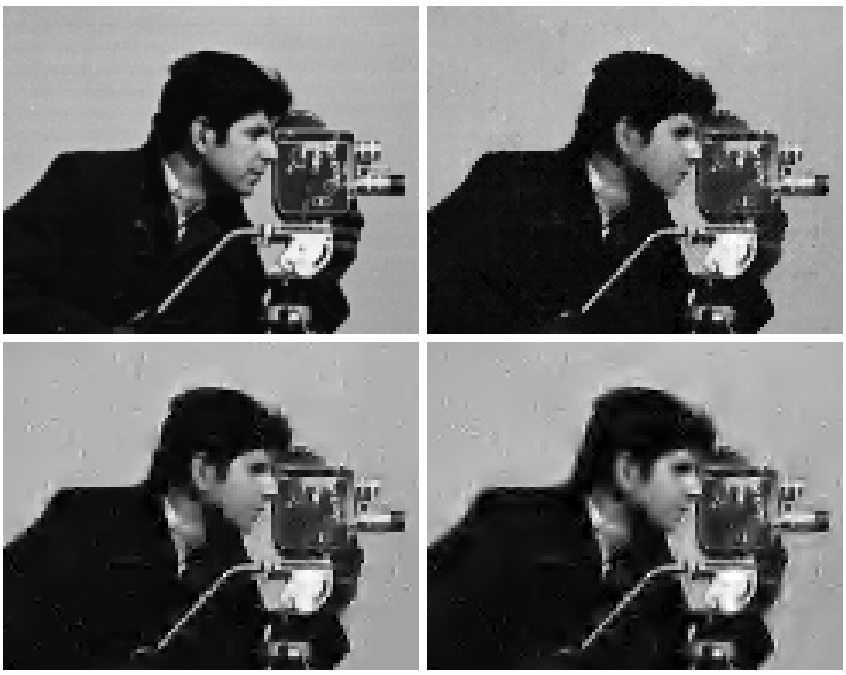


Figure 8.2: Denoising of the *Cameraman* image, $\sigma = 0.1$. Clockwise from top-left: original image, anisotropic *LPA-ICI* estimate, *ISNR*=8.1dB, translation-invariant Daubechies wavelets (DB4), *ISNR*=7.4dB, and translation-invariant Haar wavelets, *ISNR*=7.8dB.

If we assume that this residual noise is uncorrelated, the standard deviation of the directional estimates $\hat{y}^{[2]}_{h,\theta_k}(x)$ for the second stage of the recursive algorithm would be simply calculated as the convolution $(g^2_{h,\theta_k} \circledast \hat{\sigma}^2_{\hat{y}^{[1]}})^{1/2}$, avoiding the use of the complicated kernel $G^{[2]}_{x,h,\theta_k}$. This reasoning may be extended to further iterations, assuming that the noise in $\hat{y}^{[l]}$ is always uncorrelated. However, as this assumption does not hold, the quality of estimation deteriorates, and typically results in oversmoothing of details in the image. It turns out, for low-order kernels, that a simple compensating factor for the standard deviation can effectively reduce this degeneration. This modification of the calculation of the variance may be interpreted as an attempt to filter out only the white component of the residual noise.

After setting the initial conditions $y^{[0]} = z$ and $\hat{\sigma}^{[0]}_y \equiv \sigma$, the l-th recursive step of the modified recursive algorithm is [2]

$$\hat{y}^{[l]} = \mathcal{LI}(\hat{y}^{[l-1]}), \qquad \hat{\sigma}_{\hat{y}^{[l]}} = \left(\sum\nolimits_k \left(\hat{\sigma}^{[l]}_k\right)^{-2}\right)^{-1/2}, \qquad l = 1, 2, \ldots \quad ,$$

where $\hat{\sigma}^{[l]}_k = \hat{\sigma}_{\hat{y}^{[l]}_{h+(x,\theta_k),\theta_k}}$, $\hat{\sigma}_{\hat{y}^{[l]}_{h,\theta_k}} = \alpha(g^2_{h,\theta_k} \circledast \hat{\sigma}^2_{\hat{y}^{[l-1]}})^{1/2}$, and $\alpha < 1$ being the fixed correcting factor.

[2] This expression is based on the variance (4.22). In the actual algorithms, however, we use the fusing formula given in the preceding footnote. This is because the kernels overlap in the origin pixel.

In spite of the striking simplicity of the modification, simulation results show that it enables *ICI* to properly select the adaptive scale. Moreover, convergence of the above recursive system is easily guaranteed, since $\hat{\sigma}_{\hat{y}}^{[l]} = \mathcal{O}(\alpha^l) \to 0$. More precisely, since $\|g_x^+\|_2 \leq 1$, there exist a constant c such that $|\hat{y}^{[l]}(x) - \hat{y}^{[l+1]}(x)| < c\hat{\sigma}_{\hat{y}}^{[l]}(x) \leq c\alpha^l\sigma$. This implies that $\hat{y}^{[l]}(x)$ is a Cauchy sequence. Qualitatively, the actual convergence rate of the algorithm depends on $\mu(U_x^+) \approx \|g_x^+\|_2^{-1}$, and usually the algorithm reaches a numerical steady-state already after three iterations. The proposed recursive method can be used for accurate detail-preserving image denoising, segmentation and edge detection applications.

8.2.1 Simulations

Table 8.1 shows the *ISNR* and *MAE* (ℓ^1-distance) results for the restoration of the *Cameraman* image, corrupted by additive Gaussian white noise, $\sigma = 0.1$. This noisy observation is shown in Figure 8.1). Zero-order uniform kernels for a total of eight directions and four scales, $h \in \{1, 2, 3, 5\}$, were used with fixed $\alpha = 2/3$. These results are illustrated (for a fragment of the image) in Figure 8.3. The table shows a fast convergence of the iterations and criteria values attesting the high quality of the filtering.

iteration #	noisy	1	2	3	4	5	6
ISNR (dB)	0	7.361	8.098	8.119	8.120	8.120	8.120
MAE (ℓ^1)	20.38	7.894	6.597	6.538	6.535	6.535	6.535

Table 8.1: *ISNR* and *MAE* results for the *Cameraman* image denoising experiment (σ=0.1, *SNR*=14.39dB).

By using polynomial-order kernel mixtures[3] and a larger set of scales, it is possible to achieve, for the same experiment, an *ISNR* of 7.50, 8.23 and 8.47dB at the first, second and third iteration (shown in Figure 8.5), respectively. A similar performance cannot be achieved by the non-recursive algorithm.

Figure 8.4 shows another recursive anisotropic denoising example for the *Cheese* image.

Recursions with a different fusing

Figure 8.6 show the second iterations corresponding to the example shown in Figure 4.12 of Section 4.7. In the example, besides the standard fusing formula (4.8), also the different formula (4.20) has been used. To calculate the variance of the fused estimate from the initial iteration (needed in the second iteration in order to compute the standard deviations of the directional estimates), the two corresponding formulas (4.23) and (4.24)[4] are used.

[3] An explanation of the polynomial-mixture kernels [7] will be given in the following section.

[4] Observe that formula (4.24) assumes constant variance. Therefore it can be used to compute the variance after the first iteration, but not after the second. For this reason, we do not perform a third iteration.

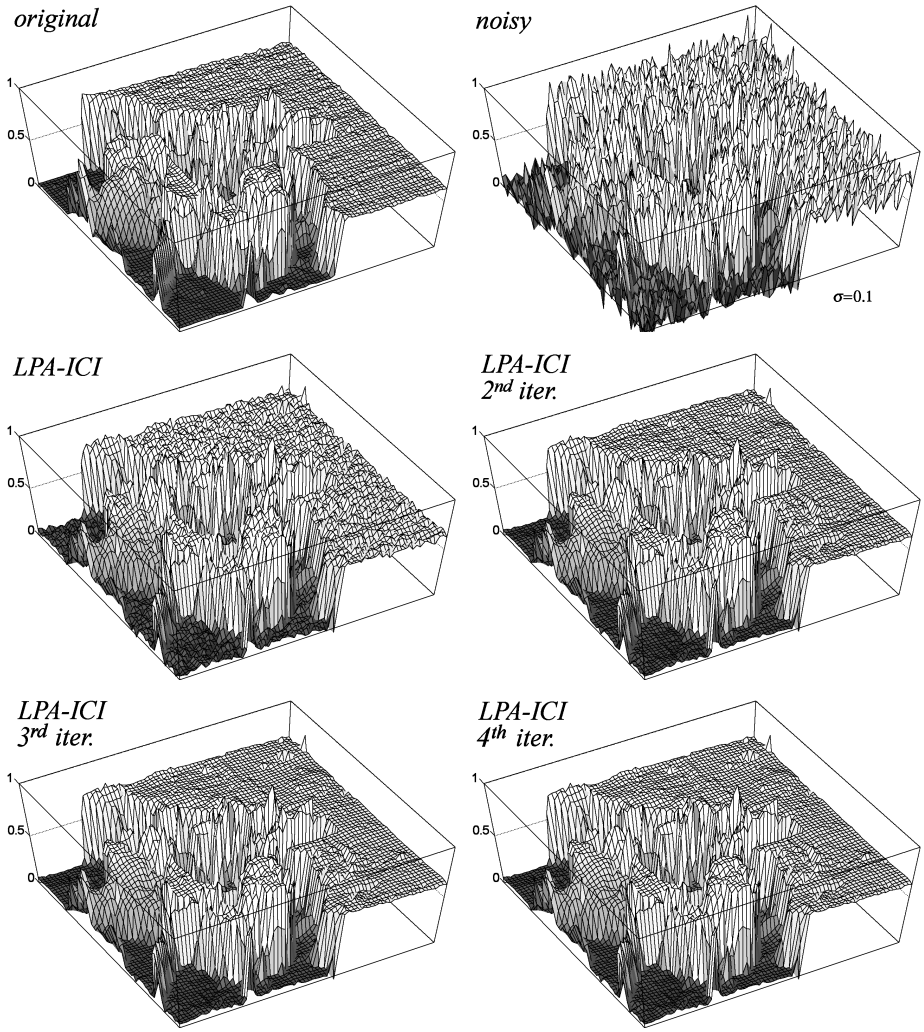


Figure 8.3: A fragment of the *Cameraman* image, filtered by the recursive anisotropic *LPA-ICI* algorithm. After the third iteration, the recursive procedure yields essentially identical estimates, confirming the fast convergence of the algorithm.

8.3 Signal-dependant noise

Each successive iteration of the recursive anisotropic *LPA-ICI* algorithm, which we presented in the previous section, can be interpreted as anisotropic *LPA-ICI* filtering in the presence of some heteroskedastic noise with variance $\alpha\hat{\sigma}^2_{\hat{y}^{[l-1]}(x)}$. In this sense, this recursive algorithm can be considered as a solution for a very general class of noisy observations. However, every step of the algorithm – and in particular the first one – assumes that such space-variant variance is known. Such variance is not estimated, it is calculated.

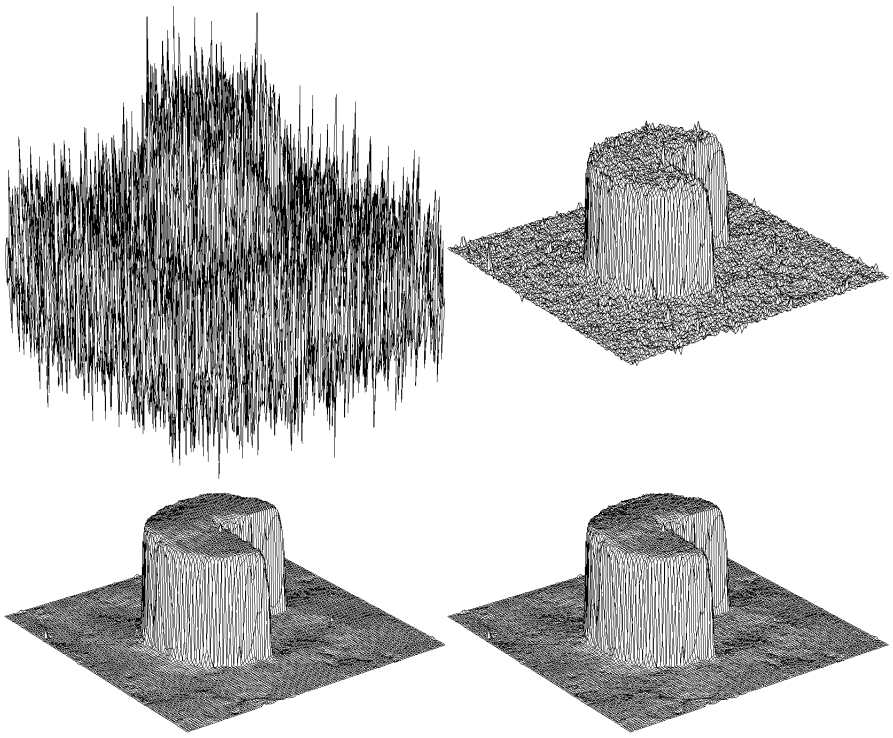


Figure 8.4: Recursive anisotropic *LPA-ICI* denoising of the *Cheese* image. Clockwise, from top-left, the noisy image and the estimates from the first, second and third iteration.

8.3.1 Recursive variance update

The algorithm that we propose for the filtering of signal-dependant noise is based on the above recursive *LPA-ICI* algorithm. However, to deal with the unknown σ_z^2, we introduce an initial recursive procedure, in which the fused estimate $\hat{y}$ is used to update the estimate $\hat{\sigma}_z^2$ of σ_z^2 through the variance function: $\hat{\sigma}_z^2 = \rho(\hat{y})$. This procedure is iterated a few times before the beginning of the actual recursive algorithm, and is schematized here below:

$$
\begin{array}{lllll}
 & \hat{y}^{[0]} = z & & & \\
\text{compute} & \hat{y}^{[1]} = \mathcal{LI}(z) & \text{with} & \hat{\sigma}^2_{\hat{y}^{[1]}_{h,\theta}} = g^2_{h,\theta} \circledast \rho\left(\hat{y}^{[0]}\right) & \boxed{\begin{array}{c}\text{RECURSIVE}\\ \text{VARIANCE}\\ \text{UPDATE}\end{array}} \\
\text{compute} & \hat{y}^{[2]} = \mathcal{LI}(z) & \text{with} & \hat{\sigma}^2_{\hat{y}^{[2]}_{h,\theta}} = g^2_{h,\theta} \circledast \rho\left(\hat{y}^{[1]}\right) & \\
 & \vdots & & \vdots &
\end{array}
$$

Observe that, contrary to the recursive *LPA-ICI* filtering, in the above recursion the filtered image is always z. The estimates of y are used only in order to update the estimate of σ_z^2 which is then used to calculate the variance of the directional estimates. The value of this variance not only has an impact on the *ICI*, but also on the adaptive weights used in the fusing.

In practice, it is enough to perform only a few (say, two or three) of the above recursions in order to obtain a satisfactory estimate of σ_z^2. Once such $\hat{\sigma}_z^2$ is available, the recursive anisotropic *LPA-ICI* may start.

Figure 8.5: Restored *Cameraman* image after three iteration of the recursive anisotropic algorithm, using polynomial-mixture kernels, 8 directions, and $H = \{1, 2, 3, 4, 6, 8, 10, 12\}$: $ISNR$=8.47dB, SNR=22.86dB, $PSNR$=28.44dB, MAE=6.12, MAX=111.10.

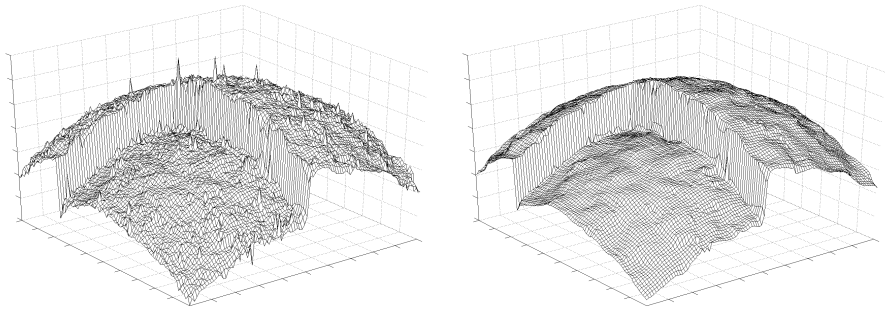


Figure 8.6: Second iterations of the recursive *LPA-ICI* procedure following the initial results (first iteration) shown in Figure 4.12. Standard fusing (4.8) and modified fusing (4.20) are used, respectively, for the left and the right image.

