Novel Spatially Adaptive Anisotropic Local Approximation Techniques in Image Processing

Vladimir Katkovnik, Karen Egiazarian, Jaakko Astola

Institute of Signal Processing
Tampere University of Technology
Finland

EI-2006 San Jose Short Course,
January 17 2006

Course description

An adaptive LPA is developed to deal with anisotropic signals contaminated by a random noise.

The adaptation is based on recent new statistical methods and proved theoretically as well as practically to be very efficient. The algorithm searches for a largest local star-shaped neighborhood where LPA fits well to data.

Overall the developed technique defines a wide class of nonlinear spatially adaptive filters demonstrating the state-of-art performance.

The material of the course is based on the forthcoming book:

V. Katkovnik, K. Egiazarian, J. Astola, ”*Local Approximation Techniques in Signal and Image Processing," SPIE Press, 2006.*

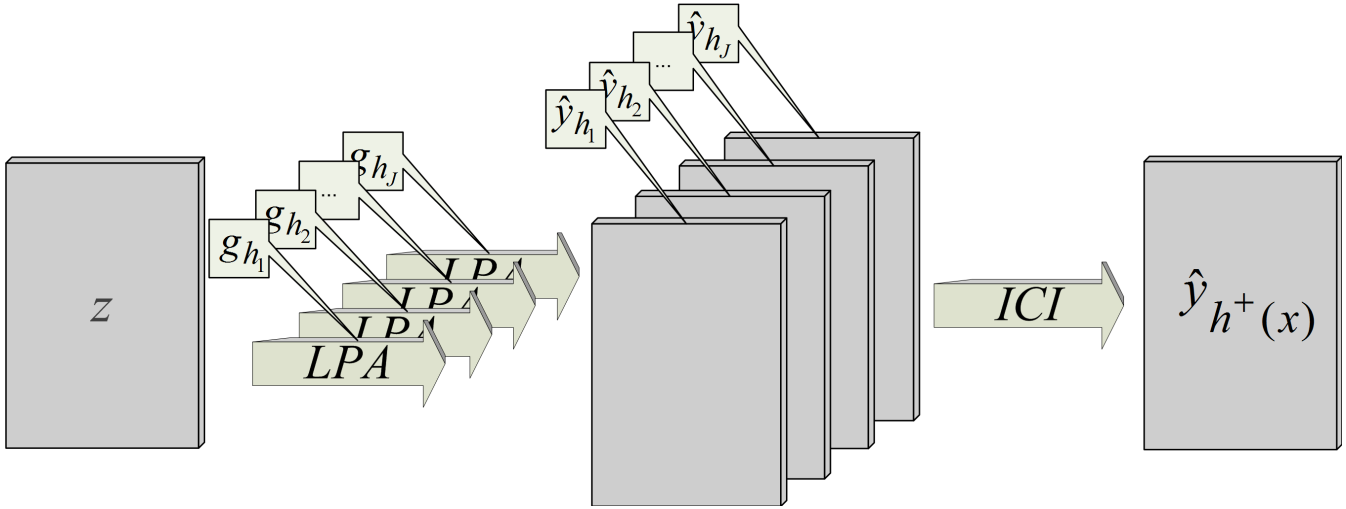
Three key words associated with image analysis operators arise, to which we will give a more and more precise meaning:

- Locality;
- Anisotropy;
- Adaptivity.

The locality means that there is a neighborhood small enough where the image intensity is slowly varying and well approximated by some continuous basis functions.

In the same time the image intensity is anisotropic demonstrating different behavior in different directions. It follows that a good local approximation can be achieved only in a non – symmetric neighborhood.

Starting from the local anisotropic approximation we introduce a multiscale multidirectional filter bank which works as a useful front-end for representing the structures of visual images. Equipped with special adaptation rules it enables highly efficient nonlinear filtering.



This adaptive system allows different interpretations:

- Nonstationary filter;
- Switch filter;
- Match filter.

Developed algorithms are implemented in *Matlab* codes and based on efficient calculations. The codes are available on http：//www.cs.tut.fi/~lasip/.

Topics

1. Basic Ideas, Approach, Theory:

   LPA and filter design, directional LPA, LPA accuracy, adaptive scale selection.

2. Applications:

   image denoising, adaptive scale differentiation, image deblurring, 3D inverse imaging, multiframe blind deblurring.

3. Non-gaussian Observations:

   local maximum likelihood and quasi-likelihood, photon-limited (poissonian) Imaging, inverse poissonian imaging.

4. Summary.

Basic *LPA*

We wish to mention early works in statistics using local polynomials by an Italian meteorologist Schiaparelli (1866) and a Danish actuary Gram (1879) (famous for developing the Gram-Schmidt orthogonalization procedure).

In sixties/seventies of the twentieth century the idea became a subject of an intensive theoretical study and applications: in statistics due Nadaraya (1964), Watson (1964), Parzen (1962), Stone (1977) and in engineering sciences due Brown (1963), Savitzky and Golay (1964), Petersen (1969), Katkovnik (1971, 1976), Cleveland (1979).

It is appeared under different names: moving (sliding, windowed) least square, Savitzky – Golay filter, reproducing kernels, moment filters, etc.

Local approximation

Let y be a scalar function of a scalar argument x. We wish to reconstruct this function using observations

$$z_s = y(X_s) + \varepsilon_s,\ s = 1, \ldots, n,$$

where coordinates X_s are known and ε_s are zero mean random errors.

In a neighborhood of x the Taylor series gives for $y(X_s)$:

$$y(X_s) \simeq y(x) - y^{(1)}(x)(x - X_s) + y^{(2)}(x)(x - X_s)^2/2 + \ldots.$$

Since the function y and the derivatives $y^{(1)}$ and $y^{(2)}$ are unknown we look for fitting data $y(X_s)$ in the form

$$y(X_s) \simeq C_0 - C_1(x - X_s) + C_2(x - X_s)^2/2 + \ldots,$$

where the coefficients C_0, C_1 and C_2 are used as estimates for $y(x)$, $y^{(1)}(x)$ and $y^{(2)}(x)$.

How to formalize a local fit?

The weighted least square method gives the following criterion

$$J(x, C) = \sum_s w(x - X_s) e_s^2,$$

$$e_s = z_s - y(x, X_s),$$

$$y(x, X_s) = C_0 - C_1(x - X_s) + C_2(x - X_s)^2/2,$$

where w is a window function.

Minimizing J on C:

$$\hat{C}(x) = \arg\min_C J(x, C).$$

The notation $\hat{C}(x)$ emphasizes that the estimate of C depends on x.

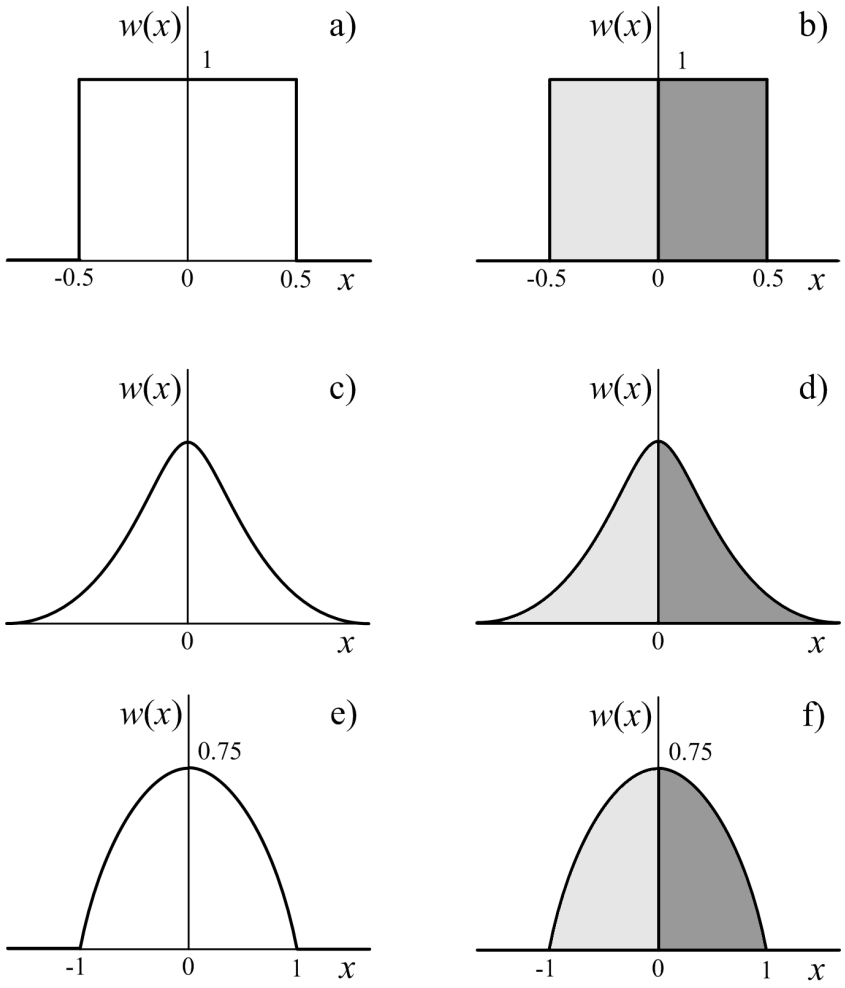
The model in J can be zero, first, second or higher orders depending on the power of used polynomials.

The parameter $\hat{C}(x)$ immediately gives the estimates of the function y and the derivatives $y^{(r)}$:

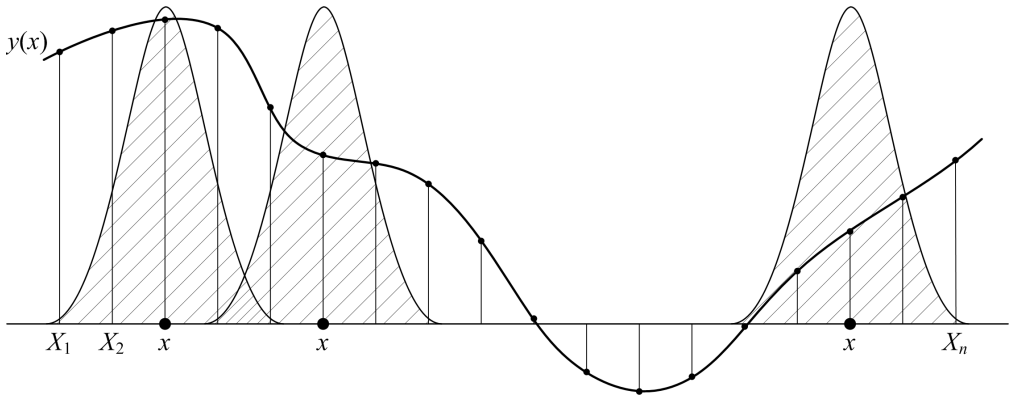
$$\hat{y}(x) = \hat{C}_0(x),$$
$$\hat{y}^{(1)}(x) = \hat{C}_1(x),$$
$$\hat{y}^{(2)}(x) = \hat{C}_2(x).$$

The conventional windows can be used: Kaiser, Hamming, Bartlett, Blackman, Chebyshev, etc.