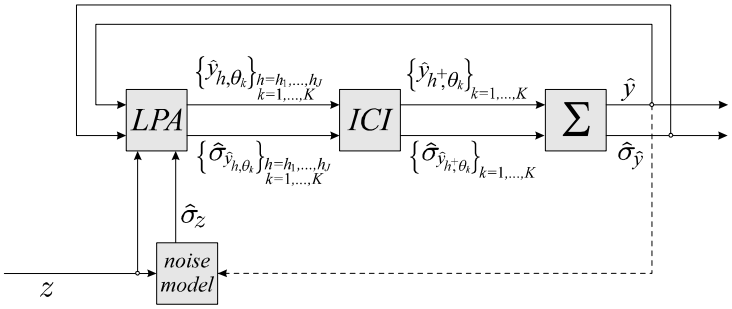


Figure 8.7: General layout of the recursive anisotropic *LPA-ICI* filtering (solid line), and of the recursive variance update (dashed line).

Figure 8.7 show the general layout of the two iterative procedures. We describe various modifications of these recursions in [22].

8.3.2 Poisson denoising experiments

We begin by showing some experimental results taken from [47][5], where we focus on the Poisson observation model. We compare the performance of the recursive *LPA-ICI* (with recursive variance update) against state-of-the-art wavelet-based methods, which exploit quite sofisticated statistical modeling – in wavelet domain – of the Poissonian nature of the observations.

Poisson observations

In our simulations for the Poissonian case, in order to achieve different level of randomness (i.e. different *SNR*) in the noisy observations, we first multiply the true signal y^{TRUE} (which has range [0,1]) by a scaling factor χ:

$$y = \chi \cdot y^{\text{TRUE}}, \quad z \sim \mathcal{P}(y).$$

[5] [47]: Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Anisotropic local likelihood approximations", *Proc. of Electronic Imaging 2005*, 5672-19, 2005.

True image Noisy image, χ=120 Noisy image, χ=30 Cross-section

Figure 8.8: *Cameraman* fragment: true and Poisson noisy images, with different level of randomness.

Thus, $E\{z\} = var\{z\} = y = \chi \cdot y^{\text{TRUE}}$, and $y/std\{z\} = \sqrt{\chi}\sqrt{y^{\text{TRUE}}}$, i.e. better *SNR* corresponds to larger χ.

This modelling of Poisson data allows to produce a comparison with the similar simulation scenarios appeared in a number of publications [91, 76, 96, 68]. We make a comparison with the wavelet-based methods recently developed for the Poisson data and demonstrating a good performance.

Only to visualise the data we divide back by the factor χ, so that the expected range of the signal intensity is again $[0, 1]$. Figure 8.8 illustrates the effect of this scaling factor in modelling Poisson observations. Comparing the images in this figure, we can see that the noise level for χ=120 is much lower than it is for χ=30. From the cross-section we can note that the level of this random disturbance is clearly signal-dependent. Large value of the signal means larger level of the noise.

Optimization of the algorithm

Some work has been done in order to optimize the design parameters of the above algorithm. After this optimization, the algorithm with these parameter values was used for multiple experiments, part of which is presented in what follows.

Similarly as it is proposed in [7], we use polynomial mixtures. The directional kernels $g_{h,\theta}$ are defined as a linear combination of zero and first order kernels:

$$g_{h,\theta} = \alpha \tilde{g}_{h,\theta}|_{m=(0,0)} + (1-\alpha)\tilde{g}_{h,\theta}|_{m=(1,0)}. \tag{8.1}$$

These $\tilde{g}_{h,\theta}|_{m=(0,0)}$ and $\tilde{g}_{h,\theta}|_{m=(1,0)}$ are directional-*LPA* kernels designed from a set of uniform window functions w_h constant on their sectorial support. The scale parameters h, which define the length of the support, were taken from the following set:

$$H = \{1,\ 2,\ 3,\ 4,\ 6,\ 8,\ 10,\ 12\}.$$

The *ICI* rule is applied for the selection of the length h of the kernel $g_{h,\theta}$.

The parameter α in the combined kernel $g_{h,\theta}$ (8.1) is taken with different values for the different steps of the algorithm, starting from $\alpha = 1$ (zero order), and increasing then the importance of the first order component as the algorithm progress. We set $\Gamma = 0.7$ for all iterations.

Figure 8.9: Filtering a fragment of the *Cameraman* image: (clockwise from top-left) noisy data, χ=60 (*MSE*=23.9), estimate from the first iteration of the *LPA-ICI-AV* algorithm (*MSE*=7.90, *ISNR*=4.81dB), estimate after the fourth iteration (*MSE*=4.36, *ISNR*=7.40dB), and original image.

8.3.3 Simulation results

Images shown in Figures 8.9 and 8.10 show the noisy and original images and the estimates obtained at the first (initialization) and the last (fourth) iteration. The *MSE* values demonstrate a fast performance improvement in the successive iterations. The quality of the final estimates is quite good visually and numerically. In particular, for *Cameraman* we achieve: *ISNR*=9.34dB for χ=30, *ISNR*=8.05dB for χ=60, *ISNR*=7.45dB for χ=90, *ISNR*=6.82dB for χ=120.

Some numerical results and comparison with other methods for the *Cameraman* and *Lena* images are presented in Table 8.2 and Table 8.3. The results in the tables are the values of the *MSE*, calculated as the mean value of $|\hat{y} - y|^2$. This table includes and extends the results shown in [68].

Comparing the *MSE* values obtained for the successive steps we can note that the main improvement is achieved in first three steps. Starting from the second step of the recursive procedure, the *LPA-ICI* shows superior results which are essentially better than those from the other methods[91, 68].

Figure 8.10: Filtering the *Lena* image: top row, left - noisy data χ=60 (*MSE*=29.0, *SNR*=15.3 dB); top row, right - first iteration of the *LPA-ICI-AV* algorithm (*MSE*=7.07, *ISNR*=6.13 dB); second row, left - fourth iteration (*MSE*=2.62, *ISNR*=10.4 dB); second row, right - original image.

	Cameraman			
	χ=30	χ=60	χ=90	χ=120
Noisy image	13.9	27.5	42.1	56.0
TN method [91]	2.76	7.73	14.11	21.59
Improved TN method [68]	2.13	5.37	9.22	13.59
LPA-ICI, 1st step	3.75	8.04	13.4	18.8
LPA-ICI, 2nd step	2.28	5.30	8.96	13.2
LPA-ICI, 3rd step	1.79	4.50	7.79	11.8
LPA-ICI, 4th step	1.62	4.30	7.58	11.6

Table 8.2: *MSE* for the *Cameraman* image for different algorithms and different levels of noise.

	Lena			
	χ=30	χ=60	χ=90	χ=120
Noisy image	14.5	29.0	43.6	58.0
TN method [91]	2.33	6.46	11.62	17.89
Improved TN method [68]	1.98	5.32	9.35	14.03
LPA-ICI, 1st step	3.29	7.07	11.3	15.9
LPA-ICI, 2nd step	1.76	4.07	6.80	9.84
LPA-ICI, 3rd step	1.20	3.05	5.33	7.99
LPA-ICI, 4th step	0.99	2.62	4.72	7.19

Table 8.3: *MSE* for the *Lena* image for different algorithms and different levels of noise.

8.3.4 Other types of noise

This section is based on the experimental part of [22][6]. We consider a wider class of signal-dependant noises, and we compare our results with other state-of-the-art adaptive algorithms.

These simulations are, in a sense, more interesting than those presented in the previous section: here, exactly the same algorithm parameters are used for the restoration from three different kinds of noise. It means that specific aspects of the noise distributions cannot be exploited. These experimental results confirm the generality, and the robustness of the *ICI* algorithm with respect to various non-Gaussian-distributed estimates.

Three common types of signal-dependant noise are considered: the "scaled" *Poisson* noise, $\chi z \sim \mathcal{P}(\chi y)$, $\chi \in \mathbb{R}^+$, the *film-grain* noise, $z = y + Ky^\alpha \eta$, $K, \alpha \in \mathbb{R}^+$ and $\eta \sim \mathcal{N}(0,1)$, and the "multiple-look" *speckle* noise, $z = L^{-1} \sum_{i=1}^{L} y\epsilon_i$, $\epsilon_i \sim \mathcal{E}(\beta)$, $\beta \in \mathbb{R}^+$. The calligraphic letters $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{E}$ denote, respectively, the Poisson, Gaussian, and exponential distributions. For the above observation models, the variance functions $\rho(y) = \sigma_z^2$ are $\rho(y) = y/\chi$, $\rho(y) = K^2 y^{2\alpha}$, and $\rho(y) = y^2\beta/L$, respectively.

To enable an objective comparison with the many simulations presented in [81], we set $\chi = 0.1$, $K = 3.3$, $\alpha = 0.5$, $L = 4$, and $\beta = 1$. The true signal y

[6] [22]: Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Anisotropic local approximations for pointwise adaptive signal-dependent noise removal", (accepted) *XIII European Signal Proc. Conf., EUSIPCO 2005*, September 2005.

	test image	noisy	Lee	*NURW*	*ANF*	***LPA-ICI***
Poisson	*Lena* 512×512	1240	—	—	—	82 (11.8)
	Peppers 512×512	1197	—	—	—	79 (11.8)
	Lena 256×256	1239	200	177	151	120 (10.1)
	Peppers 256×256	1206	184	160	145	120 (10.0)
	Aerial 256×256	766	231	252	179	179 (6.3)
Film-grain	*Lena* 512×512	1343	—	—	—	83 (12.1)
	Peppers 512×512	1304	—	—	—	80 (12.1)
	Lena 256×256	1346	206	185	160	125 (10.3)
	Peppers 256×256	1311	199	169	150	120 (10.4)
	Aerial 256×256	828	242	267	188	185 (6.5)
Speckle	*Lena* 512×512	4375	—	—	—	196 (13.5)
	Peppers 512×512	4303	—	—	—	182 (13.7)
	Lena 256×256	4349	365	371	381	269 (12.1)
	Peppers 256×256	4304	370	372	378	269 (12.0)
	Aerial 256×256	1707	348	387	318	329 (7.1)

Table 8.4: *MSE* values for different images, noise models, and methods. In the last column, the value in parentheses is the *ISNR* (dB).

is assumed to have range [0,255]. Note in term of their variance function, the Poissonian and the film-grain observations with $\alpha = 0.5$ are treated identically (up to a multiplicative factor). Nevertheless, even when $K^2 = 1/\chi$ (i.e. when their corresponding variance functions coincide), their corresponding observations are quite different, because of the different distributions. In particular, Poissonian observations are always integer and positive, i.e. $z \in \mathbb{N}/\lambda$, whereas Gaussian distributed observations can take any real value.

In [81], where the main focus is on the adaptive-neighborhood filter (*ANF*) (a technique which - like ours - is based on anisotropic adaptation), are also considered the "refined" Lee filter (Lee) [62] and the noise-updating repeated Wiener filter (*NURW*) [39]. Table 8.4 includes the results from [81] and extends them with those obtained by recursive anisotropic *LPA-ICI* method. Comparing the *MSE* values given in Table 8.4, we may note that for the *Lena* and *Pepper* images the developed algorithm gives essentially better results for all types of noise. For the *Aerial* image we obtain figures which are very close to the best, given by *ANF* algorithm. An illustration of some of these results, attesting the advanced filtering performance of our method, is given in Figure 8.11.

Real data from cameraphone's CMOS sensor

We also show some results obtained using real data acquired using the CMOS sensor of a Nokia cameraphone. The statistical characteristics of the sensor's raw-data have been studied, and were found to follow very accurately the observation model (5.1). The corresponding variance function $\rho(y)$ has been estimated and used in the algorithm. In extreme low-light conditions, or for extremely short exposure-times, the signal-to-noise ratio can be dramatically low. Figures 8.12(left) and 8.13(top) show, respectively, the raw data captured in dim light with an exposure time of 1ms, and the recontructed color-image using the full image-processing chain (which includes white-balance, color cor-

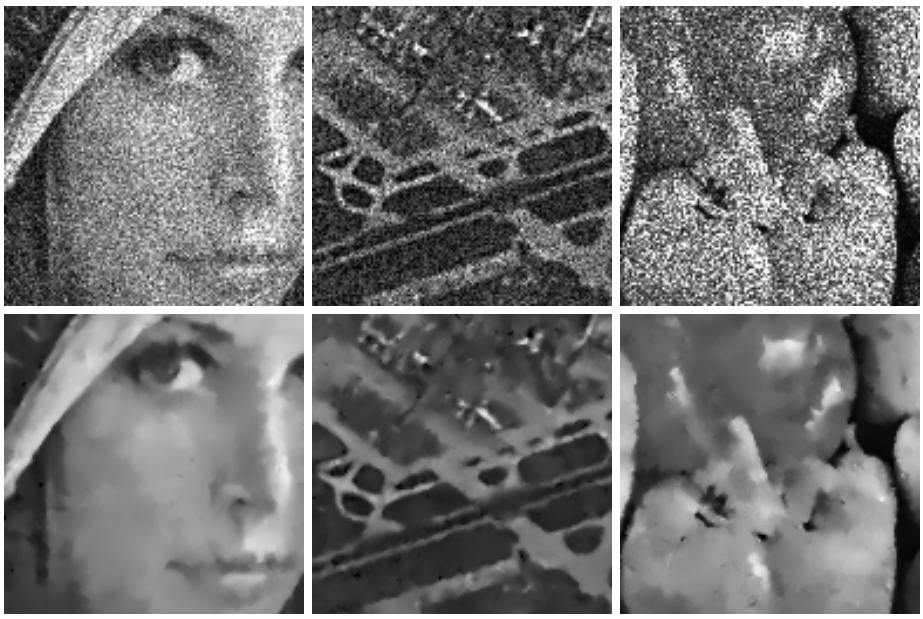


Figure 8.11: Fragments of the noisy images before (top row) and after (bottom row) the anisotropic *LPA-ICI* restoration: (from left to right) Poisson *Lena*, film-grain *Aerial*, and speckle *Peppers* 256×256.

rection, some denoising and the color-array interpolation). Figures 8.12(right) and 8.13(bottom) show the corresponding results obtained when the raw data is filtered by the adaptive *LPA-ICI* method using the estimated variance function $\rho(y)$: smooth areas are faithfully restored and finer details are accurately preserved.

All these experiments were produced using the same algorithm parameters. In this particular implementation, the variance update is performed three times and the recursive adaptive filtering is repeated twice. A set H of seven scales is used, and the anisotropic estimates are obtained by fusing eight directional adaptive estimates. Again, the algorithm uses convex mixtures g_h^λ of zero-order *LPA* kernels g_h^0 and first-order *LPA* kernels g_h^1: $g_h^\lambda = (1-\lambda)g_h^0 + \lambda g_h^1$. These polynomial mixtures [7] allow to achieve a better fit of the data but, contrary to the pure higher-order polynomial produce estimates with a sensibly lower variance.

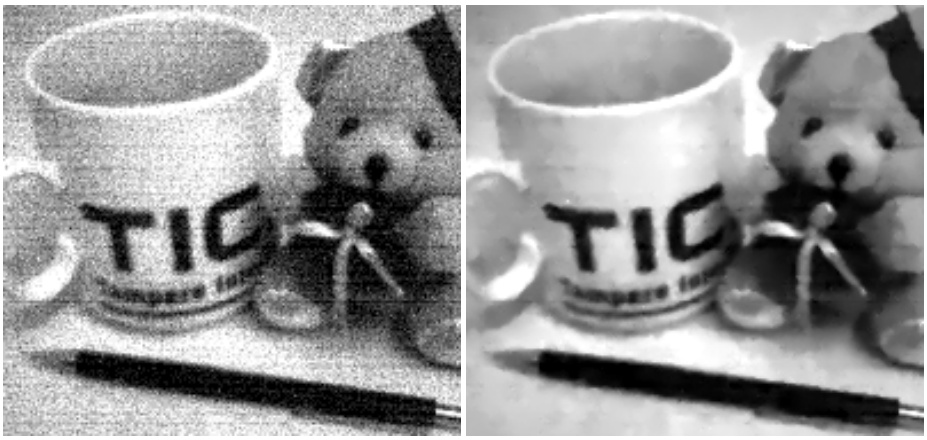


Figure 8.12: Raw data from cameraphone's CMOS sensor, R channel, 1ms exposure (left), and reconstructed image using the *LPA-ICI* adaptive method with the estimated variance function.

Figure 8.13: Color image reconstructed, using the standard imaging chain, from the noisy raw data (top) and from the *LPA-ICI*-filtered raw data (bottom).

Chapter 9

Deconvolution

In this chapter we describe three methods which are based on the nonparametric "regularized inverse-regularized Wiener inverse" deconvolution ([45],[44]). Significant improvement was achieved by endowing the basic deconvolution algorithm with the higher adaptivity of the anisotropic *LPA-ICI* estimator [46][1] (Section 9.1). Some consistent changes have been required in order to enable the anisotropic deconvolution algorithm to perform efficient filtering of Poissonian distributed observations [23] (Section 9.2). A rather different application, which can be modelled as a particular deconvolution problem, is considered in the last section of this chapter: inverse halftoning [21].

9.1 Additive white Gaussian noise

9.1.1 Introduction

We wish to recover an image y from noisy observations

$$z = (v \circledast y) + \sigma\eta,$$

where v is the point-spread function (PSF) of the optical system. It is assumed that the PSF is known and that the noise η is standard Gaussian. In the frequency domain the observation equation has the form

$$Z = YV + \sigma\eta, \tag{9.1}$$

where capital letters are used for the discrete Fourier transform of the corresponding variables.

An unbiased solution of a deconvolution problem of the form (9.1) can be obtained in a straightforward manner by first inverting the convolution operator V and then removing the noise $V^{-1}\sigma\eta$.

However, it is now standard to approach such inverse problems by the method of regularization, in which one applies, rather than the inversion, a regularized inverse operator [13]. A special common point of most methods

[1] [46]: Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Directional varying scale approximations for anisotropic signal processing", *Proc. XII European Signal Proc. Conf., EUSIPCO 2004,* Vienna, pp. 101-104, September 2004.

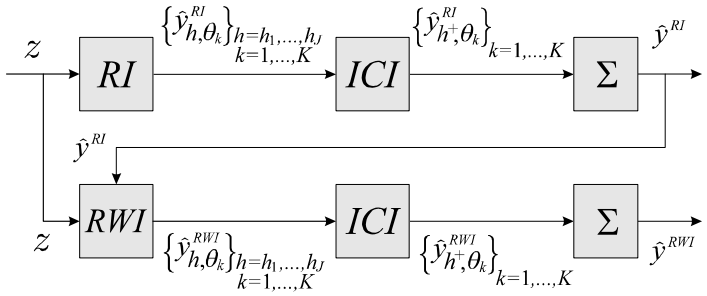


Figure 9.1: Anisotropic *LPA-ICI* regularized Wiener inverse algorithm. In the first line of the flowchart the *RI* estimates are calculated for a set of scales and directions, the *ICI* is used to obtain the pointwise-adaptive directional estimates $\hat{y}^{RI}_{h^+,\theta_k}$ that are then fused into the anisotropic $\hat{y}^{RI}$ estimate. In the second line the *RWI* estimates are calculated using $\hat{y}^{RI}$ as a reference signal in Wiener filtering, again *ICI* and fusing are performed to obtain the final $\hat{y}^{RWI}$ estimate.

starting from the frequency domain equation (9.1) is that some basis functions are used to approximate the object function y in the form of series with coefficients defined from the observations. These functions may be Fourier harmonics, eigenfunctions of the convolution operator in *SVD* methods or wavelets in wavelet multiresolution decompositions. There exist a lot of deconvolution techniques based on this sort of approaches.

Basically different ideas and methods arise from the pointwise nonparametric estimation approach [15]. These methods mostly do not assume any underlying global parametric model of the object and do not use some global parametric series for object approximation. It is assumed only that the object is composed from piecewise regular elements and every point of the object allows a good local approximation. The main goal of estimation is to build a pointwise approximation using the observations from a neighborhood. There is a number of proposals for nonparametric smoothing of *non-blurred* noisy images which allow for preserving the sharp edge structure as well as the edge detection and reconstruction. Actually, these methods are based on kernel smoothing with a special choice of the kernels. Spatial pointwise adaptation is now commonly considered as a crucial element of the nonparametric estimation. These adaptation methods, even for an originally linear method, are finalized in nonlinear estimators [28, 43, 67, 78]. The recently proposed *LPA-ICI deconvolution* [27, 45, 44, 46] exploits the nonparametric smoothing for a deconvolution algorithm where the regularized inversion of the convolution operation and the filtering of the noise are performed *simultaneously* in an adaptive fashion.

The anisotropic version of the *LPA-ICI* estimator [46] is a powerful tool to further improve the adaptivity of the nonparametric approach. We briefly describe this estimator in the following section.

9.1.2 Adaptive *RI-RWI* deblurring algorithm

The considered technique is based on the following regularized inversion (*RI*) and regularized Wiener inversion (*RWI*) algorithms, using the directional-*LPA*

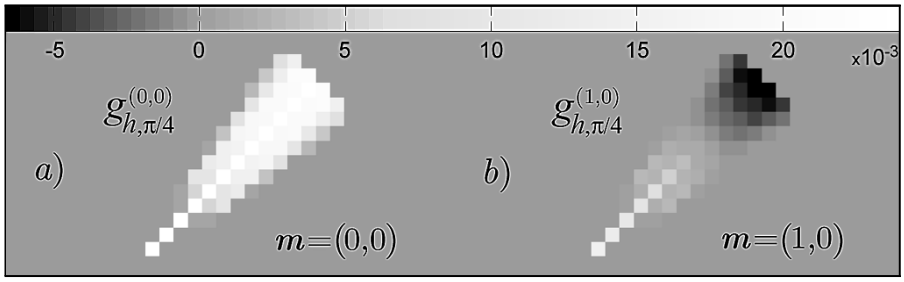


Figure 9.2: Directional smoothing (function estimation) kernel (a) and differentiating kernel (b) obtained by the directional-*LPA* design with $m = (0,0)$ and $m = (1,0)$ respectively.

kernels g_{h,θ_k}:

$$\hat{Y}^{RI}_{h,\theta_k} = \frac{\overline{V}G_{h,\theta_k}}{|V|^2+\varepsilon_1^2}Z, \qquad (RI), \tag{9.2}$$

$$\hat{Y}^{RWI}_{h,\theta_k} = \frac{\overline{V}|Y|^2G_{h,\theta_k}}{|VY|^2+\varepsilon_2^2\sigma^2}Z, \qquad (RWI), \tag{9.3}$$

where $\varepsilon_1, \varepsilon_2 > 0$ are regularization parameters. The estimate of y is given by the *RWI* deconvolution scheme (9.3) that uses the *ICI* based *RI* estimate as a reference signal Y. Thus, we arrive to two steps procedure (see Figure 9.1). The adaptive procedure assumes that the estimates $\{\hat{y}^{RI}_{h,\theta_k}\}_{h\in H}$ are calculated according to (9.2) for a set of scales H and the *ICI* rule selects the best scales for each direction and for each pixel. In this way we obtain the directional varying scale adaptive estimates $\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}$, $k = 1,\dots,K$, which are fused in the final one $\hat{y}^{RI}$ according to (4.8). This $\hat{y}^{RI}$ serves as the reference signal in the *RWI* procedure (see Figure 9.1). The adaptive *RWI* algorithm is similar and gives the *ICI* adaptive varying scales estimates $\hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}$ for each direction and x. Then, the final estimate $\hat{y}^{RWI}$ is obtained by fusing these directional ones again according to (4.8).

The use of the *ICI* rule requires the calculation of the standard deviations of the individual varying scale directional estimates $\{\hat{y}^{RI}_{h,\theta_k}\}_{h\in H}$ and $\{\hat{y}^{RWI}_{h,\theta_k}\}_{h\in H}$. These standard deviations can be easily calculated by the l^2-norm of the frequency response of the corresponding filters:

$$\sigma_{\hat{y}^{RI}_{h,\theta_k}} = \sigma\left\|\frac{\overline{V}G_{h,\theta_k}}{|V|^2+\varepsilon_1^2}\right\|_2, \qquad \sigma_{\hat{y}^{RWI}_{h,\theta_k}} = \sigma\left\|\frac{\overline{V}|Y|^2G_{h,\theta_k}}{|VY|^2+\varepsilon_2^2\sigma^2}\right\|_2.$$

The *ICI* adaptive scales $h^+(\cdot,\theta_k)$ represent the distribution of image features across the direction θ_k, as shown in Figure 9.4 (right) where smaller scales are darker.

Table 9.1 presents results for four different experiments: Cameraman image, 9×9 boxcar v, *BSNR*=40dB (Experiment 1, see Figure 9.3); $v(x_1,x_2) = (1+x_1^2+x_2^2)^{-1}$, $x_1,x_2 = -7,\dots,7$, $\sigma^2 = 2$ (Exp.2) or $\sigma^2 = 8$ (Exp.3), and Lena image, v is a 5×5 separable filter with the weights $[1,4,6,4,1]/16$ in horizontal and vertical directions, *BSNR*=15.93dB (Exp.4). For these experiments a set

Method Experiment	1	2	3	4
Anisotropic *LPA-ICI* [46]	8.23	7.78	6.04	3.76
GEM (Dias) [10]	8.10	7.47	5.17	–
EM (Figueiredo and Nowak) [18]	7.59	6.93	4.88	2.94
ForWaRD (Neelamani et al.) [75]	7.30	6.75	5.07	2.98

Table 9.1: *ISNR* (dB) of the proposed algorithm and of methods [10], [18] and [75] for the four experiments.

of eight directions, $\{\theta_k\}_{k=1}^{8} = \{0, \pi/4, \pi/2, \ldots, 7/4\pi\}$ and five scales, $\#H = 5$, are used. Function estimation kernels were designed on conically-supported windows choosing the *LPA* orders $m = (1,0)$ and $m = (0,0)$ for the *RI* and *RWI* stages, respectively. These kernels are shown in Figure 9.2. For smaller scales in H the $\operatorname{supp} w_h$ is a 1-pixel-width line.

Overall, the *SNR* improvement (*ISNR*) in Table 9.1 shows that the new developed *RWI* algorithm demonstrates a good performance and outperforms some state-of-the-art techniques. Visual inspection is also in favor of the new algorithm. Figure 9.4 (left) shows a fragment of the restored Cameraman image.

The directionality of the kernels is an important element of this good performance. For example, in the same algorithm non-directional quadrant kernels give *ISNR*=7.52dB for Exp.1 (see [44]) versus *ISNR*=8.23dB in Table 9.1.

9.1.3 Derivative estimation and edge detection from noisy blurred observations

As a further illustration of the flexibility of our approach we present two examples of differentiation of y using the noisy blurred observations. Let us replace in the *RWI* stage of the algorithm (9.3) the smoothing kernels $g_{h,\theta_k}^{(0,0)}$ by the discrete derivative-estimation kernels $g_{h,\theta_k}^{(1,0)}$. Then the output $\hat{y}_{h^+,\theta_k}^{RWI}$ of the two stage algorithm gives the estimate of the directional right-hand derivative $\partial_{+\theta_k} y$. Figure 9.5 (left) shows the diagonal derivative estimate $\hat{\partial}_{\theta_2}$ calculated for $\theta = \pi/4$ as the mean of the two one-sided directional derivatives with $\theta_2 = \pi/4$ and $\theta_5 = \theta_2 + \pi = 5\pi/4$, $\hat{\partial}_{\theta_2} = (\hat{y}_{h^+,\theta_2}^{RWI} - \hat{y}_{h^+,\theta_5}^{RWI})/2$.

Further, for the edge detection, we calculate the sum of the absolute values of these derivatives $\sum_{k=1}^{4} |\hat{\partial}_{\theta_k}|$. The image of this sum is shown in Figure 9.5 (right). It demonstrates a very accurate recovery of the image edges from the blurred noisy image data. Observe that the in the above formulas concerning the combination of different derivatives there is no weighting by the inverse of the variance. Exactly as it is done in Section 7.2, and in particular in Figure 7.13, the absolute values of the different derivatives are summed together in order to reveal the edges.

9.2 Poisson deconvolution

A spatially adaptive image deblurring algorithm is presented for Poisson observations. The *RI-RWI* algorithm is modified in such a way that the signal-dependant characteristics of the Poissonian noise can exploited. This allows to accurately compute the pointwise variances of the directional estimates.

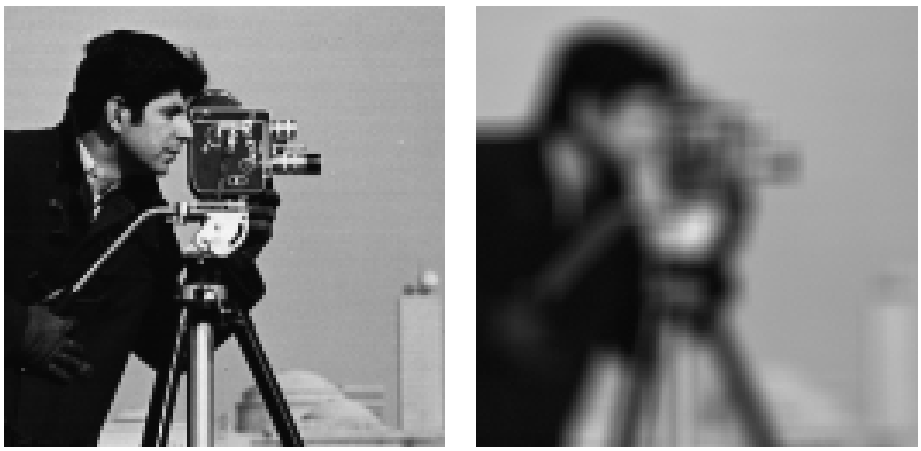


Figure 9.3: Original Cameraman image (left) and noisy blurred observation (Experiment 1) (right)

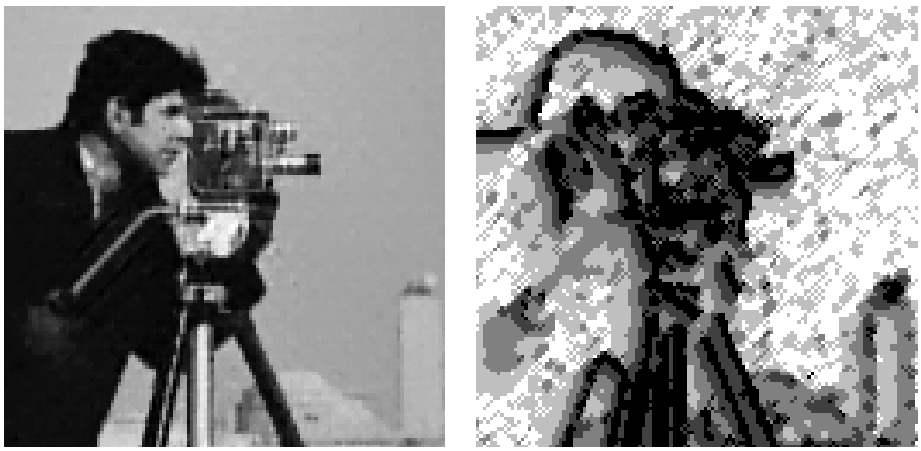


Figure 9.4: Anisotropic *LPA-ICI* deconvolution algorithm performance: restored image, *ISNR*=8.23dB (left) and adaptive scales scales $h^+(\cdot, \pi/4)$ (right)

This section is essentially based on [23][2].