The examples of symmetric and nonsymmetric left/right windows.



Nonparametric estimation



The sliding window estimates belong to the class of the nonparametric estimates as we use the model

$$\hat{y}(x, X_s) = \hat{C}_0(x) - \hat{C}_1(x)(x - X_s) + \hat{C}_2(x)(x - X_s)^2/2$$

only for $X_s = x$ only and it gives

$$\hat{y}(x) = \hat{y}(x, X_s)|_{x=X_s} = \hat{C}_0(x).$$

For the estimation of the derivatives we have

$$\hat{y}^{(1)}(x) = \partial_{X_s}\hat{y}(x, X_s)|_{x=X_s} = \hat{C}_1(x)$$

and

$$\hat{y}^{(2)}(x) = \partial^2_{X_s}\hat{y}(x, X_s)|_{x=X_s} = \hat{C}_2(x).$$

The coefficient $\hat{C}_j$ are calculated for a fixed x and the model is used for this x only.

In the nonparametric case everything is completely different, the dependence on x is defined by the varying coefficients $\hat{C}_0(x)$, $\hat{C}_1(x)$, $\hat{C}_2(x)$ while the basis functions are used with a fixed zero value of the argument.

The localization relaxes the sensitivity of the standard parametric models to the order of the model and transforms the polynomial fit in a much more flexible tool than it can be expected from the standard parametric approach.

Examples

Zero order 1D estimator (demo_ZeroOrderEstimator1D.m)

Let us assume that the zero order LPA is used, $m = 0$:

$$J(x, C_0) = \sum_s w(x - X_s) e_s^2,\ e_s = z_s - y(x, X_s),\ y(x, X_s) = C_0,$$

and minimization of $J(x, C_0)$ on C_0 gives

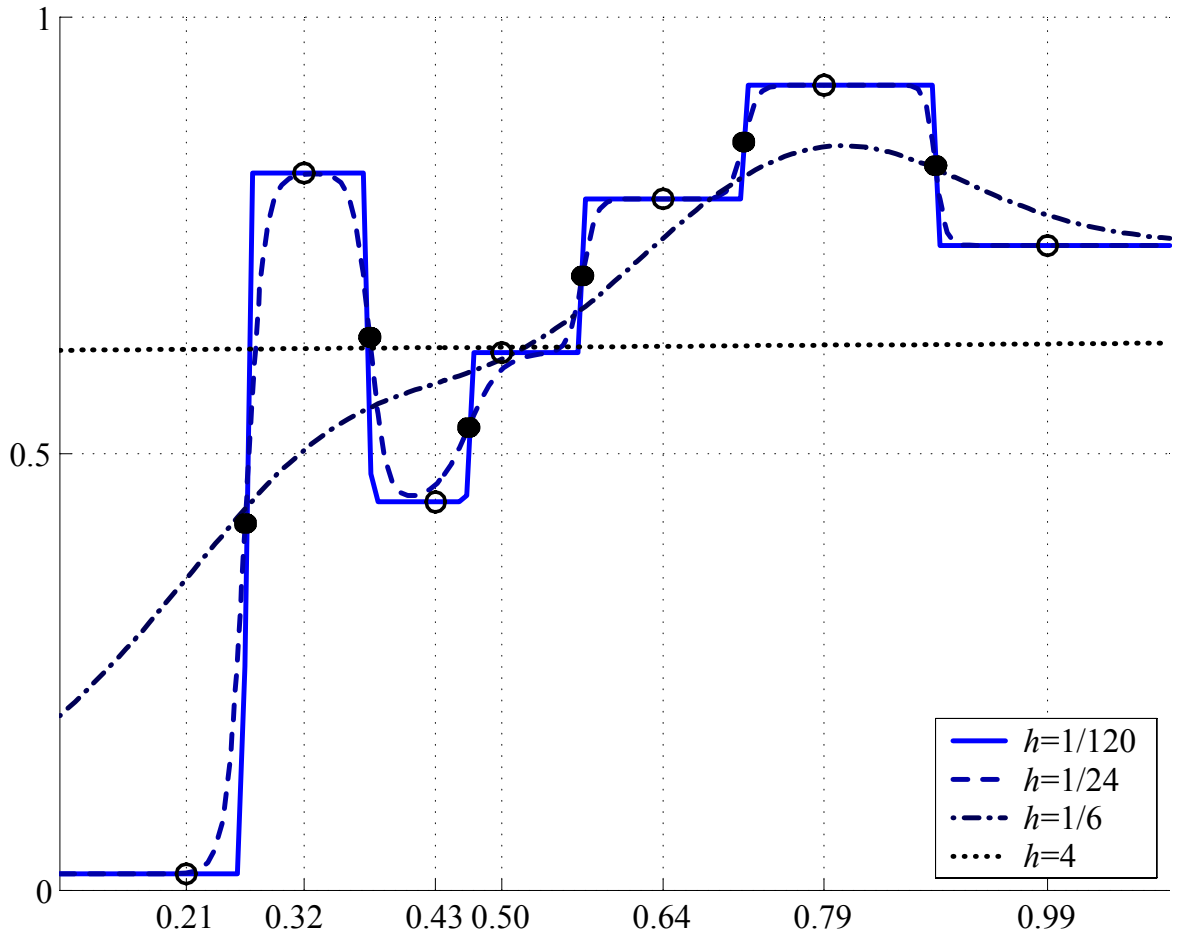
$$\partial_{C_0} J(x, C_0) = 0 \Rightarrow \sum_s w(x - X_s) C_0 = \sum_s w(x - X_s) z_s.$$

The LPA model has a form of the moving weighted average $Nadaray - Watson$ estimator:

$$\hat{C}_0 \triangleq \hat{y}_h(x) = \sum_s g_h(x, X_s) y_s,$$

$$g_h(x, X_s) = \frac{w_h(x - X_s)}{\sum_s w_h(x - X_s)},$$

$$w_h(x) = \frac{1}{h} w(\frac{x}{h}).$$



The LPA of the degree $m = 0$ with the $Gaussian$ window.

First order 1D estimator (demo_FirstOrderEstimator1D.m)

Let us go to the first order LPA, $m = 1$. It means that the LPA model includes the polynomials of the zero and first degrees and two coefficients C_0 and C_1 should be found.

The solution for $\hat{C} = (C_0, C_1)^T$ can be presented in the form

$$\hat{y}_h(x) = \hat{C}_0(x,h) = \sum_s g_h(x,X_s) z_s,$$

$$g_h(x,X_s) = \frac{w_h(x-X_s)[S_2(x) + S_1(x)(x-X_s)]}{S_2(x)S_0(x) - S_1^2(x)}$$

for the signal estimate.

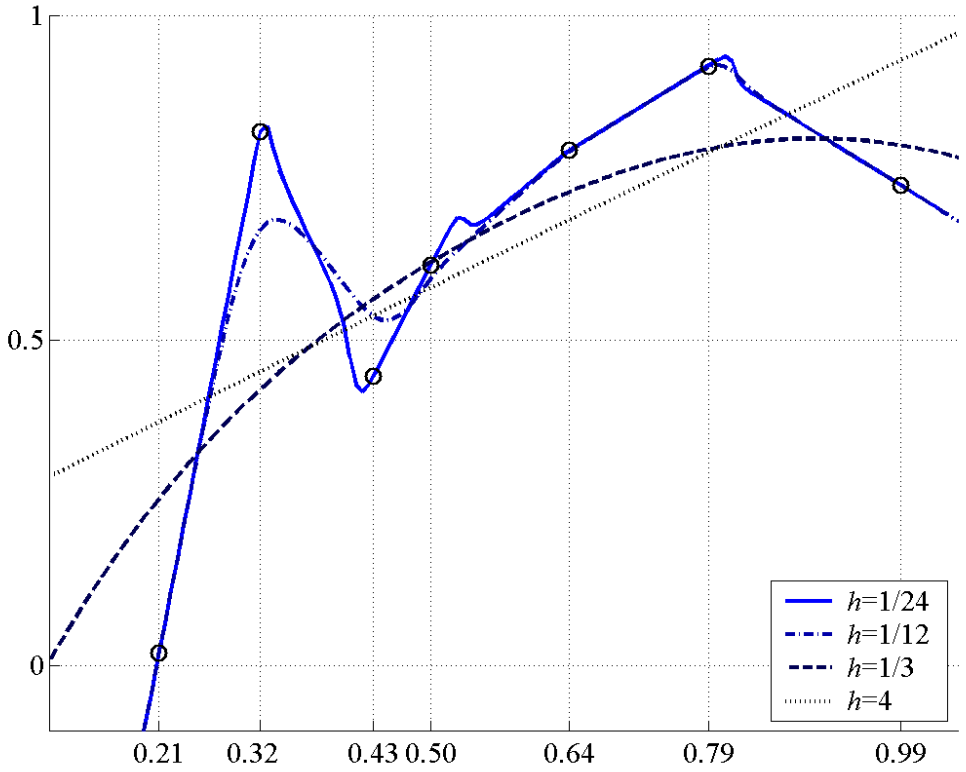
For the derivative estimate

$$\hat{y}_h^{(1)}(x) = \hat{C}_1(x,h) = \sum_s g_h^{(1)}(x,X_s) z_s,$$

$$g_h^{(1)}(x,X_s) = -\frac{w_h(x-X_s)[S_1(x) + S_0(x)(x-X_s)]}{S_2(x)S_0(x) - S_1^2(x)},$$

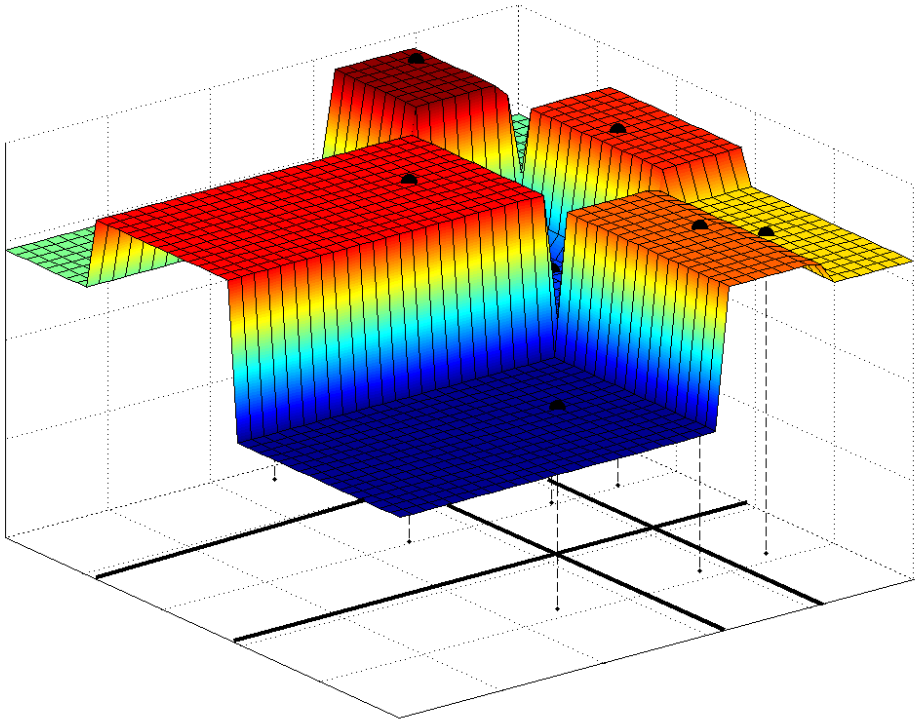
$$S_l(x) = (-1)^{-l} \sum_s w_h(x-X_s)(x-X_s)^l,$$

$$l = 0,1,2.$$



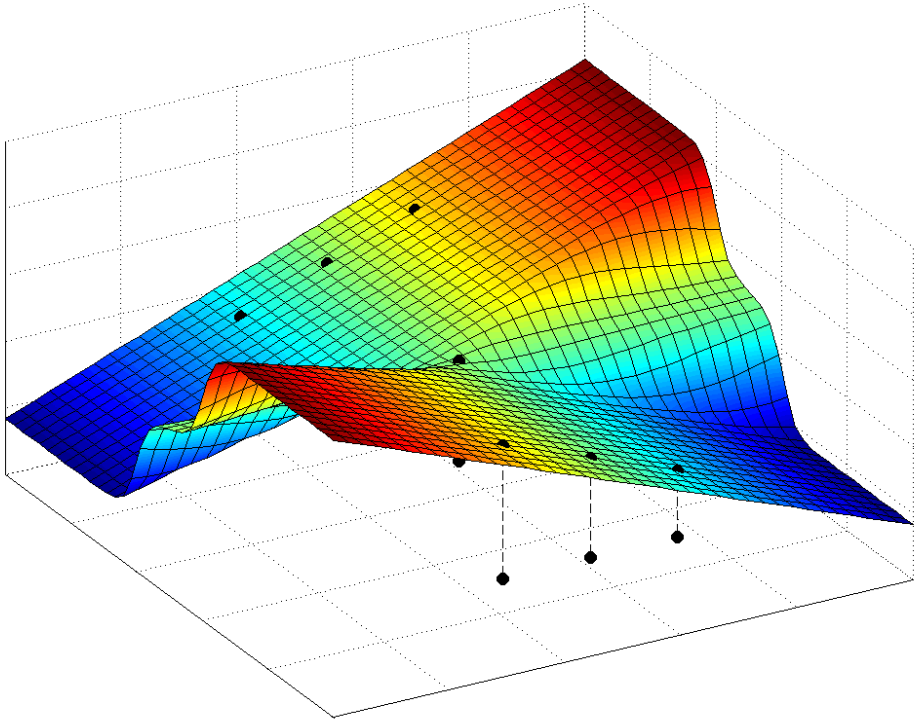
The LPA of the degree $m = 1$ with the $Gaussian$ window.

Zero order 2D estimation (demo_ZeroOrderEstimator2D.m)



The LPA of the degree $m = 0$ with the $Gaussian$ window.

First order 2D estimator (demo_FirstOrderEstimator2D.m)



The LPA of the degree $m = 1$ with the $Gaussian$ window.

LPA Filter Design

Observation model

We are given observations of y in the form $y_s = y(X_s)$,

$$X_s = (x_1(s),\ x_2(s)),\ s = 1,\ldots,n.$$

Noisy observations:

$$z_s = y(X_s) + \varepsilon_s,\ s = 1,\ldots,n.$$

It can be denoted also that

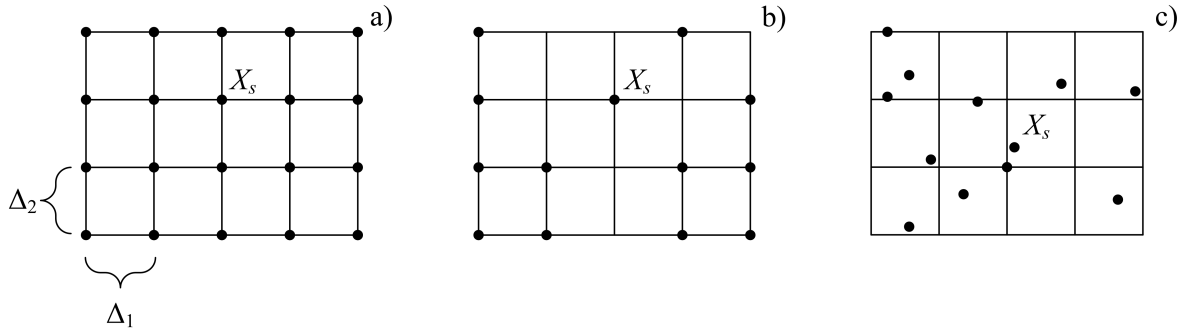
$$X = \{X_s :\ s = 1,\ldots,n\}.$$

For a regular grid the coordinates of the pixels can be given explicitly as

$$X = \{l_1\Delta_1,\ l_2\Delta_2 \colon l_1 = 1,\ldots,n_1,\ l_2 = 1,\ldots,n_2\},\ n = n_1 n_2,$$

where Δ_1 and Δ_2 are sampling intervals for x_1 and x_2, respectively.

In general LPAis applicable for regular and irregular data grids



Examples of regular and irregular grids

LPA technique

Let $x \in \mathbb{R}^2$ be a "center" (desired point) of the LPA. The estimate for the point $v \in \mathbb{R}^2$ in a neighborhood of x is of the form

$$y_h(x,v) = C^T \phi_h(x-v),\ \phi_h(x) = \phi(x/h),$$

$$\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_M(x))^T,$$

$$C = (C_1, C_2, \ldots, C_M)^T,\ x = (x_1, x_2),\ v = (v_1, v_2),$$

where $\phi \in \mathbb{R}^M$ is a vector of linear independent $2D$ polynomials

$$(-1)^{k_1+k_2} \frac{x_1^{k_1}}{k_1!} \frac{x_2^{k_2}}{k_2!}$$

with the polynomial degrees $k_1 + k_2$ from 0 up to m, $C \in \mathbb{R}^M$ is a vector of the parameters. The character h, $h > 0$, is reserved to denote a scale.

The total number of the polynomials in $\phi = (\phi_1, \ldots, \phi_M)^T$ and the unknown parameters in the vector $C = (C_1, \ldots, C_M)^T$ is equal to

$$M = \frac{(m+2)!}{2 \cdot m!} = (m+2)(m+1)/2.$$

For $m = 0,\ 1,\ 2$ and 3 complete sets of the $2D$ linear independent polynomials are of the form

$\phi_1 = 1,$ for $m = 0;$

$\phi_2 = -x_1,\ \phi_3 = -x_2,$ for $m = 1;$

$\phi_4 = x_1^2/2,\ \phi_5 = x_2^2/2,\ \phi_6 = x_1 x_2,$ for $m = 2;$

$\phi_7 = -x_1^3/6,\ \phi_8 = -x_2^3/6,\ \phi_9 = -x_1^2 x_2/2,\ \phi_{10} = -x_1 x_2^2/2,$ for $m = 3;$

with $M = 1$ for $m = 0$, $M = 3$ for $m = 1$, $M = 6$ for $m = 2$ and $M = 10$ for $m = 3$.

The formula for $y_h(x, v)$ can be interpreted as a truncated Taylor series that in the scalar form is as follows

$$y_h(x,v) = \sum_{i=1}^{M} C_i \phi_{i,h}(x-v).$$

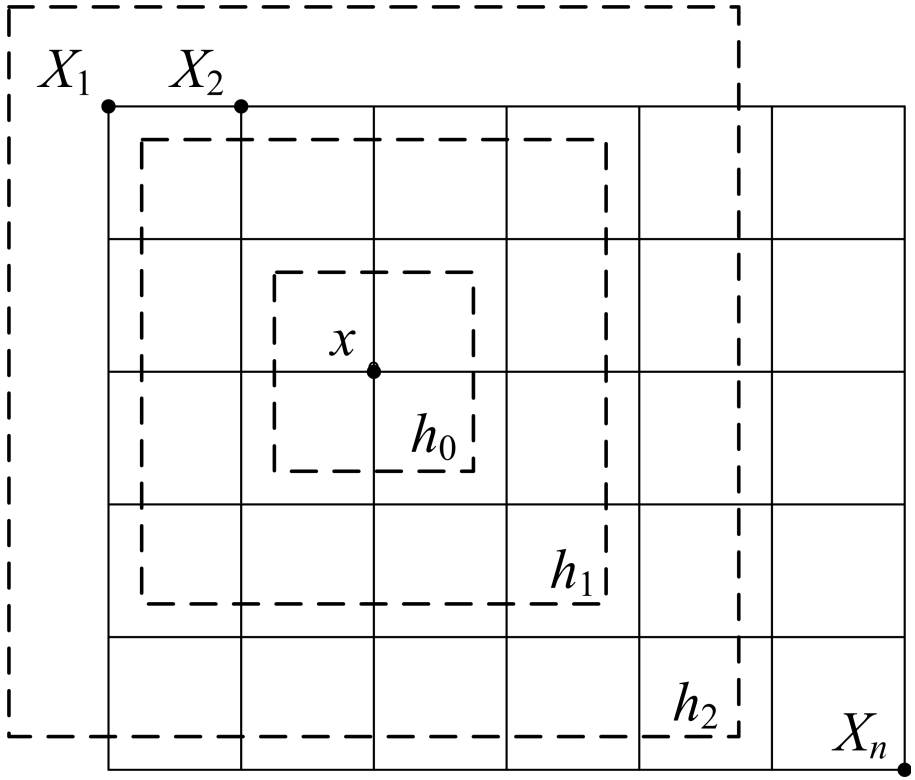
The weighted least square criterion $J_h(x,C)$ is used in the LPA in order to find the coefficient C :

$$J_h(x,C) = \sum_s w_h(x-X_s)(z_s - y_h(x,X_s))^2,$$

where the window function w formalizes the localization of fitting with respect to the center x and

$$w_h(x) = \frac{1}{h^2} w(\frac{x}{h}).$$

In the model $y_h(x, v)$ the argument x is considered as a fixed invariant parameter defining the center of the approximation and the approximations for different X_s are obtained by using this model with the varying second argument v assuming $v = X_s$.