9.2.1 Introduction

In many imaging systems the recorded observations have the physical meaning of numbers of detected photons. The photons are counted at different spatial locations and in this way form an image of an object. This sort of scenario is typical for many imaging problems in medicine, including positron and single-photon emission tomography, in gamma astronomy, microscopy, and photon-limited optical imaging. The Poisson distribution is the conventional probabilistic model for the random number of photons detected during an exposure time. An im-

[2] [23]: Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", (accepted) *IEEE 2005 Int. Conf. Image Processing, ICIP 2005,* September 2005.

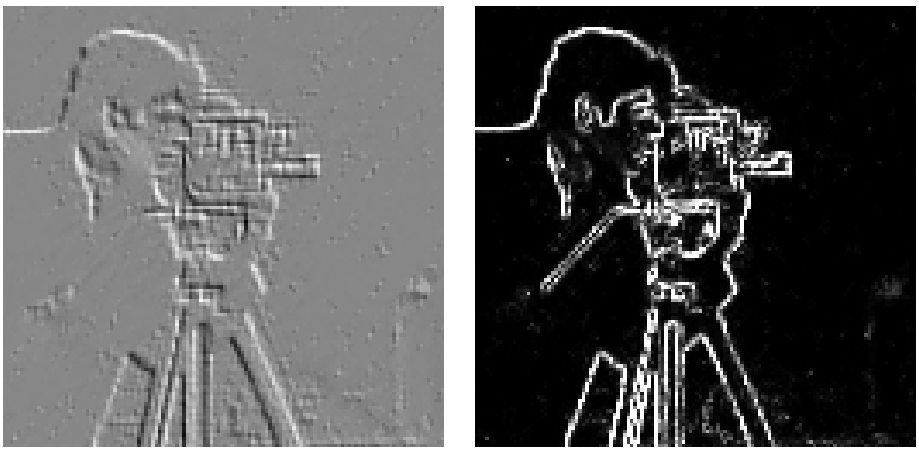


Figure 9.5: Directional derivative (left) and edge detection (right)

portant consumer application where Poissonian distributions dominate are the widespread CCD/CMOS-sensor digital cameras (e.g. [93]).

An optical blurring is typically introduced into the observation process. This distortion of the image is commonly modeled by the convolution $(y \circledast v)(x)$ of the true image y with the point-spread function (PSF) v of the optical system. It is assumed that the observations $z(x)$ are Poissonian, according to the model

$$z\left(x\right) \sim \mathcal{P}\left(\left(y \circledast v\right)\left(x\right)\right), \tag{9.4}$$

where $\mathcal{P}$ denotes the Poisson distribution. This model means that $E\{z(x)\} = (y \circledast v)(x)$ and $\sigma_z^2\left(x\right) = var\{z(x)\} = (y \circledast v)(x)$. Thus, the observation variance $\sigma_z^2\left(x\right)$ is signal dependent and, consequently, spatially variant. In our approach we make explicit use of this variance function to reconstruct the image y from the noisy observations z. Observe that (9.4) can be rewritten in the additive form $z\left(x\right) = \left(y \circledast v\right)\left(x\right) + \eta\left(x\right)$, where the noise term $\eta\left(x\right)$ has zero mean and variance $\sigma_\eta^2\left(x\right) = \left(y \circledast v\right)\left(x\right)$.

9.2.2 Poissonian *RI-RWI* algorithm

For the AWGN model, the *RI-RWI* estimates (9.2)-(9.3) are computed completely in the Fourier domain. For the Poissonian case there are some modifications. The main problem is that variance of the estimates is spatially varying (noise variance is not constant over the image). This makes necessary the computation of the pointwise-varying variance of each of the estimates. Also some slight changes in the form of the Wiener denominator are required.

9.2.3 Linear inverse with directional adaptive *LPA-ICI* filtering

This algorithm uses the nonparametric *regularized inverse* (*RI*) and *regularized Wiener inverse* (*RWI*) *LPA-ICI* deconvolutions developed for the Gaussian inverse in [45],[46] and for inverse halftoning (colored noise) in [21]. For the

Gaussian noise case, the filtering was performed completely in the Fourier domain, according to (9.2) and (9.3). For the Poissonian case there are some modifications. The main problem is that, as the observation variance $\sigma_z^2(x)$ is not constant, the standard deviations of the directional estimates are spatially varying. It makes to compute a pointwise-varying variance for each of the estimates. Secondly, some change in the form of the Wiener denominator is required, with the constant σ^2 replaced by a correct estimate of the Poissonian noise power spectrum.

In order to calculate all these elements efficiently, a mixed space/frequency domain approach is exploited. Let us start from the regularized inverse stage.

9.2.4 Poissonian *RI* inverse

The actual regularized inversion is performed in the frequency domain, and then the *LPA* filtering is performed as a convolution of the pure regularized inverse z^{RI} against the *LPA* kernel $g_{h,\theta}$ in the spatial domain:

$$T^{RI}(f) = \frac{V(-f)}{|V(f)|^2 + \varepsilon_1^2}, \qquad t^{RI} = \mathcal{F}^{-1}(T^{RI}), \tag{9.5}$$

$$z^{RI} = \mathcal{F}^{-1}(T^{RI} Z), \qquad \hat{y}^{RI}_{h,\theta} = z^{RI} \circledast g_{h,\theta}. \tag{9.6}$$

Estimation of the standard deviation of the *RI-LPA* estimates (needed for the *ICI* adaptive-scale selection and for the fusing of the directional adaptive-scale estimates) is also calculated in a mixed frequency/space domain. The variance of $\hat{y}^{RI}_{h,\theta}$ is obtained as

$$\sigma^2_{\hat{y}^{RI}_{h,\theta}} = \mathcal{F}^{-1}\left(\mathcal{F}\left(\left(t^{RI} \circledast g_{h,\theta}\right)^2\right) \cdot \Sigma_z^2\right), \tag{9.7}$$

where $\Sigma_z^2 = \mathcal{F}\left(\sigma_z^2\right)$ is the Fourier transform of the space-varying variance of z. Here σ_z^2 is estimated directly from the noisy observations, i.e. $\hat{\sigma}_z^2 = z$ and $\Sigma_z^2 = Z$. This is the simplest possible unbiased estimate of the variance, accordingly to the Poissonian rule $E\{z\} = var\{z\}$.

All the varying scale estimates $\{\hat{y}^{RI}_{h,\theta}\}_{h\in H}$ obtained for each θ are fed (together with their standard deviations $\{\sigma_{\hat{y}^{RI}_{h,\theta}}\}_{h\in H}$) into the *ICI* algorithm, which selects the pointwise-adaptive scale $h^+(x,\theta)$. This is done *independently* for each direction θ. In this way, the adaptive-scale directional estimates $\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}$, $k = 1,\ldots,K$, are constructed.

Fusing these directional estimates is done using the inverse variances as weights in the convex combination

$$\begin{aligned} \hat{y}^{RI}(x) &= \sum\nolimits_k \lambda_k^{RI}(x)\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}(x), \\ \lambda_k^{RI}(x) &= \sigma_k^{RI-2}(x)/\textstyle\sum_i \sigma_i^{RI-2}(x), \\ \sigma_i^{RI-2}(x) &= 1/\sigma^2_{\hat{y}^{RI}_{h^+(x,\theta_i),\theta_i}}(x). \end{aligned} \tag{9.8}$$

The final estimate of the *RI* stage is the anisotropic $\hat{y}^{RI}$. The anisotropy of this estimate is a direct consequence of the selection of an adaptive scale for each direction.

The use of the space domain convolutions (9.6) and (9.7) instead of multiplications in Fourier domain can speed-up calculations significantly, since the

support of the directional-*LPA* kernels $g_{h,\theta}$ is usually very small. Moreover, this choice allows more freedom in the handling of the boundary conditions. Observe that the formula for the variance (9.7) can be rewritten easily in the standard convolution form $\sigma^2_{\hat{y}^{RI}_{h,\theta}} = \left(\mathcal{F}^{-1}\left(T^{RI}G_{h,\theta}\right)\right)^2 \circledast \sigma^2_z$.

9.2.5 Poissonian *RWI* inverse

The regularized Wiener inverse algorithm proceeds similarly:

$$T^{RWI}(f) = \frac{V(-f)|Y(f)|^2}{|V(f)Y(f)|^2 + \varepsilon_2^2 \Phi_\eta(f)}, \quad t^{RWI} = \mathcal{F}^{-1}(T^{RWI}), \tag{9.9}$$

$$z^{RWI} = \mathcal{F}^{-1}(T^{RWI}Z), \qquad \hat{y}^{RWI}_{h,\theta} = z^{RWI} \circledast g_{h,\theta}. \tag{9.10}$$

Here, Φ_η is the power spectrum of the noise. It can be shown that for Poissonian observations Φ_η is constant and equal to the spatial mean of $E\{z\}$ over the image domain. As $E\{z\} = y \circledast v$ is unknown, its value may be estimated as $\hat{y}^{RI} \circledast v$. However, since $E\{\eta(x)\} = 0$, we simply set $\Phi_\eta = mean_x(z)$. This is an accurate approximation of $mean_x(E\{z\})$ for large size images.

The final fused estimate of the *RI* stage, $\hat{y}^{RI}$, is used quite naturally as a "pilot" estimate in the Wiener filtering. It means that $|Y|^2$ in (9.9) is replaced by $|\hat{Y}^{RI}|^2$.

Similarly to the regularized inverse stage, also the standard deviations of the *RWI-LPA* estimates are calculated in mixed frequency/space domain. Again, the variance of $\hat{y}^{RWI}_{h,\theta}$ is obtained as

$$\sigma^2_{\hat{y}^{RWI}_{h,\theta}} = \mathcal{F}^{-1}\left(\mathcal{F}\left(\left(t^{RWI} \circledast g_{h,\theta}\right)^2\right) \cdot \Sigma_z^2\right).$$

In this second stage, σ_z^2 is estimated more accurately than in the previous one (in order to get a better estimate for Σ_z^2), from the regularized inverse estimate: $\hat{\sigma}_z^2 = \hat{y}^{RI} \circledast v \simeq y \circledast v = \sigma_z^2$. Then, the *ICI* rule selects the pointwise-adaptive-scale estimate $\hat{y}^{RWI}_{h^+(x,\theta),\theta}(x)$, for every x, and for each specified direction θ.

The fusing procedure is performed exactly as for the *RI*, with

$$\begin{aligned} \hat{y}^{RWI}(x) &= \sum\nolimits_k \lambda_k^{RWI}(x)\hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}(x), \\ \lambda_k^{RWI}(x) &= \sigma_k^{RWI-2}(x) / \sum\nolimits_i \sigma_i^{RWI-2}(x), \\ \sigma_i^{RWI-2}(x) &= 1/\sigma^2_{\hat{y}^{RWI}_{h^+(x,\theta_i),\theta_i}}(x). \end{aligned}$$

The final output of the two-stage Poissonian *RI-RWI* is the anisotropic adaptive estimate $\hat{y}^{RWI}$.

9.2.6 Comments

In general, the regularized inverse and regularized Wiener inverse are linear filters which actually are not appropriate to the problem with the varying signal dependent observation variance. In particular, even the *ideal* Wiener filter, which is obtained by setting $\varepsilon_2^2 = 1$ in (9.9) and by using the "oracle" estimates for $|Y|$ and Φ_η, achieves quite a poor performance, as shown in Figure 9.7(right). Main reason is that the Wiener filter itself is not able to produce a

global estimate fitting nonstationary varying-variance observations[3]. However, the directional *RI* and *RWI* filters generate sets of estimates rich enough to select from, and the *ICI* efficiently performs this adaptive selection.

The anisotropic fusing (9.8) of these adaptive estimates for various directions yields a remarkable improvement in the restoration.

The presented algorithm can be modified, so to be used for restoration from signal-dependant noises other than the Poissonian one. Moreover, if the randomness of the noise is particularly high, the first stage can be executed once or more times again in order to refine the estimate of σ_z^2 by using a feedback mechanism similar to the one described in Section 8.3.1.

9.2.7 Numerical experiments

In our simulations, in order to achieve a desired level of randomness (i.e. desired *SNR*) in the noisy Poissonian observations, we first multiply the true signal y^{TRUE} (which has range [0,1]) by a scaling factor $\chi > 0$: $y = \chi \cdot y^{TRUE}$, $z \sim \mathcal{P}(y \circledast v)$. Thus, $E\{z\} = \sigma_z^2 = \chi \cdot y^{TRUE} \circledast v$, and $E\{z\}/std\{z\} = \sqrt{\chi}\sqrt{y^{TRUE} \circledast v}$, i.e. better *BSNR* (*SNR* of the blurred observation against its expectation) corresponds to larger χ.

We consider a deblurring experiment similar to the one considered in the previous section. The *Cameraman* image is heavily blurred by a 9×9 "boxcar" uniform PSF and degraded by noise, with a *BSNR*=32.5dB. The PSF v is assumed to be known. To create a noisy Poissonian distributed observation with that *BSNR*, the parameter χ=17600 is selected. Despite so large value of χ, the non-uniformity of the noise is still quite an essential issue for the Poisson deblurring, as the following simulations show. The actual values of the standard deviations σ_z are in the range of $[0, 0.0075]$ (assuming that the image is renormalized back to the range $[0, 1]$). It is interesting to note that this level of randomness is as much as what can be observed in images taken with a consumer-level CMOS[4] sensor under normal light conditions.

The proposed *RI-RWI* adaptive algorithm is implemented with the following parameters. As in [46], a set of eight directions, $\{\theta_k\}_{k=1}^{8} = \{0, \pi/4, \pi/2, \ldots, 7/4\pi\}$ and five scales, $\#H = 5$, are used. Function estimation kernels were designed on conically-supported windows choosing the first and zero *LPA* orders for the *RI* and *RWI* stages, respectively. For smaller scales in H the kernel support is a 1-pixel-width line. The *ICI* thresholds and regularization parameters for the *RI* and *RWI*, are $\Gamma_{\text{RI}} = 1.5$, $\Gamma_{\text{RWI}} = 1.4$, $\varepsilon_1 = 0.03$ and $\varepsilon_2 = 0.28$.

Figure 9.6 shows details of the blurred Poisson noisy observation and the reconstructed *Cameraman* image. The reconstruction is visually quite good, with most of the details properly restored and no significant distortions. The objective values of *ISNR* and *RMSE* are given in Table 9.2. Figure 9.8 shows the adaptive scales selected by the *ICI* for a vertical direction from the *RI* and a horizontal direction from the *RWI* stage of the algorithm. It is remarkable how these scales reveal the features of the image across the corresponding direction.

To demonstrate the improvement arising from our modified algorithm, we compare it against the standard Gaussian version of Section 9.1. First, we re-

[3] Nevertheless, linear Wiener filters have been used quite extensively for the restoration of blurred images with Poissonian and more generally, signal-dependant noise (e.g. [17],[59]), mostly because of their lower complexity and good stability.

[4] Raw data from Nokia 6600 camera phone.

Algorithm	*ISNR*	*RMSE*
Poissonian anisotropic LPA-ICI RI-RWI	**6.61**	**0.0428**
Optimized *Gaussian LPA-ICI RI-RWI*	6.03	0.0458
Gaussian LPA-ICI RI-RWI [46]	5.38	0.0493

Table 9.2: *MSE* and *ISNR* (dB) for *Cameraman* image for Poisson image reconstruction.

Figure 9.6: Deblurring the *Cameraman* image. Left is a fragment of the noisy blurred observation (*BSNR*=32.5dB). Right is the reconstructed image obtained by the proposed Poissonian adaptive deconvolution algorithm, *ISNR*=6.61dB.

store the image applying the algorithm in a straightforward manner, estimating the noise using a *MAD* estimator (it gives constant $\hat{\sigma} = 0.0045$), and using the standard parameters that were optimized for the Gaussian case. Second, we tune the parameters, in order to compensate to the wrong noise model assumed by algorithm, trying to obtain the best possible restoration. Results are shown in the Table and in Figure 9.9. Numerically, both results obtained by the Gaussian algorithms are worse than the one obtained with the algorithm specifically designed for the Poissonian data. Comparing images in Figure 9.9 we may note the enhancement obtained by the parameter optimization. A further comparison with the reconstructed image in Figure 9.6(right) obtained by the algorithm developed for the Poissonian data demonstrates an obvious visual advantage of the proposed algorithm.