Signal estimation

Let $\hat{y}_h(x)$ denote the LPA estimate (approximate) of $y(x)$, where the subscript h shows a dependence of the estimate on this scale parameter:

$$\hat{y}_h(x) = y_h(x,v)|_{v=x} = y_h(x,x) = \hat{C}^T\phi(0).$$

Note that $\phi(0) = \phi_h(0)$. For the polynomials (monomials) it yields

$$\hat{y}_h(x) = \hat{C}^T\phi(0) = \hat{C}_1,$$

because $\phi(0)$ is a vector-column $M \times 1$ with the single nonzero first element equal to 1,

$$\phi(0) = (1,0,\ldots,0)^T.$$

Derivative estimation

In order to obtain the estimate of the derivative $\partial^{(r_1,r_2)}y(x)$ we calculate the derivative $\partial_{v_1}^{r_1}\partial_{v_2}^{r_2}y_h(x,v)$ and following the very idea of the pointwise estimation use it for $v = x$:

$$\hat{y}_h^{(r_1,r_2)}(x) = \partial_{v_1}^{r_1}\partial_{v_2}^{r_2}y_h(x,v)|_{v=x} = C^T\partial_{v_1}^{r_1}\partial_{v_2}^{r_2}\phi((x-v)/h)|_{v=x}.$$

This derivative estimate is obtained in the form

$$\boxed{\hat{y}_h^{(r_1,r_2)}(x) = (-1/h)^{r_1+r_2}\hat{C}^T\partial_{(x_1,x_2)}^{(r_1,r_2)}\phi(0).}$$

For the monomials $\phi(x)$

$$\partial_{(x_1,x_2)}^{(r_1,r_2)}\phi(0) = (0,\ldots,0,\underbrace{(-1)^{r_1+r_2}}_{r_1,r_2-th},0,\ldots,0)^T$$

is a vector-row $1 \times M$ with the only nonzero element corresponding to the polynomial $(-1)^{r_1+r_2}x_1^{r_1}x_2^{r_2}/r_1!r_2!$ in $\phi(x)$. This nonzero derivative is equal to $(-1)^{r_1+r_2}$.

The LPA model of the degree m can be used in order to estimate the derivative $\partial_{x_1}^{r_1}\partial_{x_2}^{r_2}y(x)$ of the orders $r = (r_1, r_2)$,

$$|r_1 + r_2| \leq m.$$

The estimates of the signal and the derivatives are always defined by the above general formulas.

Estimate of C

Let us produce a minimization of $J_h(x, C)$ with respect to C.

The corresponding minimum condition $\partial_C J_h(x, C) = 0$ gives a linear normal equation

$$\Phi_h \hat{C} = \sum_s w_h(x - X_s)\phi_h(x - X_s)z_s,$$

$$\Phi_h = \sum_s w_h(x - X_s)\phi_h(x - X_s)\phi_h^T(x - X_s).$$

Provided $\det \Phi_h \neq 0$ the solution can be written as

$$\hat{C}(x, h) = \Phi_h^{-1} \sum_s w_h(x - X_s)\phi_h(x - X_s)z_s.$$

Estimate of signal

Let us use $\hat{C}(x,h)$ instead of C. Then

$$\hat{y}_h(x) = y_h(x,x) = \hat{C}^T(x,h)\phi(0).$$

Estimate of derivative

We obtain the derivative estimates in the form:

$$\hat{y}_h^{(r_1,r_2)}(x) = (-1/h)^{r_1+r_2} \cdot \hat{C}^T(x,h)\phi^{(r_1,r_2)}(0),$$

where

$$\phi^{(r_1,r_2)}(0) = \partial_{x_1}^{r_1}\partial_{x_2}^{r_2}\phi(x)|_{x=0}.$$

Kernel estimates

Estimate of signal

Inserting $\hat{C}$ in the estimate formula we find the signal estimate as an output of the linear filter:

$$\hat{y}_h(x) = \sum_s g_h(x, X_s) z_s,$$

$$g_h(x, X_s) = w_h(x - X_s)\phi_h^T(x - X_s)\Phi_h^{-1}\phi(0),$$
$$\Phi_h = \sum_s w_h(x - X_s)\phi_h(x - X_s)\phi_h^T(x - X_s),$$

where $g_h(x, X_s)$ is the impulse response of this filter for a fixed x.

The matrices Φ_h and Φ_h^{-1} are symmetric and $g_h(x, X_s)$ can be given in the equivalent form

$$g_h(x, X_s) = w_h(x - X_s)\phi^T(0)\Phi_h^{-1}\phi_h(x - X_s).$$

Estimate of derivative

Substituting $\hat{C}(x,h)$ into $\hat{y}_h^{(r)}(x)$ we obtain this estimate of the derivative $\partial^r y(x)$ in the form

$$\hat{y}_h^{(r)}(x) = \sum_s g_h^{(r)}(x, X_s) z_s,$$

$$g_h^{(r)}(x, X_s) = (-1/h)^{|r|} w_h(x - X_s) \phi_h^T(x - X_s) \Phi_h^{-1} \phi^{(r)}(0),$$

with $g_h^{(r)}(x, X_s)$ as the impulse response of this differentiating filter for a fixed x.

These formulas define a differentiation filter (differentiation operator) of the polynomial order (of the order) m.

Concerning the terminology we note that in statistics the weights g_h and $g_h^{(r)}$ are named kernels and then the estimates are kernel estimates.

Reproducing polynomial kernels

The introduced LPA estimates are accurate for polynomials.

Proposition. Let the estimates $\hat{y}_h(x)$ and $\hat{y}_h^{(r)}(x)$, $x \in \mathbb{R}^d$, be the LPA estimates of the order m and y be polynomial, $y \in P^m \subset \mathbb{R}^d$, $z_s = y(X_s)$.

Then, for any $h > 0$ such that $\det \Phi_h \neq 0$

$$\hat{y}_h(x) = y(x),\ x \in \mathbb{R}^d,$$

$$\hat{y}_h^{(r)}(x) = y^{(r)}(x),\ x \in \mathbb{R}^d.$$

Shift-Invariant kernels

Let us assume that the observation points X_s and the estimation points x belong to the same regular grid and this grid is infinite.

Then, the matrix Φ_h is invariant. The weights $g_h(x, X_s)$, $g_h^{(r)}(x, X_s)$ become shift-invariant and depending on the difference of the arguments x and X_s only:

$$g_h(x, X_s) = g_h(x - X_s)$$
$$g_h^{(r)}(x, X_s) = g_h^{(r)}(x - X_s).$$

The estimates can be represented in the convolution form with the shift-invariant on x (independent on x) kernels:

$$\hat{y}_h(x) = \sum_s g_h(u_s) z(x - u_s) = \sum_s g_h(x - X_s) z(X_s),$$
$$g_h(u_s) = w_h(u_s) \phi_h^T(u_s) \Phi_h^{-1} \phi(0).$$

and

$$\hat{y}_h^{(r)}(x) = \sum_s g_h^{(r)}(u_s) z(x - u_s) = \sum_s g_h^{(r)}(x - X_s) z(X_s),$$

$$g_h^{(r)}(u_s) = (-1/h)^{|r|} w_h(u_s) \phi_h^T(u_s) \Phi_h^{-1} \phi^{(r)}(0),$$

$$\Phi_h = \sum_s w_h(u_s) \phi_h(u_s) \phi_h^T(u_s).$$

In this representation the differences

$$u_s = x - X_s$$

are relative coordinates of the pixels X_s with respect to the estimation point x.

The considered passage to the shift-invariant estimates is valid for the general multidimensional signals, $x \in \mathbb{R}^d$.

Using the standard notation for multidimensional convolution, the estimates can be represented in the following compact form

$$\hat{y}_h(x) = (g_h \circledast z)(x),$$
$$\hat{y}_h^{(r)}(x) = (g_h^{(r)} \circledast z)(x).$$

Vanishing moments

Let X_{Z^d} be the infinite regular d-dimensional grid:

$$X_{\mathbb{Z}^d} = \{l_1\Delta_1, \ldots, l_d\Delta_d : l_j \in \mathbb{Z}, j = 1, \ldots, d\},$$

where Δ_j are the sampling intervals on the corresponding arguments.

Proposition. Let the kernels $g_h[k]$ and $g_h^{(r)}[k]$, $k \in \mathbb{Z}^d$, be polynomial of the order m. Then the following vanishing moment conditions hold for these kernels

$$\sum_{l \in \mathbb{Z}^d} g_h[l] l^i = \delta_{i,0},\ 0 \le |i| \le m,$$

$$\frac{1}{i!}\sum_{l\in\mathbb{Z}^d} g_h^{(r)}[l]l^i = \frac{(-1)^{|r|}}{\Delta^r}\delta_{i,r},\ 0 \le |i| \le m,\ 0 < |r| \le m,$$

where

$$l^i = l_1^{i_1} \bullet \ldots \bullet l_d^{i_d},\ \frac{l^i}{i!} = \frac{l_1^{i_1}}{i_1!} \bullet \ldots \bullet \frac{l_d^{i_d}}{i_d!},\ |i| = i_1 + \ldots + i_d.$$

For $d = 1$ the vanishing moment conditions for the smoothing kernel are

$$\sum_{l=-\infty}^{\infty} g_h[l]l^i = \delta_{i,0},\ 0 \le i \le m,$$

i.e. the kernel is normalized, $\sum_{l=-\infty}^{\infty} g_h[l] = 1$ and has zero higher moments up to the order m.

For the differentiation kernel the moment of the order $i = r$ is equal to $(-1/\Delta)^r$ and all other moments are zero.

Frequency domain

Classical discrete signal processing algorithms are mostly based on shift-invariant (time-invariant) linear operators. The frequency characteristics and the Fourier transform are efficient methods to characterize this type of operators as well as to implement fast algorithms for calculations.

Discrete-time Fourier transform

On the definition, $DTFT$ of the signal $z[k]$ is

$$\mathcal{F}\ \{z[k]\}\ = Z(e^{-j\bar{\omega}}) = \sum_{k\in\mathbb{Z}^d} e^{-jk\bar{\omega}} z[k].$$

Here we use the normalized frequency $\bar{\omega} = \omega \bullet \Delta$.

Standard transformations show that

$$\boxed{Y_h(e^{-j\bar{\omega}}) = G_h(e^{-j\bar{\omega}})Z(e^{-j\bar{\omega}}).}$$

where

$$G_h(e^{-j\bar{\omega}}) = \mathcal{F}\{g_h[k]\} = \sum_{l\in\mathbb{Z}^d} e^{-jl\bar{\omega}} g_h[l]$$

is the $DTFT$ of the kernel g_h.

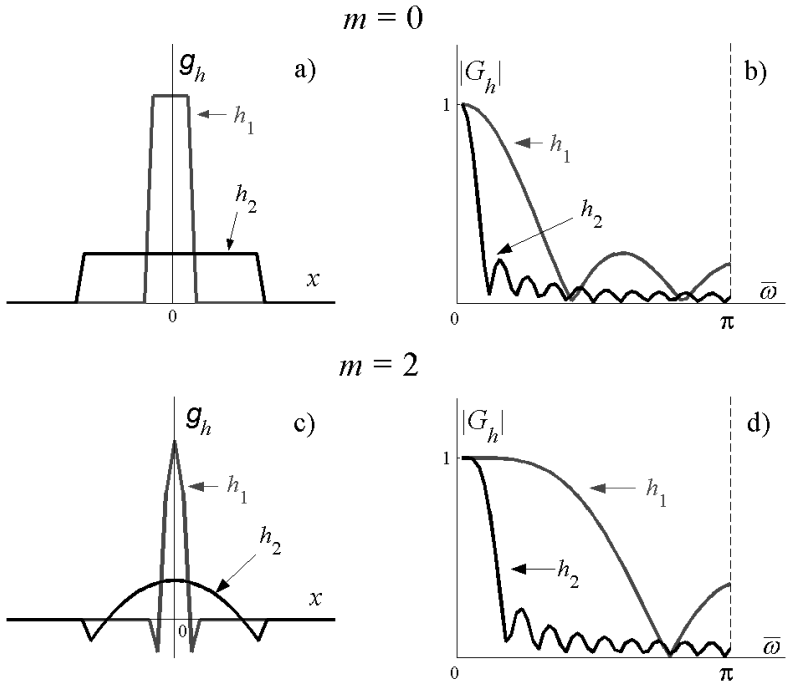
In a similar way, the transfer function of the differentiation operator is of the form

$$\boxed{G_h^{(r)}(e^{-j\bar{\omega}}) = \mathcal{F}\{g_h^{(r)}[k]\} = \sum_{k\in\mathbb{Z}^d} g_h^{(r)}[k]\exp(-jk\bar{\omega})}$$

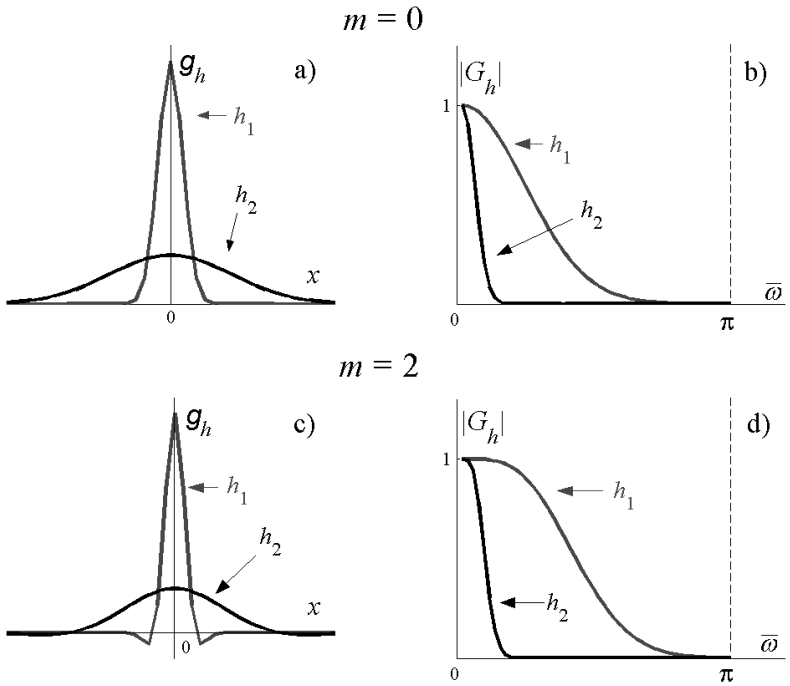
and the input-output link for the differentiating operator is

$$\boxed{Y_h^{(r)}(e^{-j\bar{\omega}}) = G_h^{(r)}(e^{-j\bar{\omega}})Z(e^{-j\bar{\omega}}).}$$

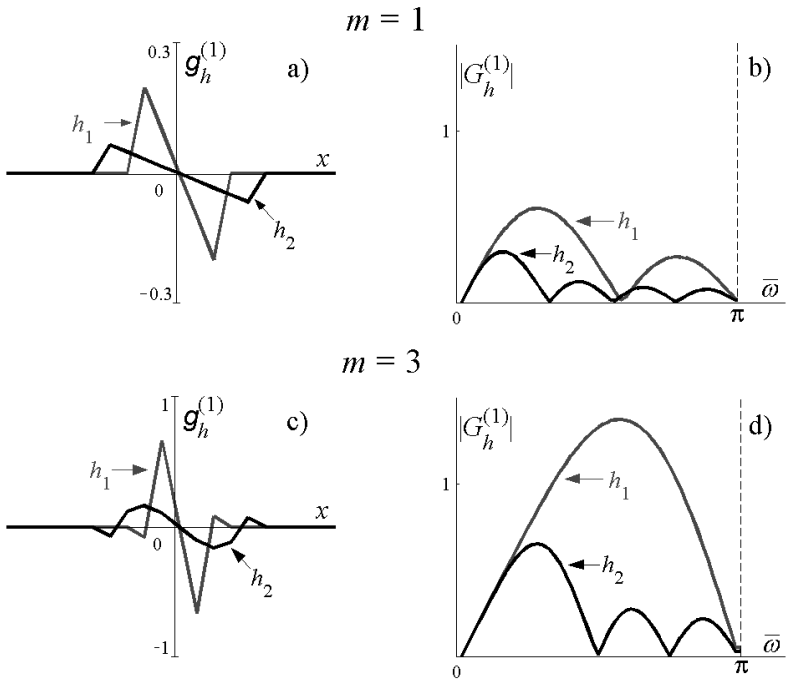
1D examples (demo_DrawKernel1D.m)



The kernels g_h and frequency characteristics $|G_h|$ of the smoothing filters, the rectangular w, $m = 0$ and $m = 2$.

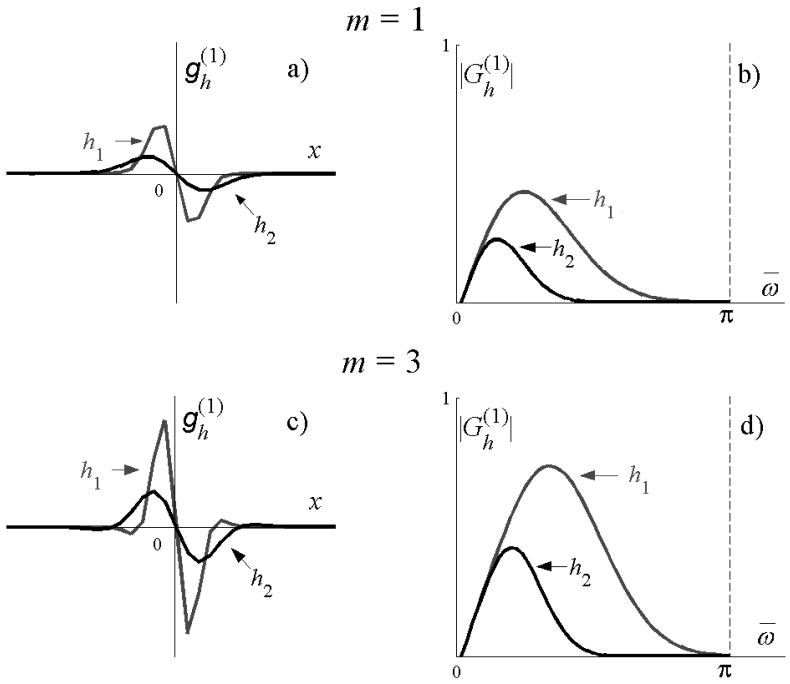


The kernels g_h and frequency characteristics $|G_h|$ of the discrete smoothing filters. The LPA design with the Gaussian window and the degrees $m = 0, m = 2$.



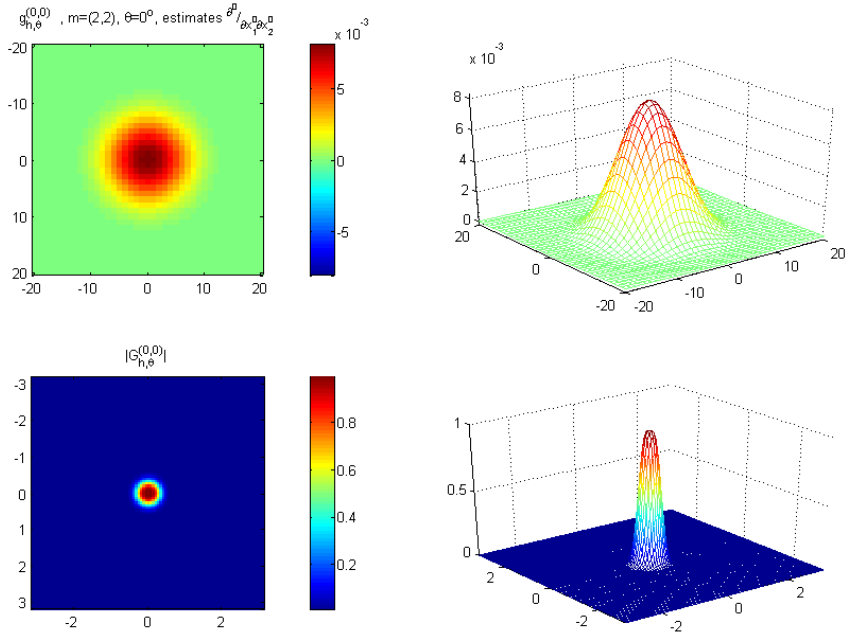
The kernels $g_h^{(1)}$ and frequency characteristics $|G_h^{(1)}|$ of the differentiation filters.

The rectangular window and the degrees $m = 1, m = 3$.

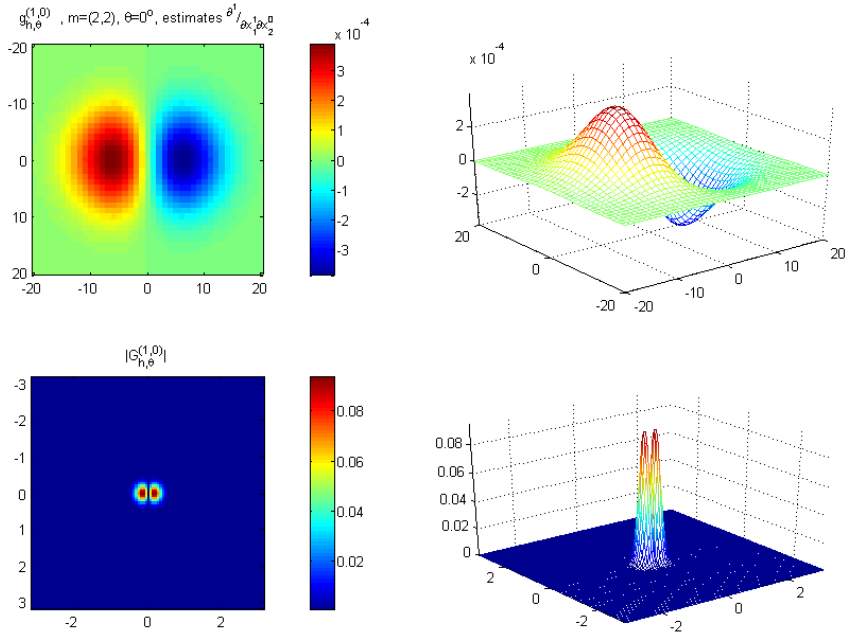


The kernels $g_h^{(1)}$ and frequency characteristics $|G_h^{(1)}|$ of the differentiation filters.
The Gaussian window and the degrees $m = 1$, $m = 3$.