9.3 Inverse halftoning

In this section, which is substantially based on [21][5], an original inverse halftoning algorithm for restoring a continuous tone image from a given error-diffusion halftone image is presented. The algorithm is based on the anisotropic deconvolution strategy introduced earlier in this chapter. The linear model of error diffusion halftoning proposed by Kite et al. [55] is exploited. It approximates error diffusion as the sum of the convolution of the original grayscale image

[5] [21]: Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "Inverse halftoning based on the anisotropic LPA-ICI deconvolution", *Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Proc., SMMSP 2004, Vienna, Austria*, pp. 49-56, September 2004.

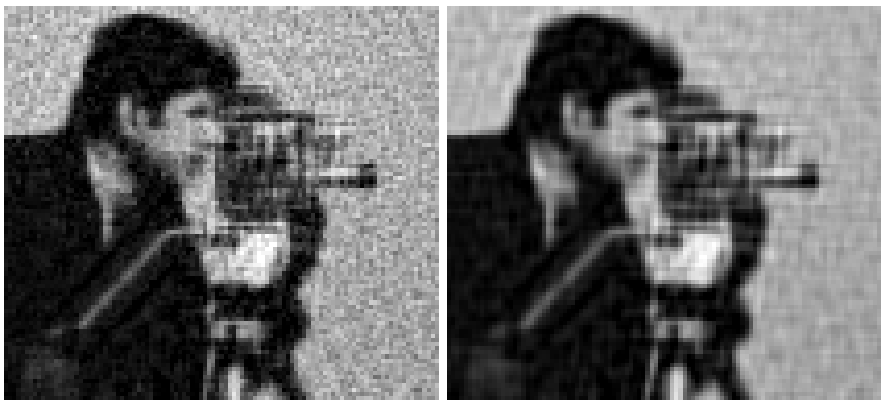


Figure 9.7: Result of the pure regularized inverse z^{RI} from (9.6) (left) and the "oracle" Wiener estimate, $ISNR$=5.22dB (right).

Figure 9.8: Adaptive scales: RI $h^+(\cdot, \pi/4)$ (left), and RWI $h^+(\cdot, 0)$ (right). Darker color represents smaller scales.

with a specific kernel and colored random noise. Under this model the inverse halftoning can be therefore formulated as a special deconvolution problem.

The deconvolution is performed following the *RI-RWI* (regularized inverse-regularized Wiener inverse) scheme and exploiting the anisotropic *LPA-ICI* estimator. This adaptive varying scale estimator, based on the directional-*LPA* technique and the *ICI* scale-selection algorithm, allows near optimal edge adaptation. As a result, the reconstructed continuous-tone image presents smooth areas faithful to the unknown original and yet preserves all the details found in the halftone. Conventional inverse-halftoning algorithms often produce estimates that are either oversmooth (loss of details) or still noisy.

The simulation experiments reported at the end of this section confirm the state-of-the-art performance of the proposed algorithm, both visually and in mean-squared-error sense.

Figure 9.9: Filtering the Poisson data by the algorithm developed for the Gaussian one. Standard selection of the algorithm parameters gives a poor estimate, $ISNR$=5.38dB (left). Up to some extent, it can be improved by manually optimizing some algorithm parameters, $ISNR$=6.03dB (right).

Figure 9.10: An illustration of halftoning and inverse halftoning. Detail of the *Lena* image: original (left), Jarvis error-diffusion halftone (center), and *LPA-ICI* estimate ($PSNR$=33.0dB) (right).

9.3.1 Halftoning and inverse halftoning

In the last two decades the color-depth of digital images, graphic cards, computer displays and digital cameras has steadily increased. The current standard for consumer devices is 8 or more bits for each color channel. In particular, for grayscale images this is equivalent to 256 or more different intensity values. Since the human eye is usually not able to distinguish between so close adjacent shades of gray, such grayscale images are often called *continuous-tone* images. Coarser palettes are nowadays considered only for lossy image/video compression applications.

Despite this progress, many output and rendition devices are still unable to reproduce these continuous tone shades and can provide only a binary (black-and-white) output. Typical examples of such devices are office and industrial printers but also low-cost displays for mobile devices.

Digital halftoning is the rendition process of a continuous tone into a binary image. Although the naive approach where shades lighter or darker than a 50% gray level are thresholded, respectively, to white or black, is the simplest to implement, it is almost never used because of its visually poor result on photographic images. Taking into account the characteristics of the human visual system, which acts as low-pass filter, halftones are generated in such a way

that the difference between the halftoned binary image and the original grayscale image is compacted into the high frequency end of the Fourier spectrum.

Halftoning techniques include ordered dithering or screening (dispersed-dot and clustered-dot), error diffusion, blue-noise dithering [92], and direct binary search [1]. The latter is known to provide the highest quality halftones. However the most widely used methods, because of their computational efficiency, are order dithering and error diffusion. Figure 9.10 illustrates the halftoning process.

Halftoned images may look good printed on paper. However, due to their high frequency characteristics they cannot be used in many situations. For example, scanning and reprinting a high-resolution halftone would result in a poor quality output (see, for example, the left column of Figure 9.16). Halftones, when displayed on a computer screen (which has a resolution significantly inferior to that of printer) present evident aliasing artifacts. Further processing, such as resizing or contrast enhancement, can severely degrade the image quality. Moreover, standard compression techniques are not able to process halftones efficiently. The development of applications such as high-quality digital archive of old newspapers or scientific journals can thus still be considered as challenging tasks. In all these cases it would be desirable to process, whenever available, the original grayscale image rather than the black and white halftone.

Inverse halftoning is the reconstruction process of a continuous tone image from its binary halftone, as illustrated in Figure 9.10. It is clear, from the above discussion, that inverse halftoning should mimick the human visual system. Thus, all inverse halftoning techniques perform some sort of low-pass filtering. A fixed-kernel low-pass filtering is simple to implement, nevertheless very seldom yields satisfactory results. In the recent years inverse halftoning has gained renewed interest and several new adaptive methods have been proposed [98]. They include thresholding in transform domain [73, 74], projection onto convex sets (POCS) [34, 4], MAP projection [97], anisotropic diffusion [56] and look-up tables (LUT) based on learning/training [71, 72, 52].

In what follows, we describe a novel inverse halftoning technique, which combines a linear model for error diffusion [56] and the proposed anisotropic deconvolution scheme based on the regularized inverse-regularized Wiener inverse (*RI-RWI*) *LPA-ICI* from Section 9.1.2. We assume that the error diffusion kernel is known. In particular, we show simulation results obtained for the Floyd-Steinberg [19] and Jarvis et al. [38] error diffusion kernels.

Just as for the traditional image deblurring problem, also for inverse halftoning the anisotropic *LPA-ICI*-based deconvolution yields state-of-the-art performance through a two stage, non-iterative, filtering procedure where blur and noise are simultaneously removed. The anisotropy of the proposed estimator allows to restore accurately edges and details, producing a result quite faithful to the original.

9.3.2 Error diffusion

Roughly speaking, the error-diffusion halftoning works by raster-scanning the continuous-tone image and recursively distributing, or "diffusing", the quantization errors due to binarization on the neighboring pixels.

Let y be the original continuous tone image, x the pixel coordinate and z

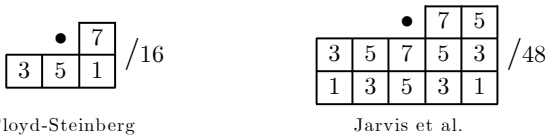


Figure 9.11: Error filters h^{ed} for the Floyd-Steinberg [19] (left) and Jarvis [38] error diffusions. The black dot indicates the center of the kernel.

the halftoned image (to be generated); after setting the initial conditions [19]

$$\begin{aligned} &\tilde{y}_1 = y, \\ &x_1 = (1,1) \qquad \text{(start from top-left pixel)}, \end{aligned}$$

error diffusion is precisely defined by the following iterative procedure:

$$\begin{cases} z\left(x_n\right) &= \left[\tilde{y}_n\left(x_n\right)\right]_{0,1}; \\ e_n &= \tilde{y}_n\left(x_n\right) - z\left(x_n\right); \\ \tilde{y}_{n+1}\left(x_n + x\right) &= \tilde{y}_n\left(x_n + x\right) + e_n h^{\mathrm{ed}}\left(x\right) \quad \forall x; \\ x_{n+1} &= \text{successor}\left(x_n\right); \end{cases}$$

where h^{ed} is a weight kernel, $\left[\,\cdot\,\right]_{0,1}$ is the binarization (or rounding) operation (i.e. $[\tilde{y}]_{0,1} = 1$ iff $\tilde{y} \geq \frac{1}{2}$, otherwise $[\tilde{y}]_{0,1} = 0$) and "successor" denotes the next pixel to be processed in raster-scanning.

In other words, at every step, the error-diffusion algorithm

$$\begin{cases} \text{binarizes the current pixel (i.e. rounding to } \{0,1\}); \\ \text{computes the quantization error;} \\ \text{diffuses error on neighboring pixels using weights from } h^{\mathrm{ed}}; \\ \text{moves to the next pixel (in raster-scanning);} \end{cases} .$$

The kernel h^{ed} is called the *error filter*. Examples of error filters are shown in Figure 9.11. Observe that the weights are non-zero only for those pixels that have not been already scanned. It means that the diffusion never goes backwards with respect to the scanning direction and after a pixel has been binarized its value is not modified by future iterations. The algorithm ends when the bottom-right pixel has been processed. The diffusion of the quantization error guarantees that the local averages of the halftoned z are close to the corresponding local averages of the continuous tone y.

Although the iterative nature of the procedure restricts its computational speed, on the other hand the simplicity of the iteration step, the negligible memory footprint, and the excellent rendition quality made error diffusion one of the most established halftoning techniques.

Several modifications to the above procedure (such as different pixel-scan ordering or threshold modulation) are possible [92].

9.3.3 Linear model of error diffusion

In [54] Kite et al. propose the following linear model as an approximation of error diffusion halftoning. Let Y and Z be the Fourier transforms of y and z, respectively. Then

$$Z = PY + Q\eta, \tag{9.11}$$

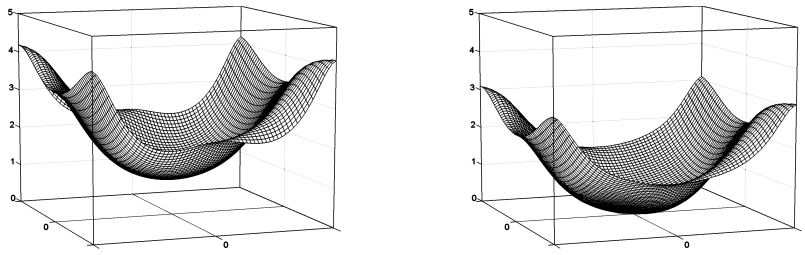


Figure 9.12: Absolute value of P (left) and Q (right) corresponding to the Floyd-Steinberg error filter.

where η is white Gaussian noise (with standard deviation σ),

$$P = \frac{K^{\text{gain}}}{1 + (K^{\text{gain}} - 1)\, H^{\text{ed}}}, \qquad Q = \frac{1 - H^{\text{ed}}}{1 + (K^{\text{gain}} - 1)\, H^{\text{ed}}},$$

H^{ed} is the frequency response of the error filter h^{ed} and K^{gain} is a gain constant. K^{gain} is found [54, 55] to be essentially independent on y and depends instead only on the used error filter: for example, $K^{\text{gain}} = 2.0$ and $K^{\text{gain}} = 4.5$ for the Floyd-Steinberg and Jarvis error filters respectively.

Since the typical H^{ed} is a low-pass, Q is a high-pass filter (see Figure 9.12). This is consistent with the fact that error-diffusion halftoned images differ from the continuous-tone original mostly for the high-frequency components of the spectrum (blue noise).

The model (9.11) has been proved to be quite accurate [55], and it has been already exploited in a number of algorithms (e.g. [57], [73], [74]).

Convolutional model

In the spatial domain, multiplications are replaced by convolutions and (9.11) becomes

$$z = p \circledast y + q \circledast \eta \tag{9.12}$$

where p and q are the impulse responses of P and Q respectively.

According to this model, the inverse halftoning process can be formulated as a deconvolution problem, where p is the point-spread function and the observations z are contaminated by the colored blue noise $q \circledast \eta$.

Deconvolution

Just as for the conventional deconvolution problem discussed in the previous sections, an unbiased solution of a deconvolution problem of the form (9.11) or (9.12) can be obtained in a straightforward manner by first inverting the convolution operator P and then removing the noise $P^{-1}Q\eta$.

Unlike deconvolution examples from the previous sections, where the PSD was a low-pass filter, the filter P corresponding to the error-diffusion halftoning has an absolute value always larger that one, as shown in Figure 9.12. Thus, in principle, there should not be need of any regularization for the inversion of the convolution. In practice, however, it is found that – although quite accurate – the model (9.11) is not completely exact, and the regularization can act as

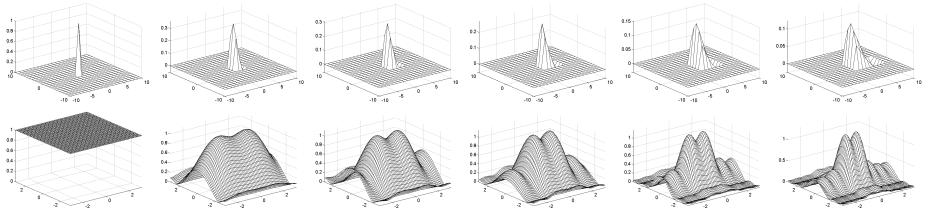


Figure 9.13: Varying-scale directional-*LPA* kernels (top) and the absolute value of their Fourier transforms (bottom); $m = (1, 0)$, $\theta = 0$.

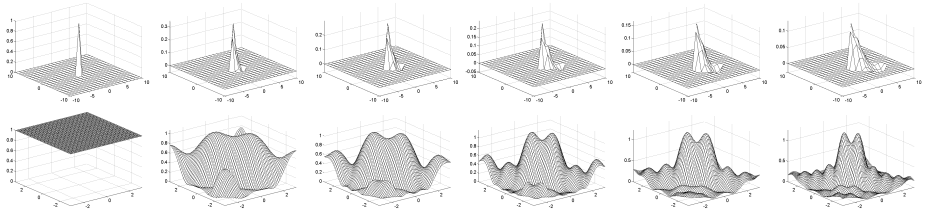


Figure 9.14: Varying-scale directional-*LPA* kernels (top) and the absolute value of their Fourier transforms (bottom); $m = (1, 0)$, $\theta = 7\pi/4$.

an efficient stabilizing device, which can attenuate the distortions due to the imprecisions of the assumed model.

9.3.4 Anisotropic *LPA-ICI* inverse-halftoning

We approach the inverse-halftoning problem according to the above convolutional model, using the anisotropic *LPA-ICI RI-RWI* to perform the regularized deconvolution.

Kernels

A collection of compactly supported directional-*LPA* kernels $\left\{g_{h_j,\theta_k}\right\}_{h_j\in H,k=1,\ldots,K}$ has been designed specifically for the inverse halftoning problem. As usual, each kernel is characterized by a direction θ_k and a scale parameter h_j, and for each fixed θ_k, $\{g_{h_j,\theta_k}\}_{h_j\in H}$ is a family of varying-scale directional kernels. Figure 9.13 and Figure 9.14 show two of these families of kernels corresponding to two of the eight specified directions.

Adaptive deconvolution algorithm

Analogously to the previous sections, capital letters are used for the discrete Fourier transform of the corresponding functions. We denote by $\overline{P}$ the complex conjugate of P. The considered technique is based on the following regularized inversion (*RI*) and regularized Wiener inversion (*RWI*) estimates, using the

directional-*LPA* kernels g_{h_j,θ_k} :

$$\hat{Y}^{RI}_{h_j,\theta_k} = \frac{\overline{P}G_{h_j,\theta_k}}{|P|^2 + |Q|^2 \varepsilon_1^2} Z, \qquad (RI) \tag{9.13}$$

$$\hat{Y}^{RWI}_{h_j,\theta_k} = \frac{\overline{P}|Y|^2 G_{h_j,\theta_k}}{|PY|^2 + \varepsilon_2^2 |Q|^2 \sigma^2} Z. \qquad (RWI) \tag{9.14}$$

The adaptive procedure assumes that the estimates $\{\hat{y}^{RI}_{h_j,\theta_k}\}_{h_j \in H}$ are calculated according to (9.13) for a set of scales H and the *ICI* rule selects the adaptive scales for each direction and for each pixel. In this way we obtain the directional varying scale adaptive estimates $\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}$, $k = 1, \ldots, K$, which are fused in the anisotropic $\hat{y}^{RI}$ according to (4.8)

$$\begin{aligned} \hat{y}^{RI}(x) &= \sum\nolimits_k \mathcal{K}^{RI}_k(x) \hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}(x), \\ \mathcal{K}^{RI}_k(x) &= \sigma^{RI-2}_k(x) / \sum\nolimits_i \sigma^{RI-2}_i(x), \end{aligned} \tag{9.15}$$

where $\sigma^{RI}_k(x)$ is the standard deviation of the adaptive scale estimate $\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}(x)$.

The fused $\hat{y}^{RI}$ serves as the reference signal in the *RWI* procedure (see Figure 9.1). The adaptive *RWI* algorithm is similar and gives the *ICI* adaptive varying scales estimates $\hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}$ for each direction and x. Then, the final estimate $\hat{y}^{RWI}$ is obtained by fusing these directional ones again similarly to (9.15):

$$\begin{aligned} \hat{y}^{RWI}(x) &= \sum\nolimits_k \mathcal{K}^{RWI}_k(x) \hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}(x), \\ \mathcal{K}^{RWI}_k(x) &= \sigma^{RWI-2}_k(x) / \sum\nolimits_i \sigma^{RWI-2}_i(x). \end{aligned} \tag{9.16}$$

The final estimate of y of the proposed inverse halftoning algorithm is the output given by the *RWI* deconvolution scheme (9.3) that uses the *ICI*-based *RI* estimate as a reference signal Y. Thus, we arrive to the two steps procedure shown in Figure 9.1.

Remarks

The *ICI* adaptive scales $h^+(\cdot, \theta_k)$ represent the distribution of image features across the direction θ_k, as shown in Figure 9.15 (in the figure, darker color corresponds to smaller scales).

The variances of the estimates $\hat{y}^{RI}_{h_j,\theta_k}$ and $\hat{y}^{RWI}_{h_j,\theta_k}$ are obtained, respectively, as

$$\begin{aligned} \sigma^{RI\,2}_{h_j,\theta_k} &= \sigma^2 \left\| \frac{\overline{P}G_{h_j,\theta_k}Q}{|P|^2 + |Q|^2 \varepsilon_1^2} \right\|_2^2, \\ \sigma^{RWI\,2}_{h_j,\theta_k} &= \sigma^2 \left\| \frac{\overline{P}|Y|^2 G_{h_j,\theta_k}Q}{|PY|^2 + \varepsilon_2^2 |Q|^2 \sigma^2} \right\|_2^2, \end{aligned}$$

where $\|\cdot\|_2$ denotes the l^2-norm, and σ^2 is the variance of the noise η in formula (9.11), which is assumed to be equal to 1.

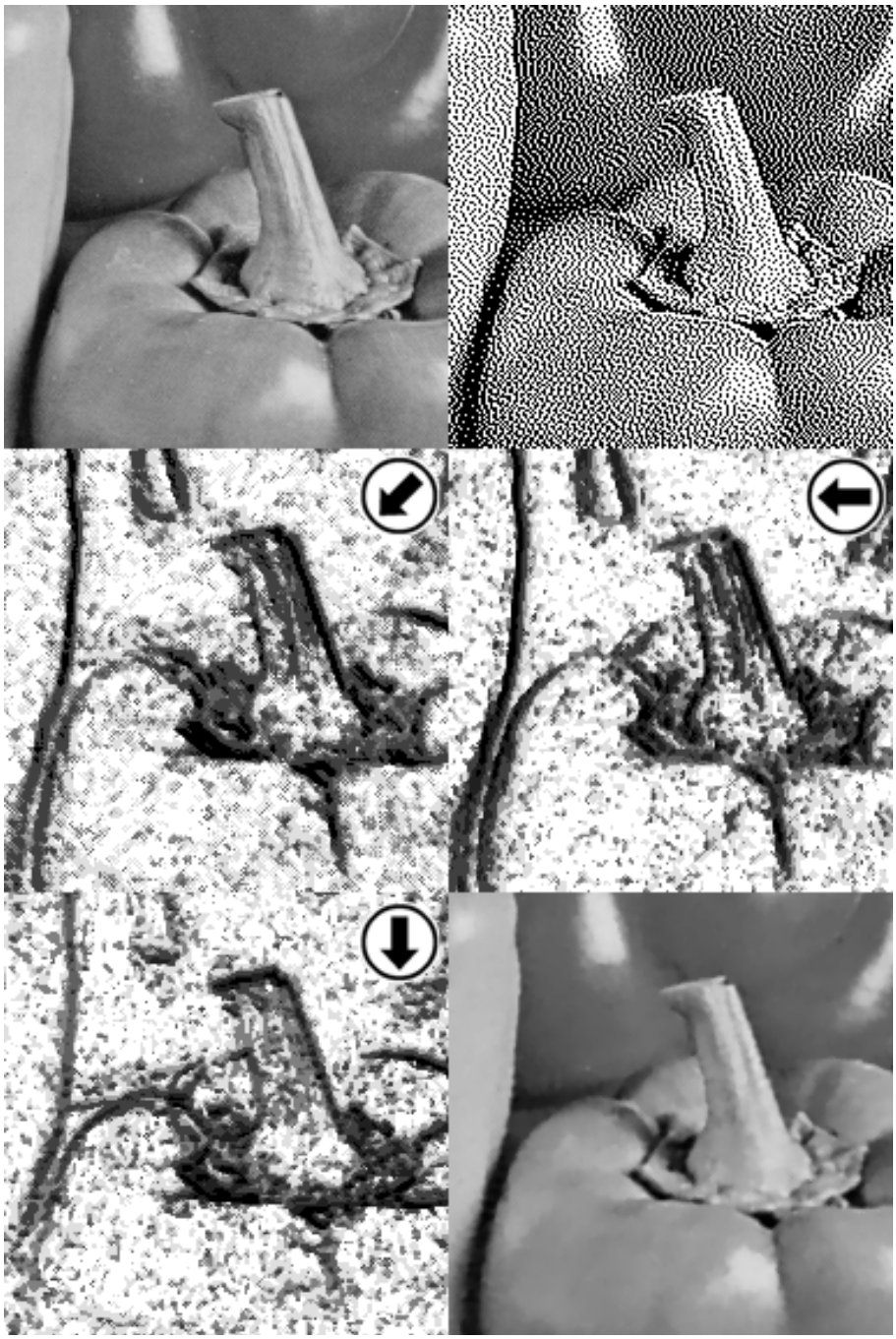


Figure 9.15: *Peppers* (detail): original image (top left), Jarvis error diffusion halftone (top right), adaptive scales $h^+(\cdot, \pi/4)$ (center left), $h^+(\cdot, 0)$ (center right), $h^+(\cdot, \pi/2)$ (bottom left), and *LPA-ICI* estimate (*PSNR*=31.6dB) (bottom right). The arrows indicate the orientation of the kernels $\tilde{g}_{h^+(x,\theta_k),\theta_k}$.

9.3.5 Simulation results

The directional-*LPA* kernels were designed on asymmetrical windows oriented along eight directions, $\{\theta_k\}_{k=1}^{8} = \{0, \pi/4, \pi/2, \ldots, 7/4\pi\}$, with orders $m = (1, 0)$ and $m = (0, 0)$ for the *RI* and *RWI* filters, respectively. A set of 6 and 9 scales was used for the *RI* and *RWI*, respectively. Some of these kernels are shown in Figure 9.13 and 9.14. The first and smallest scale is always equal to 1, i.e. the kernel is the discrete Dirac delta function.

The regularization parameters $\varepsilon_1, \varepsilon_2$, the *LPA* kernels g_{h,θ_k} and the *ICI* threshold Γ are considered as fixed design parameters of the proposed inverse halftoning algorithm. For all results and figures presented in this section one unique set of design parameters has been used.

Although relatively little investigation has been done in the optimization of these design parameters, the anisotropic *LPA-ICI* inverse halftoning delivers already more than satisfactory results (see Figure 9.16). Overall, the *PSNR* values in Table 9.3 show that the new developed algorithm demonstrates a good performance and outperforms some state-of-the-art techniques. Visual inspection is also in favor of the new algorithm. Figure 9.17 shows a fragment of the restored *Lena* image: when compared to the wavelet-based inverse halftoning method [74], the proposed *LPA-ICI* procedure shows its superiority restoring finer details without introducing any visible artifacts.

Inverse halftoning technique	*Lena*	*Peppers*
Anisotropic *LPA-ICI* [21]	**32.4**	**31.6**
WinHD (Neelamani et al.) [74]	32.1	31.2
Wavelet-Vaguelette (Neelamani et al.) [73]	31.9	31.0
Wavelet (Xiong et al.) [99]	31.7	30.7
Gradient (Kite et al.) [56]	31.3	31.4
Kernel (Wong) [97]	32.0	30.3
LUT (Meşe and Vaidyanathan) [72]	31.0	—
LMS-MMSE (Chang et al.) [6]	31.4	31.2
POCS-SVD (Hein and Zakhor) [34]	30.4	—
POCS-Wavelet (Bozkurt and Çetin) [4]	32.2	30.9

Table 9.3: *PSNR* (dB) performance of the proposed algorithm and of other methods for restoration from Floyd-Steinberg error diffusion.

Remark: Recently, in [52] it has been claimed that a decision tree learning LUT algorithm can yield a *PSNR* of 34.75dB for the *Lena* image, sensibly outperforming all previous records of other authors, in particular that of the other LUT-based algorithm [72]. However, we do not include this result in Table 9.3 as it is achieved for an usually sized 1050×1050 image. The results for the table, as well as all the figures in this section, correspond instead to the standard 512×512 images. Nevertheless, we tested our algorithm also for the "oversized *Lena*" used in [52], obtaining a *PSNR* of 37.75dB.

Figure 9.16: Examples of Floyd-Steinberg error diffusion *LPA-ICI* inverse halftoning: *Peppers* (*PSNR*=31.6dB), *Lena* (*PSNR*=32.4dB), and *Boats* (*PSNR*=29.5dB). The pictures in the right column are the estimates obtained by the proposed procedure from the binary halftone images (shown in the left column).

Figure 9.17: Visual comparison for a detail of the *Lena* image: from left to right, original and Floyd-Steinberg halftone (top row), the proposed anisotropic *LPA-ICI* estimate (*PSNR*=32.4), and the "*WInHD*" estimate [74] (*PSNR*=32.1) based on deconvolution and filtering in the complex wavelet domain [53] (central row), and two results obtained of using simple space-invariant Gaussian filters of different variance, clearly exposing the inadequateness of a non-adaptive filtering (bottom row).

Chapter 10

Other applications

In this chapter we briefly present two more applications in which the developed filtering strategy may play a significant role. The aim is to further illustrate the potential of the anisotropic *LPA-ICI* technique, and to provide ground for a few more remarks on this method. No optimization or adaptation of the standard algorithms for the following applications has been thoroghly done, and no comparison with state-of-the-art techniques in the corresponding fields is therefore provided.

A few more applications have already been considered and partly implemented, but are not discussed in this thesis. They include interpolation, color-filter array interpolation, image sharpening, segmentation, and super-resolution imaging.

10.1 Video denoising

An adaptive spatio-temporal algorithm for video denoising is presented. The local polynomial approximation (*LPA*) is exploited in order to design $3D$ directional filtering kernels. For each specified direction in the $3D$ space-time domain, an adaptive scale (size of the kernel's support) is selected using the intersection of confidence intervals (*ICI*) rule. In this way a pointwise-adaptive spatio-temporal estimator is constructed. Experimental results show a good performance of the proposed method with a significant noise attenuation and nearly perfect edges and change-point preservation.

From the theoretical point of view, the following method is a particular case of the multi-dimensional *LPA-ICI* anisotropic estimator; as an algorithm, it is a rather direct extension to the 3D case of the anisotropic *LPA-ICI* image denoising of Section 8.1.

The simulations presented at the end of this section have been prepared, under the author's supervision, by Chiara Ercole and have been presented in [14][1]. The present section is substantially based on this publication.

[1] [14]: Ercole, C., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatio-temporal pointwise adaptive denoising of video: 3D non-parametric approach", *Proc. of the 1st International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM2005*, Scottsdale, AZ, January 2005.

10.1.1 Introduction

Noise is usually present in a video sequence because of transmission over noisy channels or acquisition with poor quality devices. Besides unpleasant visual effects, that sometimes can seriously compromise the perceiving and interpretation of the content, the main and most affecting problem is the degradation of the result of further processing such as video compression, segmentation, motion estimation. When addressing the problem of restoring a corrupted video sequence, the goal is to find a denoising scheme that can guarantee good performances of video processing algorithms and also a satisfactory visual quality. Since dealing with $3D$ data set, a good approach should take into account both spatial and temporal dimensions, so to exploit the spatial and temporal correlation in the video. Nevertheless, constraints of real-time implementation make all the efforts go in the direction of simple separable filters. Although they reach the required computational speed, these filters cannot suppress noise sufficiently well without introducing disturbing artifacts such as blurry edges or smoothing away salient characteristics like details and texture. The loss of these elements can heavily affect not only the further video processing, but also the subjective perception, since they encode a great amount of visual information contained in image sequences.

We develop the anisotropic *LPA-ICI* algorithm to $3D$, where time is the third dimension completing the $2D$ space of the image frame. In this way, we build an anisotropic $3D$ denoising filter for video.

10.1.2 Coordinate system

In order to better understand the notation used for the presented results, we spend a few paragraphs for the description of the coordinate system used for the $3D$ video filtering.

Let $x = (x_1, x_2, x_3)$, with x_1, x_2 being the spatial coordinates and the third coordinate x_3 being interpreted as time or frame number. A partition of the neighborhood in the $3D$ space can be done in different ways. Here, similarly as in Section (4.2.3), we discuss a partition based on spherical coordinate system. Figure 10.1 illustrates the meaning of the spherical angular coordinates θ and φ with respect to the cardinal spatial and temporal coordinates: θ is the angular coordinate of a polar system in the frame plane, while φ is the temporal angular coordinate. Thus, purely temporal directions are obtained for $\varphi = 0 \bmod \pi$; purely spatial directions are obtained for $\varphi = \pi/2 \bmod \pi$, $n \in \mathbb{Z}$. So, referring to the frame plane, direction along axis x_1 is obtained for $\varphi = \pi/2$ and $\theta = 0$, while direction parallel to axis x_2 is obtained for $\varphi = \pi/2$ and $\theta = \pi/2$.

10.1.3 Video-denoising simulation

As an illustrative application, we wish to recover the *Akiyo* video sequence y from its noisy observation $z = y+n$, where n is an additive white Gaussian noise with zero mean and σ=20. Here, contrary to the rest of the thesis, we assume that the range of the original data is $[0, 255]$. We implemented the proposed method in the simplest possible way, where the directional kernels are uniform over 1-pixel-width segments oriented along the twenty six directions originating from the center of a cube to the eight vertices, to the middle of the twelve sides,

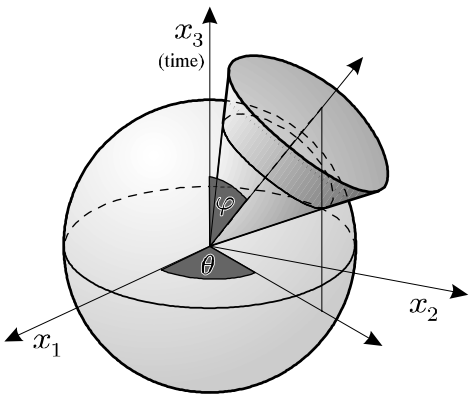


Figure 10.1: Spherical coordinates: the angle θ lies in the spatial frame plane while φ is the angle measured from the temporal axis.

to the center of the six faces. These kernels are *LPA* kernels of order zero, $m = [0, 0, 0]$. The following set of scales was used, $H = \{1, 2, 3, 5, 7, 10\}$. Exploiting the directional nature of the kernel supports, we improve the adaptive-scale selection (and thus the signal estimate) using a larger threshold Γ on the purely temporal directions, taking advantage of the high temporal correlation between frames.