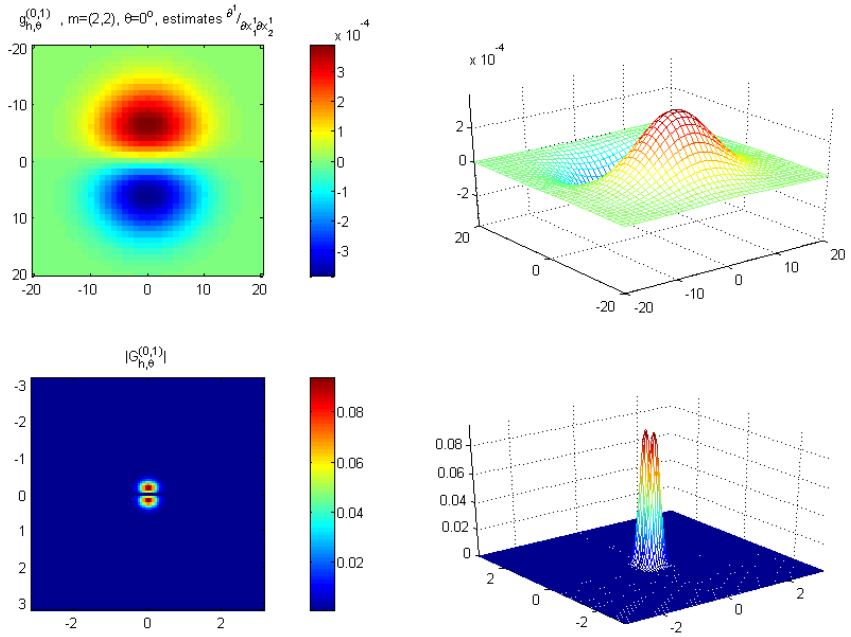
2D examples (demo_CreateLPAKernels.m)



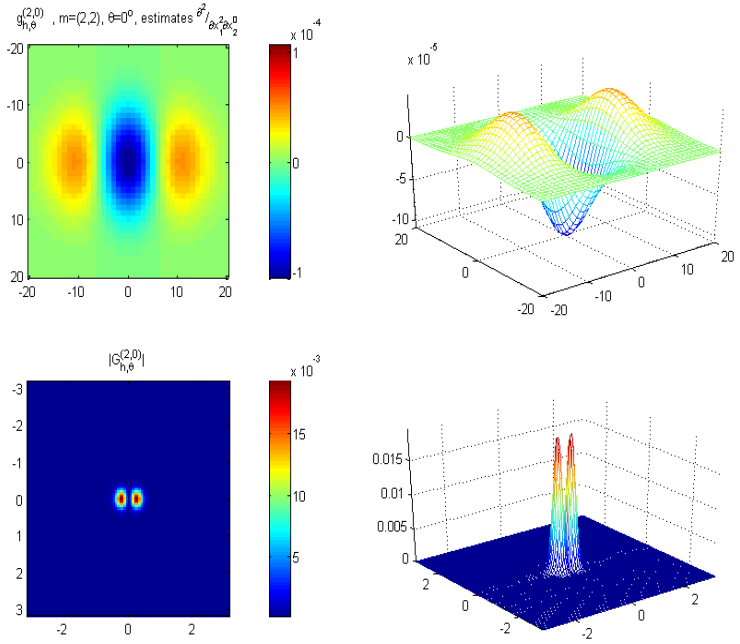
The smoothing kernel g_h and its amplitude frequency characteristic $|G_h|$: Gaussian window, $m = 2$. The lowpass filter with a peak of the frequency characteristic at $\bar{\omega} = 0$.



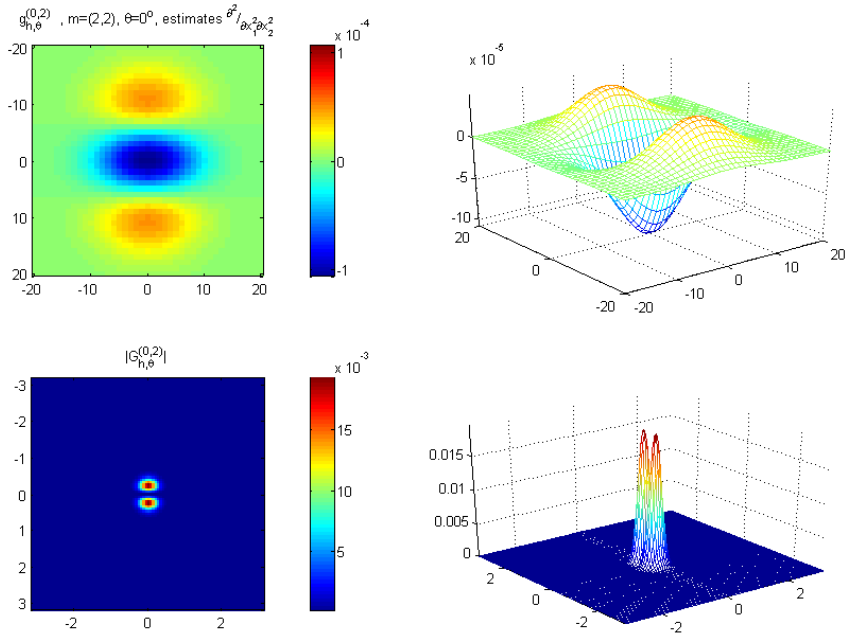
The differentiation kernel $g_h^{(1,0)}$ and its amplitude frequency characteristic $|G_h^{(1,0)}|$: Gaussian window, $m = 2$. The bandpass filter $|G_h^{(1,0)}| = 0$ at $\bar{\omega} = 0$.



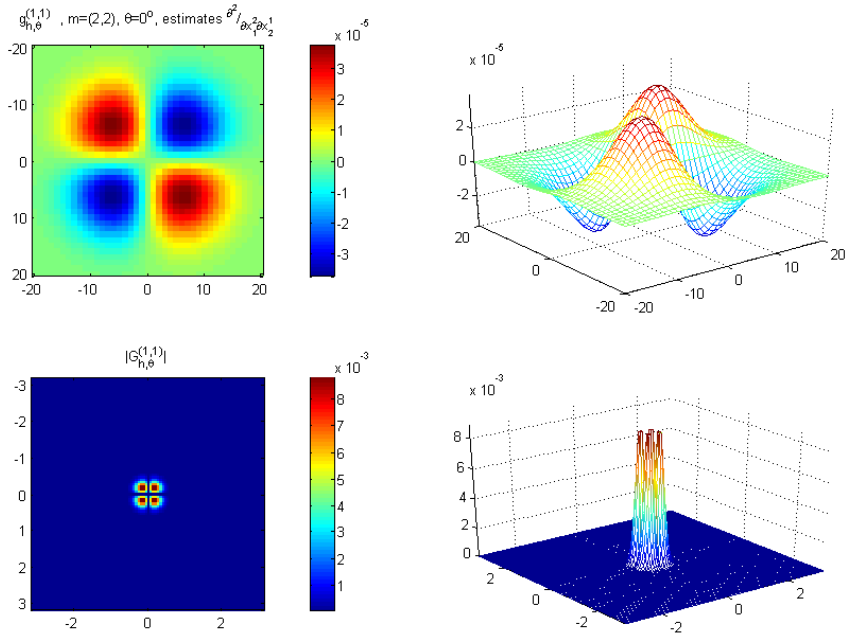
The differentiation kernel $g_h^{(0,1)}$ and its amplitude frequency characteristic $|G_h^{(0,1)}|$: Gaussian window, $m = 2$. The bandpass filter $|G_h^{(0,1)}| = 0$ at $\bar{\omega} = 0$.



The differentiation kernel $g_h^{(2,0)}$ and its amplitude frequency characteristic $|G_h^{(2,0)}|$: Gaussian window, $m = 2$. The bandpass filter $|G_h^{(2,0)}| = 0$ at $\bar{\omega} = 0$.



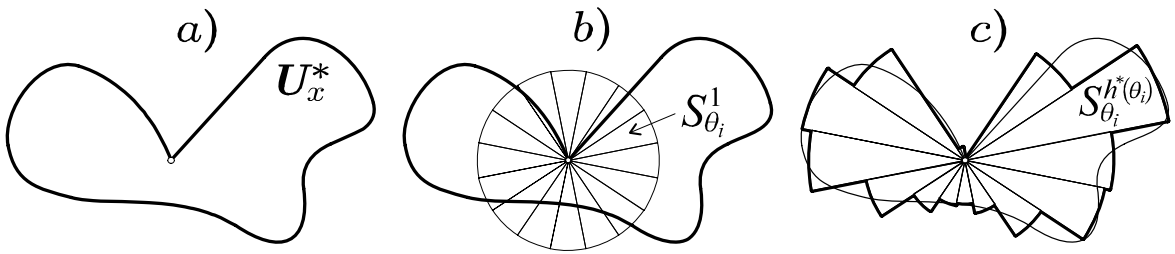
The differentiation kernel $g_h^{(0,2)}$ and its amplitude frequency characteristic $|G_h^{(0,2)}|$: Gaussian window, $m = 2$. The bandpass filter $|G_h^{(0,2)}| = 0$ at $\bar{\omega} = 0$.



The differentiation kernel $g_h^{(1,1)}$ and its amplitude frequency characteristic $|G_h^{(1,1)}|$: Gaussian window, $m = 2$. The bandpass filter $|G_h^{(1,1)}| = 0$ at $\bar{\omega} = 0$.