When the data are discrete, it is impossible to have non-overlapping supports for the directional kernels, and, in practice, they are all overlapping in the origin voxel. Although in most of the applications presented in the previous chapters, formula (4.8) has been used also for origin-overlapping kernels, the larger number of directional estimates makes (4.8) unsuitable for this particular application. Thus, in this scenario, we use the uniform fusing formula (4.20), to avoid an excessive "super weighting" in the origin for g^+.

A performance comparison of this $3D$ algorithm over the two-dimensional version, working on single frames, has been done. Table 10.1 presents results for this comparison: for the $2D$ case, Γ=0.9 gives an average PSNR of 30.32dB, while for the $3D$ case, using Γ=0.7 for all directions but the temporal ones (Γ=1.2) an average PSNR of 33.86dB is reached. Not surprisingly, experimental results show that the $3D$ method outperforms the classical $2D$ version. What is somehow unexpected is that, despite the very simple structure of the used kernels, this basic implementation of the anisotropic $3D$ filtering yields a very good performance, with about 3.5dB of improvement over the $2D$ algorithm.

In the performed tests, a smaller value for Γ is considered, with respect to the usual $2D$ algorithm, since a larger number of directional estimators are taken into account in the fusing, in accordance with the analysis from Section (4.6). However, along the purely temporal directions, to take advantage of highly stationary areas in the frames, a bigger values for Γ can be chosen. This aspect is a peculiarity that can be exploited only if the video is sufficiently stationary.

Figure 10.2 shows (a) the original version of the 41st frame of the test sequence and (b) the corrupted one. Figure 10.3 shows the same frame, restored applying (a) the $2D$ version of the method, obtaining PSNR=30.32dB and (b) the proposed method ($3D$ algorithm), reaching a PSNR of 33.75dB. Visual in-

	average	min	min*	max
noisy	22.11	21.99	21.99	22.24
$2D$	30.32	30.02	30.02	30.64
$3D$	**33.86**	**32.97**	**33.45**	**34.38**

Table 10.1: PSNR (dB) values of the noisy (σ=20) and restored *Akiyo* sequence. Filtering is performed using the $2D$ and the $3D$ *LPA-ICI* estimator. Average, minimum and maximum PSNR values are calculated frame by frame on the whole (300 frames) sequence; min* is the minimum value of PSNR obtained on the *trimmed* sequence from frame 10 to 291.

(a) Original (b) Noisy

Figure 10.2: Frame 41 from the *Akiyo* sequence: (a) original and (b) noisy observation.

spection shows that edges and salient points of the video frame are preserved better in Figure 10.3(b) than in (a).

The *ICI* adaptive scales $h_i^+(x)$ represent the distribution of image features across the direction (θ_i, φ_i). Figure 10.4 shows these adaptive scales and the corresponding directional estimates for three different directions. In particular, Figure 10.4(a)-(b) are obtained for a purely-spatial direction (left, θ=0, φ=$\pi/2$); Figure 10.4(c)-(d) are obtained for a spatio-temporal direction (θ=$3\pi/4$, φ=$3\pi/4$); Figure 10.4(e)-(f) are obtained for the purely-temporal direction in the future (φ=π). It is remarkable how the temporal directions can give important information on the motion in the video sequence, selecting larger adaptive scales (white areas in the figure) for points that show slow motion or no motion at all from frame to frame, and smaller scales (dark areas) for points that move from frame to frame.

10.2 Shading from depth map: Z-buffer shading

Let $\nu = \left(-\frac{\partial f}{\partial x_1}, -\frac{\partial f}{\partial x_2}, 1\right)$ be the normal vector to a surface $f(x_1, x_2)$. If we assume the surface to be Lambertian then the reflected or transmitted luminous intensity l in any direction from an element is proportional to the cosine of the

(a) $2D$ (b) $3D$

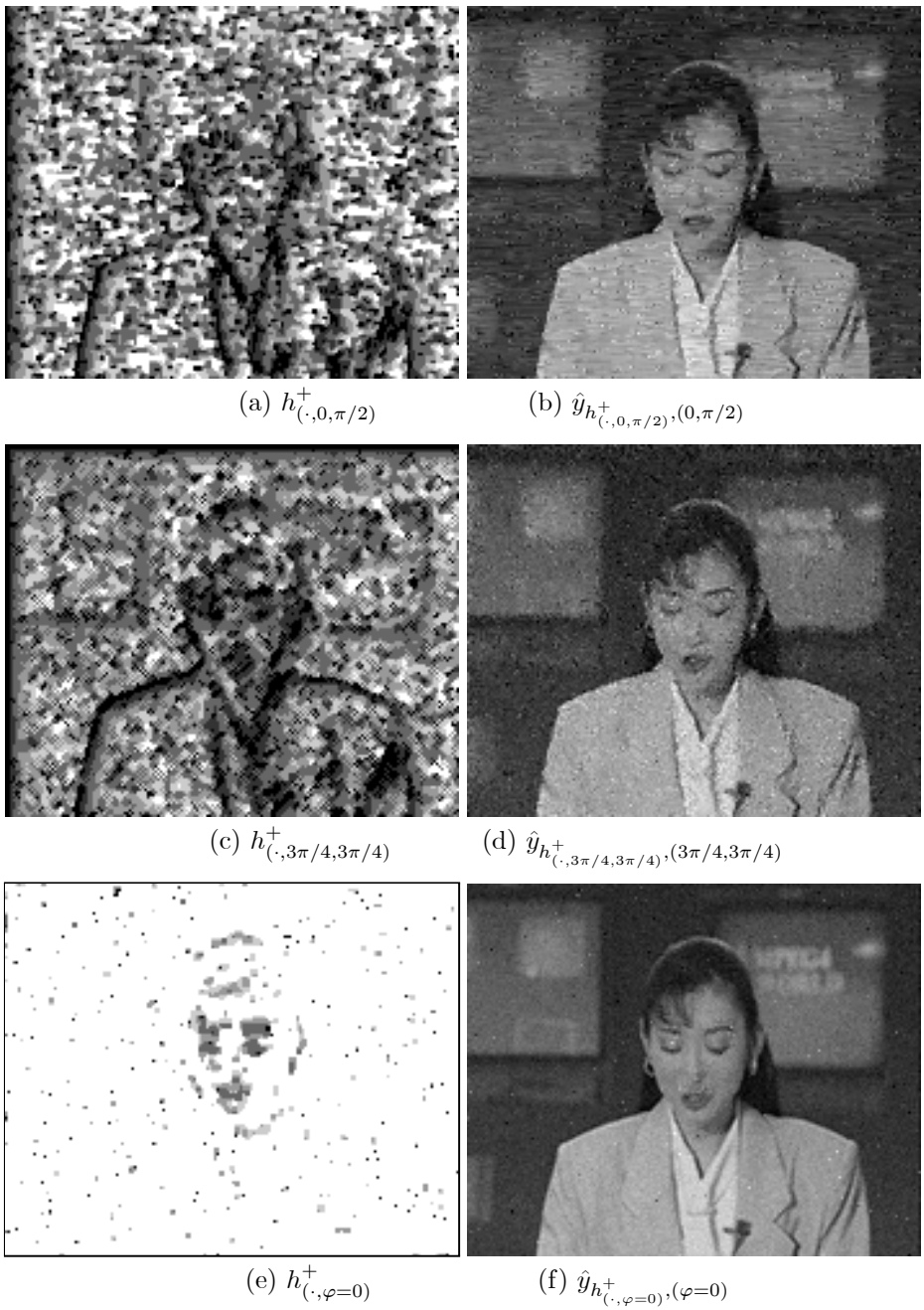
Figure 10.3: Frame 41 from the *Akiyo* sequence: (a) restored with $2D$ algorithm, (b) restored with the proposed $3D$ algorithm.

angle between ν and the direction of illumination $\upsilon = (\upsilon_1, \upsilon_2, \upsilon_z)$,

$$l \propto \frac{\nu \upsilon^T}{|\nu|\,|\upsilon|}.$$

As we discussed in Chapter 7, the estimation of the correct gradient (and then the correct normal vector) can be compromised by the failure of the differentiability assumption as well as by noise. When the normal vector is used for the realistic visualization (rendering) of a 3D surface, even a weak noise can produce a dramatic loss of quality. Indeed noise can be dealt with by means of larger derivative estimation kernels. However these larger kernels are unable to fit discontinuities in the underlying function.

Here, we present some shading examples obtained by estimating the normal vector with the use of the anisotropic gradient. A set of $K = 32$ directions has been used for this example. The advantage of the anisotropic gradient can be seen especially in Figure 10.8, where the estimator correctly recognizes the non-differentiability line along the boundary of the paws. Larger kernels are safely fitted in the anisotropic neighborhoods of smoothness, and the estimate is virtually noise-free even in the nearest vicinity of the non-differentiability line.



(a) $h^+_{(\cdot,0,\pi/2)}$ (b) $\hat{y}_{h^+_{(\cdot,0,\pi/2)},(0,\pi/2)}$

(c) $h^+_{(\cdot,3\pi/4,3\pi/4)}$ (d) $\hat{y}_{h^+_{(\cdot,3\pi/4,3\pi/4)},(3\pi/4,3\pi/4)}$

(e) $h^+_{(\cdot,\varphi=0)}$ (f) $\hat{y}_{h^+_{(\cdot,\varphi=0)},(\varphi=0)}$

Figure 10.4: Adaptive scales h_i^+ and directional estimates $\hat{y}_{h_i^+,(\theta_i,\varphi_i)}$ for the 41st frame from the *Akiyo* sequence: (a)-(b) purely spatial direction ($\theta = 0$, $\varphi = \pi/2$); (c)-(d) spatio-temporal direction ($\theta = 3\pi/4$, $\varphi = 3\pi/4$); (e)-(f) purely temporal direction in the future ($\varphi = \pi$). Darker colour is used in the left column to represent smaller scales. PSNR (dB) values for the directional estimates shown in the right column are, from top to bottom, 23.69, 23.91 and 29.56, respectively.

Figure 10.5: Rabbit: noisy depth map (Z-buffer).

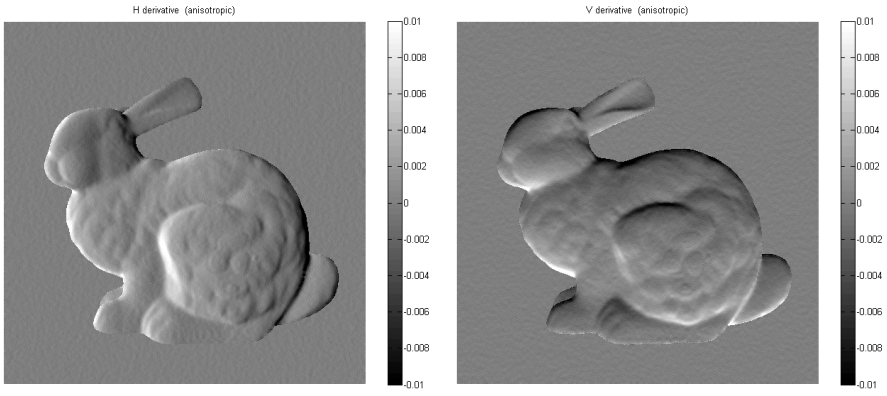


Figure 10.6: Anisotropic gradient: $\frac{\partial \varphi}{\partial x_1}$ (left) and $\frac{\partial \varphi}{\partial x_2}$ (right).

Figure 10.7: Shading obtained from the anisotropic gradient (normal vector).

Figure 10.8: Detail of the rabbit paws: from left to right, shading from gradient, kernel length 2, 3, 7, and from anisotropic gradient.

Chapter 11

Hybrid methods: *LPA-ICI* SA-DCT

The current research is focused on the combined use of the anisotropic *LPA-ICI* and some other filter. In [24][1] we present the following hybrid method.

LPA-ICI-driven Shape-Adaptive DCT

The two-dimensional separable DCT, computed on a square or rectangular support, is a well established and very efficient transform in order to achieve a sparse representation of image blocks. For natural images, its decorrelating performance is close to that of the optimum Karhunen-Loeve transform. Thus, the DCT has been successfully used as the key element in many compression and denoising applications. However, in the presence of edges such near-optimality fails. For this reason, other transforms with better edge adaptation capabilities (e.g. wavelets) have been used in denoising, and post-processing deringing filters are commonly used in MPEG-video decoders.

In the context of MPEG-4 coding, where arbitrarly shaped video-objects are introduced, a shape-adaptive DCT (SA-DCT) has been proposed [86, 87] as an extension of the classic separable DCT. The SA-DCT can be computed on a support of any shape, but retains a computational complexity comparable to that of the usual DCT. This makes the SA-DCT a well suited tool for the coding of image blocks that lie on the video-object's boundary.

We propose to use such a transform for image denoising.

The anisotropic *LPA-ICI* technique is used in order to define the shape of the transform's support in a pointwise-adaptive manner. It means that for each point in the image an adaptive estimation neighborhood is found. For each one of these neighborhoods a SA-DCT is performed. The thresholded SA-DCT coefficients are used to reconstruct a local estimate of the signal within the adaptive-shape region. Since regions corresponding to different points are in general overlapping (and generate an overcomplete representation of the signal), the local estimates are averaged together ("fused") using adaptive weights that depend on the region's statistics.

[1] [24]: Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool", (accepted) *SMMSP 2005*, Riga, June 2005.

Figure 11.1: The *Cameraman* image restored by the anisotropic *LPA-ICI* SA-DCT estimator. The improvement in performance is noticeable, by visual inspection, and by comparison of the objective criteria: $ISNR$=8.89dB, SNR=23.28dB, $PSNR$=28.87dB, $RMSE$=9.19, MAE=5.65, MAX=87.36.

We conclude the thesis with a result that shows the potential of this hybrid method. Figure 11.1 shows the *Cameraman* image restored by the "anisotropic *LPA-ICI* + SA-DCT" estimator. The observation was the noisy image shown in Figure 8.1 on page 110. Not only objective criteria are better, but also the visual appearence of the estimate is clearly superior: edges are clean, and no unpleasant ringing artifacts are introduced by the fitted transform.

Bibliography

[1] Analoui, M., and J.P. Allebach, "Model-based halftoning by direct binary search", *Proc. SPIE/IS&T Symposium on Electronic Imaging Science and Technology*, vol. 1666, pp. 96-108, 1992.

[2] Astola, J., and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, New York, CRC Press, 1997.

[3] Bertero, M., and P. Boccacci, *Introduction to inverse problems in imaging.* Inst. of Physics Publishing, 1998.

[4] Bozkurt Unal, G., and A.E. Çetin, "Restoration of Error-Diffused images using Projection onto Convex Sets", *IEEE Trans. Image Processing*, December 2001.

[5] Carr, J.C., A.H. Gee, R.W. Prager, and K. J. Dalton, "Quantitative visualisation of surfaces from volumetric data", *Proceedings of WSGC'98 - The sixth international conference in central europe on Computer Graphics and Visualization*, Plzen, Czech Republic, February 10-14, 1998.

[6] Chang, P.C., C.S. Yu, and T.H. Lee,"Hybrid LMS-MMSE Inverse Halftoning Technique", *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 95-103, Jan. 2001.

[7] Cleveland, W.S., and C. Loader, "Smoothing by local regression: principles and methods", *Statistical theory and computational aspects of smoothing*, Springer, New York, pp. 10-49, 1996.

[8] Coifman, R.R., and D. Donoho, "Translation-invariant de-noising", in *Wavelets and Statistics* (editors. A. Antoniadis and G. Oppenheim), *Lecture Notes in Statistics*, Springer-Verlag, pp.125-150, 1995.

[9] Coifman, R.R., and D. Donoho, "Translation invariant de-noising", Technical Report 475, Dept. of Statistics, Stanford University, May 1995.

[10] Dias, J.M.B., "Fast GEM wavelet-based image deconvolution algorithm", *Proc. Int. Conf. on Image Proc. ICIP 2003*, vol. 2, 2003.