Directional *LPA*

Idea



Consider a ball (disk) of the radius h defining a spherical neighborhood of x,

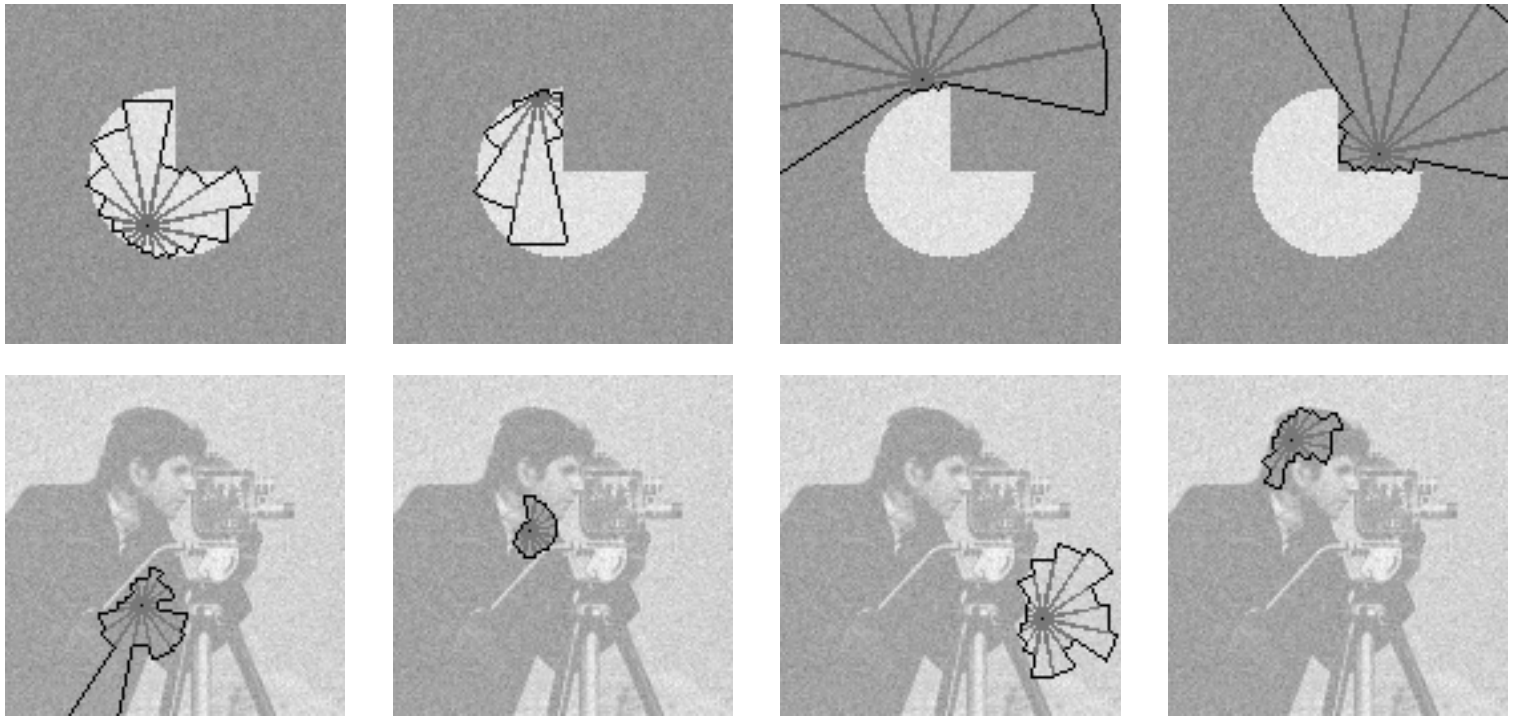
$$\mathrm{B}_h = \{u : \|x - u\| \leq h\}.$$

Introduce a sectorial partition of this ball with K nonoverlapping sectors having x as a common vertex. The adaptivity technique is used to find the varying scale for each sector independently.

In this way, we obtain K estimates $\hat{y}_{h,\theta_i}(x)$, $i = 1, \ldots, K$, with K nonoverlapping supports $S_{\theta_i}^h$ covering the ball B_h. The union of the supports of

these sectors more is not a ball B_h but a $starshaped$ set.

Directional LPA estimates $\hat{y}_{h,\theta}(x) = (g_{h,\theta} \circledast z)(x)$, where $g_{h,\theta}$ are directional sectorial kernels.



Adaptive size sectorial neighborhoods are obtained by the $LPA - ICI$ algorithm.

Directional kernel design

The standard LPA technique is modified in order to meet specific requirements of directional estimation. Modifications mainly concern the window function w which becomes nonsymmetric directional, the scales and orders of the LPA which are independently specified for each of the variables. We name this modified method by the directional LPA.

Polynomials

Let us introduce a set of the $2D$ polynomials in the form

$$(-1)^{k_1+k_2} x_1^{k_1} x_2^{k_2} / k_1! k_2! \ , \ k_1 = 0,\ldots,m_1, \ k_2 = 0,\ldots,m_2,$$

where the powers m_1 and m_2 can be selected independently.

These polynomials are linearly independent with a total number of the polynomials

$$M = (m_1 + 1)(m_2 + 1)$$

and the maximum power of the polynomials equal to $m_1 + m_2$.

It is very different from the one used before where the scalar m defines the maximum power of the $2D$ polynomials as $0 \le k_1 + k_2 \le m$.

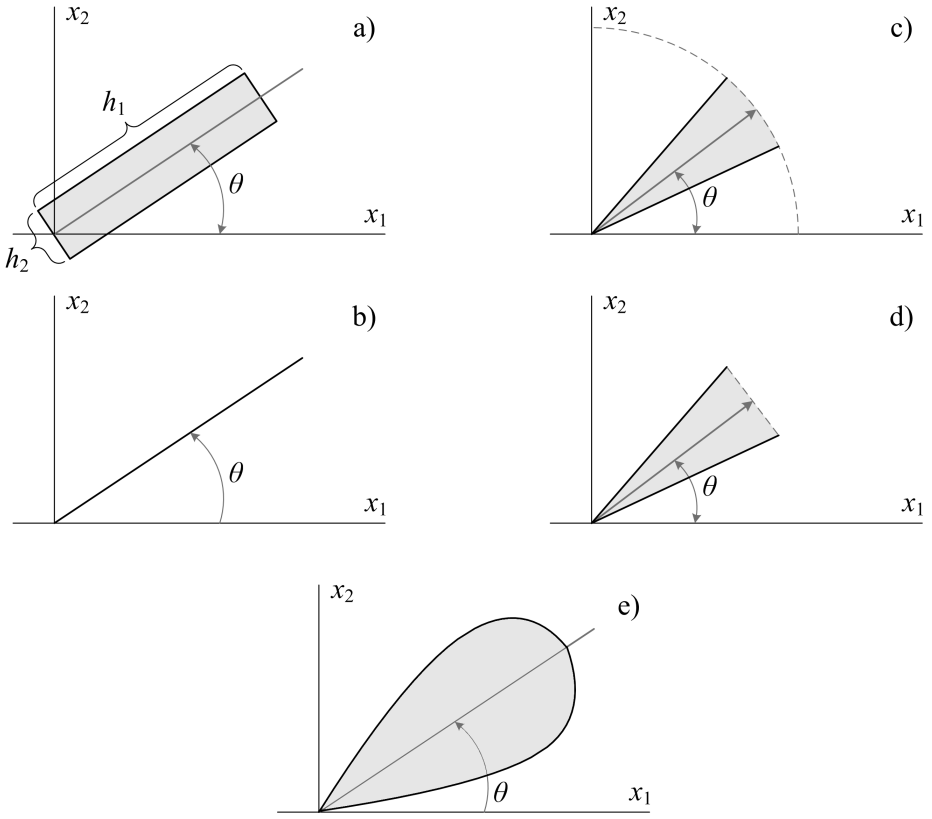
The LPA estimates are defined according to the general formula

$$y_h(x,v) = C^T \phi_h(x-v),\ \phi_h(x) = \phi(x/h),$$
$$\phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_M(x))^T,$$
$$C \in \mathbb{R}^M,\ x, v \in \mathbb{R}^2,\ x/h = (x_1/h_1, x_2/h_2).$$

For example for $m = (2,1), M = (m_1+1)(m_2+1) = 6$, the set of the polynomials $\phi_k(x), k = 1, \ldots, 6$, can be given in the form

$$\phi_1 = 1,\ \phi_2 = -x_1,\ \phi_3 = -x_2,$$
$$\phi_4 = x_1^2/2,\ \phi_5 = -x_1 x_2,\ \phi_6 = -x_1^2 x_2/2.$$

Directional windowing



Rotation

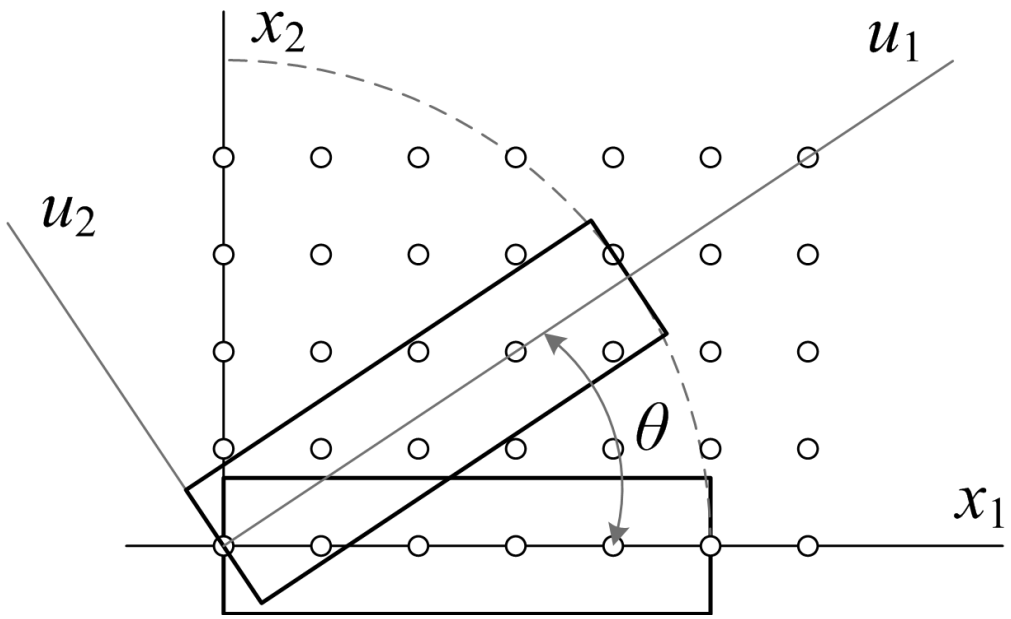
The design of the directional kernel for the direction θ consists of two successive steps.

The first one is a rotation of the support of the basic window w_h. The finite discrete grid support is found for the new direction as a result of this step.

The second step is the LPA design of the kernel for this new support. Let us introduce the "basic" x and "rotated" u coordinate systems:

$$u = U(\theta)x,\ x = U^T(\theta)u,$$

$$U(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},\ u(x,\theta) = U(\theta)x.$$



A number of pixels as well as their location inside the support depends on the rotation angle θ.

Shift-invariant kernels

The LPA in the new variables u_1, u_2 is used in order to obtain the accurate polynomial kernel for any discrete support obtained after rotation.

The rotated kernels are reproducing with respect to the signals polynomial in the variable u and satisfy to the corresponding vanishing moment conditions also in the variables u.