[11] Donoho, D.L., and I.M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, n. 81, pp. 425-455, 1994.

[12] Emery, M., A. Nemirovski, and D. Voiculescu, *Lectures on Probability Theory and Statistics - Ecole d'Eté de Probabilités de Saint-Flour XXVIII - 1998*, ed. P. Bernard, Springer-Verlag, LNM vol. 1738, 2000.

[13] Engl, H.W., M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, 1996.

[14] Ercole, C., A. Foi, V. Katkovnik, and K. Egiazarian, "Spatio-temporal pointwise adaptive denoising of video: 3D non-parametric approach", *Proc. of the 1st International Workshop on Video Processing and Quality Metrics for Consumer Electronics, VPQM2005*, Scottsdale, AZ, January 2005.

[15] Fan, J., and I. Gijbels, *Local polynomial modelling and its application*, Chapman and Hall, London, 1996.

[16] Farnebäck, G., "A Unified Framework for Bases, Frames, Subspace Bases, and Subspace Frames" *Proc. of the 11th Scandinavian Conf. on Image Analysis,* Kangerlussuaq, Greenland, 1999.

[17] Fienup, J., D. Griffith, L. Harrington, A.M. Kowalczyk, J.J. Miller, and J.A. Mooney, "Comparison of reconstruction algorithms for images from sparse-aperature systems", *Proc. SPIE* 4792-01, July 2002.

[18] Figueiredo, M.A.T., and R. D. Nowak, "An EM algorithm for wavelet-based image restoration", *IEEE Trans Image Proc.*, vol. 12, N 8, pp. 906-916, 2003.

[19] Floyd, R.W., and L. Steinberg, "An Adaptive Algorithm for Spatial Greyscale", *Proc. Society for Information Display*, vol. 17, no. 2, pp. 75-77, 1976.

[20] Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "A novel anisotropic local polynomial estimator based on directional multiscale optimizations", *Proc. 6th IMA Int. Conf. Math. in Signal Processing*, Cirencester (UK), pp. 79-82, 2004.

[21] Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "Inverse halftoning based on the anisotropic LPA-ICI deconvolution", *Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Proc., SMMSP 2004,* Vienna, pp. 49-56, September 2004.

[22] Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Anisotropic local approximations for pointwise adaptive signal-dependent noise removal", (accepted) *XIII European Signal Proc. Conf., EUSIPCO 2005*, September 2005.

[23] Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", (accepted) *IEEE 2005 Int. Conf. Image Processing, ICIP 2005,* September 2005.

[24] Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool", (accepted) *SMMSP 2005*, Riga, June 2005.

[25] Freeman, W.T., and E.H. Adelson, "The design and use of steerable filters", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991.

[26] Geusebroek, J.M., A.W.M. Smeulders, and J. van de Weijer, "Fast anisotropic Gauss filtering", *IEEE Trans. Image Processing*, 2003.

[27] Goldenshluger, A., "On pointwise adaptive nonparametric deconvolution", *Bernoulli,* vol. 5, pp. 907-925, 1999.

[28] Goldenshluger, A., and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression", *Math. Meth. Statistics*, vol. 6, pp. 135-170, 1997.

[29] Gordon, D., "Image scape shading of 3-dimensional objects", *Computer Vision, Graphics, and Image Processing,* vol. 29, pp. 361-376, 1985.

[30] Hampel, F.R., "The influence curve and its role in robust estimation", *Journal of American Statistical Association*, 62, pp. 1179-1186, 1974.

[31] Hart, J.D., *Nonparametric smoothing and lack-of-fit tests*, Springer-Verlag, New York, 1997.

[32] Hastie, T.J., and R.J. Tibshirani, *Generalized linear models*, Chapman and Hall, London, 1990.

[33] Heyde, C.C., *Quasi-likelihood and its applications,* Springer-Verlag, New York, 1997.

[34] Hein, S., and A. Zakhor, "Halftone to continuous-tone conversion of error-diffusion coded images", *IEEE Trans. on Image Processing*, vol. 4, no. 2, pp. 208-216, February 1995.

[35] Hernández, E., and G. Weiss, *A first course on wavelets*, CRC Press, 1996.

[36] Huber, P.J., *Robust statistics*, John Wiley & Sons Inc., 1981.

[37] Jain, A.K., *Fundamentals of digital image processing*, Prentice-Hall, 1989.

[38] Jarvis, J.F., C.N. Judice, and W.H. Ninke, "A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays", *Computer Graphics Image Processing,* 5, pp. 13-40, 1976.

[39] Jiang, S.S., and A.A. Sawchuk, "Noise updating repeated Wiener filter and other adaptive noise smoothing filters using local image statistics", *Appl. Opt.*, vol. 25, pp. 2326-2337, 1986.

[40] Katkovnik, V., "Problem of approximating functions of many variables". *Autom. Remote Control*, vol. 32, no. 2, part 2, pp. 336-341, 1971.

[41] Katkovnik, V., *Nonparametric identification and smoothing of data (Local approximation methods)* (in russian), Nauka, Moscow, 1985.

[42] Katkovnik, V., "A new method for varying adaptive bandwidth selection", *IEEE Trans. on Signal Proc.,* vol. 47, no. 9, pp. 2567-2571, 1999.

[43] Katkovnik, V., K. Egiazarian, and J. Astola, "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule", *J. of Math. Imaging and Vision*, vol. 16, no. 3, pp. 223-235, 2002.

[44] Katkovnik, V., K. Egiazarian, and J. Astola. *Adaptive varying scale methods in image processing.* Tampere International Center for Signal Processing, TICSP Series, no. 19, Tampere, TTY, Monistamo, 2003.

[45] Katkovnik, V., K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring", *IEEE Trans. on Image Processing*, in print, 2004.

[46] Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Directional varying scale approximations for anisotropic signal processing", *Proc. XII European Signal Proc. Conf., EUSIPCO 2004,* Vienna, pp. 101-104, September 2004.

[47] Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Anisotropic local likelihood approximations", *Proc. of Electronic Imaging 2005*, 5672-19, January 2005.

[48] Katkovnik, V., and I. Shmulevich, "Kernel density estimation with adaptive varying window size", *Pattern Recognition Letters,* no. 23, pp. 1641-1648, 2002.

[49] Katznelson, Y., *An introduction to harmonic analysis*, second edition. Dover Publications, New York, 1976.

[50] Kelly, S., M. Kon, and L. Raphael, "Pointwise convergence of wavelet expansions", *Bull. Amer. Math. Soc.,* vol. 30, pp. 87-94, 1994.

[51] Kerkyacharian, G., O. Lepski, and D. Picard, "Nonlinear estimation in anisotropic multi-index denoising", *Prob. Theory and Related Fields*, vol. 121, no. 2, pp. 137-170, 2001.

[52] Kim, H.Y. and R. de Queiroz, "Inverse halftoning by decision tree learning", *Proc. IEEE Intl. Conf. on Image Processing, ICIP*, Barcelona, Spain, 2003.

[53] Kingsbury, N., "Complex Wavelets for Shift Invariant Analysis and Filtering of Signals", *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234-253, May 2001.

[54] Kite, T.D., B.L. Evans, T.L. Sculley, and A.C. Bovik, "Digital Image Halftoning as 2-D Delta-Sigma Modulation", *Proc. IEEE Int. Conf. on Image Processing*, 1997, Santa Barbara, CA, vol. I, pp. 799-802, 1997.

[55] Kite, T.D., B.L. Evans, A.C. Bovik and T.L. Sculley, "Modeling and quality assessment of halftoning by error diffusion", *IEEE Trans. Image Proc.*, vol. 9, pp. 909-922, May 2000.

[56] Kite, T.D., N. Damera-Venkata, B.L. Evans, and A.C. Bovik, "A Fast, High-Quality Inverse Halftoning Algorithm for Error Diffused Halftones", *IEEE Trans. on Image Processing*, vol. 9, no. 9, pp. 1583-1592, Sep. 2000.

[57] Kite, T.D., B.L. Evans, and A.C. Bovik, "Fast rehalftoning and interpolated halftoning algorithms with flat low-frequency response", *Proceedings of International Conference on Image Processing, 1999. ICIP 99.* vol. 3, Oct. 1999.

[58] Kolmogorov, A.N., and S. Fomin, *Introductory Real Analysis*, (edited by R.A. Silverman), Dover Publications, New York, 1975.

[59] Kondo, K., Y. Ichioka, and T. Suzuki, "Image restoration by Wiener filtering in presence of signal-dependent noise", *Applied Optics*, vol. 16, no. 9, 1977.

[60] Kuan, D.T., A.A. Sawchuk, T.C. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal dependent noise," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 7, pp. 165-177, 1985.

[61] Lee, J.S., "Digital image enhancement and noise filtering by using local statistics", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, 1980.

[62] Lee, J.S., "Refined filtering of image noise using local statistics", *Comput. Graph. Image Proc.* vol. 15, pp. 380-389, 1981.

[63] Li, S. and W.K. Liu, "Moving Least Square Reproducing Kernel Method Part II: Fourier analysis", *Computer Meth. in Appl. Mechanics and Engineering,* vol. 139, pp. 159-194, 1996.

[64] Liu, W.K., S. Li and T. Belytschko, "Moving Least Square Reproducing Kernel Method Part I: Methodology and convergence", *Computer Meth. in Appl. Mechanics and Engineering,* vol. 143, pp. 113-154, 1996.

[65] Loader, C., *Local regression and likelihood*, Series Statistics and Computing, Springer-Verlag, New York, 1999.

[66] Mallat, S., *A wavelet tour of signal processing*, 2nd edition, Academic Press, 1998.

[67] Lepski, O., E. Mammen and V. Spokoiny, "Ideal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selection". *Annals of Statistics*, vol. 25, no. 3, 929–947, 1997.

[68] Lu, H., Y. Kim, and J.M.M. Anderson, "Improved Poisson intensity estimation: denoisng application using poisson data," *IEEE Trans. Image Processing,* vol. 13, no. 8, pp. 1128–1135, 2004.

[69] McCullagh, P., and J.A. Nelder, *Generalized linear models*, (2nd ed.), London, Chapman and Hall, 1999.

[70] Mertins, A., *Signal Analysis: Wavelets, filter banks, time-frequency transforms and applications*, John Wiley and sons, 1999.

[71] Meşe, M., and P.P. Vaidyanathan, "Look up table (LUT) Method for inverse halftoning", *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 1566-1578, 2000

[72] Meşe, M., and P.P. Vaidyanathan, "Tree-Structured Method for LUT Inverse Halftoning and for Image Halftoning", *IEEE Trans. on Image Processing*, vol. 11, no. 6, pp. 644-655, June 2002.

[73] Neelamani, R., R. Nowak, and R. Baraniuk, "Model-based Inverse Halftoning with Wavelet Vaguelette Deconvolution", *Proc. IEEE Int. Conf. Image Processing, ICIP '00*, vol. 3, pp. 973-976, Vancouver, Canada, September 2000.

[74] Neelamani, R., R. Nowak and R. Baraniuk, "WInHD: Wavelet-based Inverse Halftoning via Deconvolution", *IEEE Trans. on Image Processing*, (submitted), October 2002.

[75] Neelamani, R., H. Choi, and R. Baraniuk, "Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems", *IEEE Trans. on Image Proc.,* vol. 52, no. 2, 2004.

[76] Nowak, R.D., and R. Baraniuk, "Wavelet-domain filtering for photon imaging systems", *IEEE Trans. Image Processing,* vol. 8, no. 5, pp 666-678, 1999.

[77] Perona, P., and J. Malik, "Scale-space and edge detection using anisotropic diffusion", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

[78] Polzehl, J., and V. Spokoiny, "Image denoising: pointwise adaptive approach", *Annals of Statistics*, vol. 31, no. 1, 2003.

[79] Qiu, P., "Discontinuous regression surfaces fitting", *Annals of Statistics*, vol. 26, no. 6, pp. 2218-2245, 1998.

[80] Qiu, P., "The local piecewise linear kernel smoothing procedure for fitting jump regression surfaces", *Technometrics*, vol. 46, no. 1, pp. 87-98, 2004.

[81] Rangarayyan, R.M., M. Ciuc, and F. Faghih, "Adaptive-neighborhood filtering of images corrupted by signal-dependent noise", *Appl. Opt.*, vol. 37, pp.4477-4487, 1998.

[82] Rooms, F., W. Philips, and P. Van Oostveldt, "Integrated approach for estimation and restoration of photon-limited images based on steerable pyramids", *Proc. of EC-VIP-MC 2003*, pp. 131-136, 2003.

[83] Rudin, W., *Real and complex analysis*, third edition, McGraw-Hill, New York, 1987.

[84] Seber, G.A., and C.J. Wild, *Nonlinear regression,* Wiley, New York, 1989.

[85] Shin, B.S and G.S. Shin, "Fast Normal Estimation Using Surface Characteristics", *Proceedings of the 6th IEEE Visualization Conference (Visualization '95),* 1995.

[86] Sikora, T., and B. Makai, "Shape-adaptive DCT for generic coding of video", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59-62, 1995.

[87] Sikora, T., "Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments", *Signal Processing: Image Communication*, vol. 7, pp. 381-395, 1995.

[88] Simoncelli, E.P., and H. Farid, "Steerable wedge filters for local orientation analysis", *IEEE Trans. on Image Proc.*, vol. 5, no. 9, pp. 1377-1382, 1996.

[89] Stanković, LJ., "Performance Analysis of the Adaptive Algorithm for Bias-to-Variance Tradeoff", *IEEE Trans. on Signal Proc.*, vol. 52, no. 5, pp. 1228-1234, 2004.

[90] Starck, J., E.J. Candes, and D.L. Donoho, "The curvelet transform for image denoising", *IEEE Trans. on Image Proc.*, vol. 11, no. 6, pp. 670-684, 2002.

[91] Timmermann, K.E., and R. Nowak, "Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging" *IEEE Trans. Information Theory,* vol. 45, no. 3, pp. 846-862, 1999.

[92] Ulichney, R., *Digital Halftoning*, The MIT Press, Cambridge, Mass., 1987.

[93] Wach, H.B., and E.R. Dowski Jr., "Noise modeling for design and simulation of color imaging systems", *Proc. of IS&T/SID's12th Color Imaging Conference, CIC 12,* 2004.

[94] Wedderburn, R.W.M., "Quasilikelihood functions, generalized linear models and the Gauss-Newton method", *Biometrica*, vol. 61, pp. 439-447, 1974.

[95] White, R.L., "Image restoration using the damped Richardson-Lucy method", *Astr. Data An. Software and Systems, ASP Conf. Series*, vol. 61, pp. 292-295, 1994.

[96] Willett, R.M., and R.D. Nowak, "Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging", *IEEE Trans. Medical Imaging,* vol. 22, no. 3, pp. 332-350, 2003.

[97] Wong, P.W., "Inverse halftoning and kernel estimation for error diffusion", *IEEE Trans. on Image Processing*, vol. 4, no. 4, pp. 486-498, 1995.

[98] Wong, P.W., and N.D. Memon, "Image Processing for Halftones", *IEEE Signal Processing Magazine*, vol. 20, no. 4, July 2003.

[99] Xiong, Z., M.T. Orchard, and K. Ramchandran, "Inverse halftoning using wavelets", *IEEE Trans. Signal Processing*, vol. 8, pp. 1479-1482, Oct. 1999.

[100] Yagel, R., D. Cohen, and A. Kaufman, "Normal Estimation in 3D Discrete Space", *The Visual Computer*, vol. 8, no. 5-6, June 1992, pp. 278-291, 1992.

[101] Yin, L., R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted Median Filters: A Tutorial", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Proc.,* vol. 43, no. 3, pp. 157-192, 1996.

[102] Yu, W., K. Daniilidis, and G. Sommer, "Approximate Orientation Steerability Based on Angular Gaussians", *IEEE Trans. on Image Proc.*, vol. 10, no. 2, pp. 193-205, 2001.

[103] Zayed, A., "Pointwise convergence of a class of non-orthogonal wavelet expansion", *Proc. Amer. Math. Soc.*, vol. 128, pp. 3629-3637, 2000.