The shift invariant estimates have a convolution form

$$\hat{y}_{h,\theta}(\Delta k) = \sum_{s\in\mathbb{Z}^2} g_{h,\theta}[k-s]z_s = \sum_{s\in\mathbb{Z}^2} g_{h,\theta}[s]z_{k-s},$$

where

$$g_{h,\theta}[s] = w_h(U(\theta)\Delta s)\phi_h^T(U(\theta)\Delta s)\Phi_h^{-1}\phi(0),$$

$$\Phi_h = \sum_{s\in\mathbb{Z}^2} w_h(U(\theta)\Delta s)\phi_h(U(\theta)\Delta s)\phi_h^T(U(\theta)\Delta s).$$

In a similar way the derivative estimate can be written in the form

$$\hat{y}_{h,\theta}^{(r)}(\Delta k) = \sum_{s\in\mathbb{Z}^2} g_{h,\theta}^{(r)}[k-s]z_s = \sum_{s\in\mathbb{Z}^2} g_{h,\theta}^{(r)}[s]z_{k-s},$$

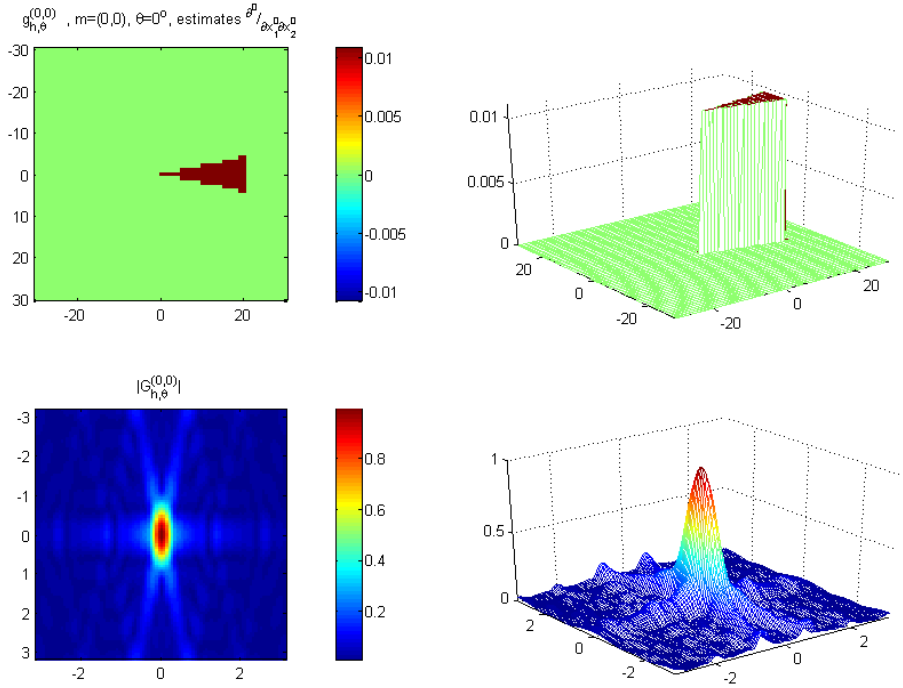
where

$$g_{h,\theta}^{(r)}[s] = \frac{(-1)^{|r|}}{h_1^{r_1}h_2^{r_2}} w_h(U(\theta)\Delta s)\phi_h^T(U(\theta)\Delta s)\Phi_h^{-1}\phi^{(r)}(0).$$

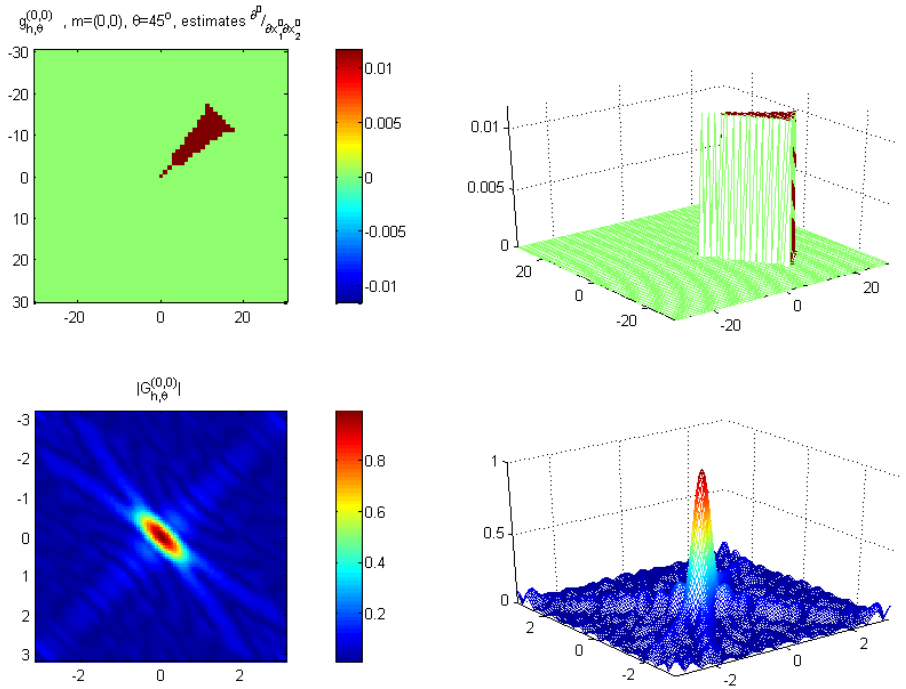
For a small sampling interval Δ

$$G_{h,\theta}(e^{-j\bar{\omega}}) \simeq G_{h,0}(e^{-jU(\theta)\bar{\omega}}).$$

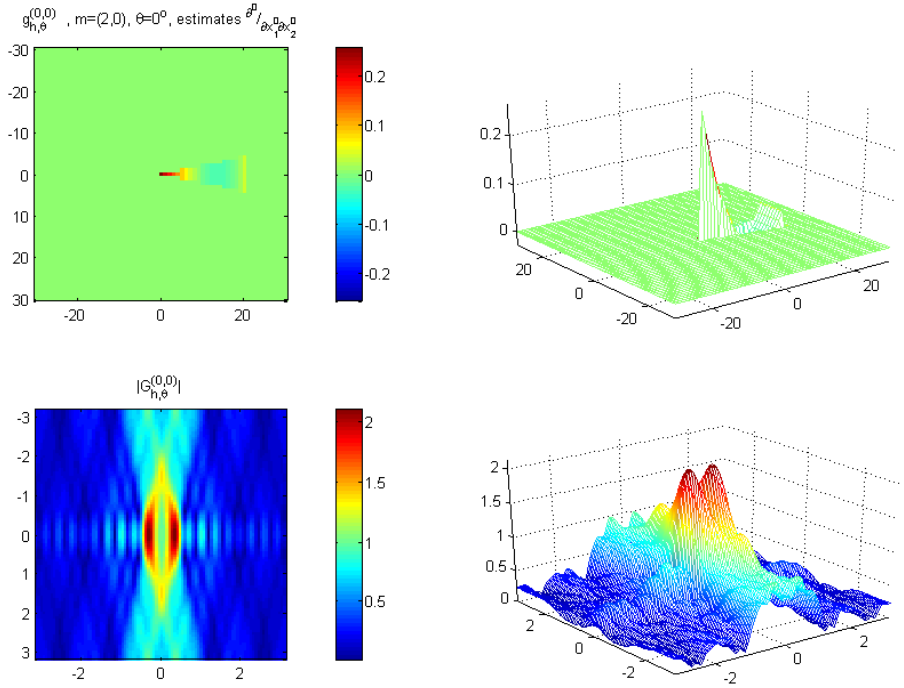
Numerical examples (demo_CreateLPAKernels.m)



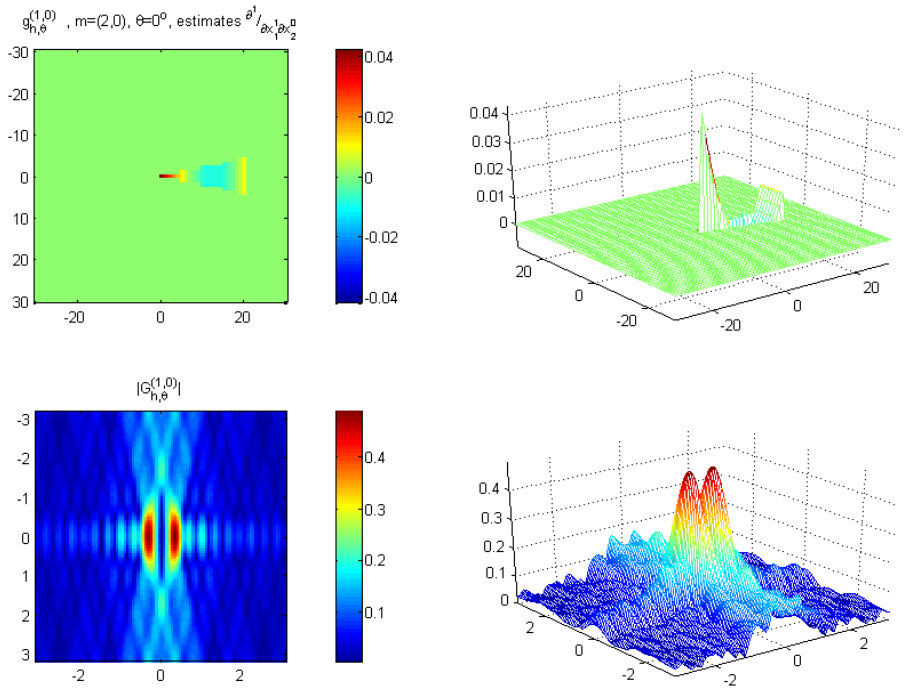
The basic kernel ($\theta = 0$) of the order $m = (0,0)$ with the sectorial uniform window-support, $h_1 = h_2 = 21$.



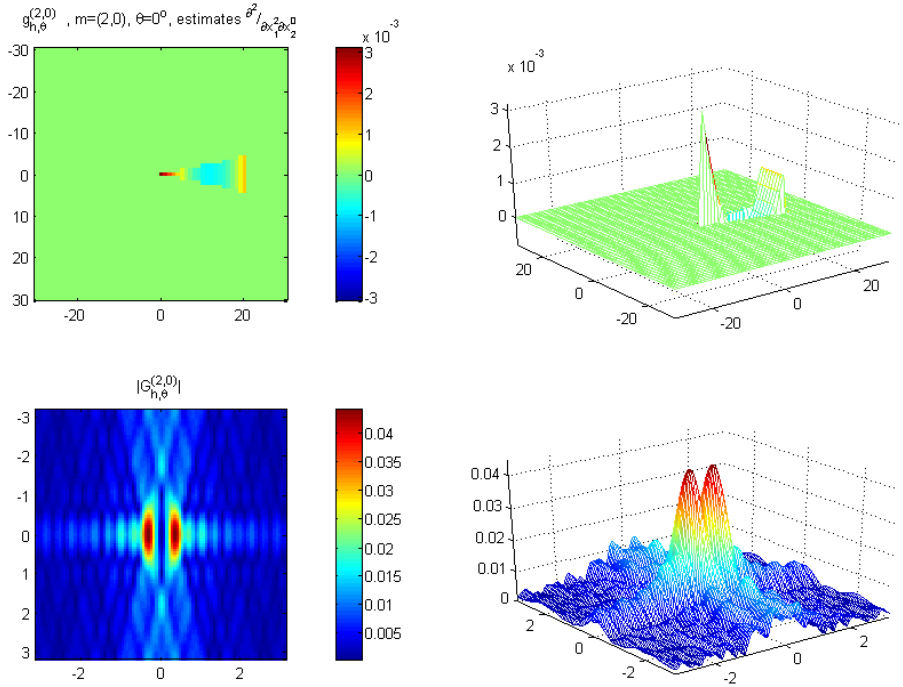
A rotation of the basic kernel $g_h^{(0,0)}$, $\theta = 45^0$.



The order $m = (2,0)$ results in a quadratic shape of $g_h^{(0,0)}$ clearly seen in $3D$ curve.

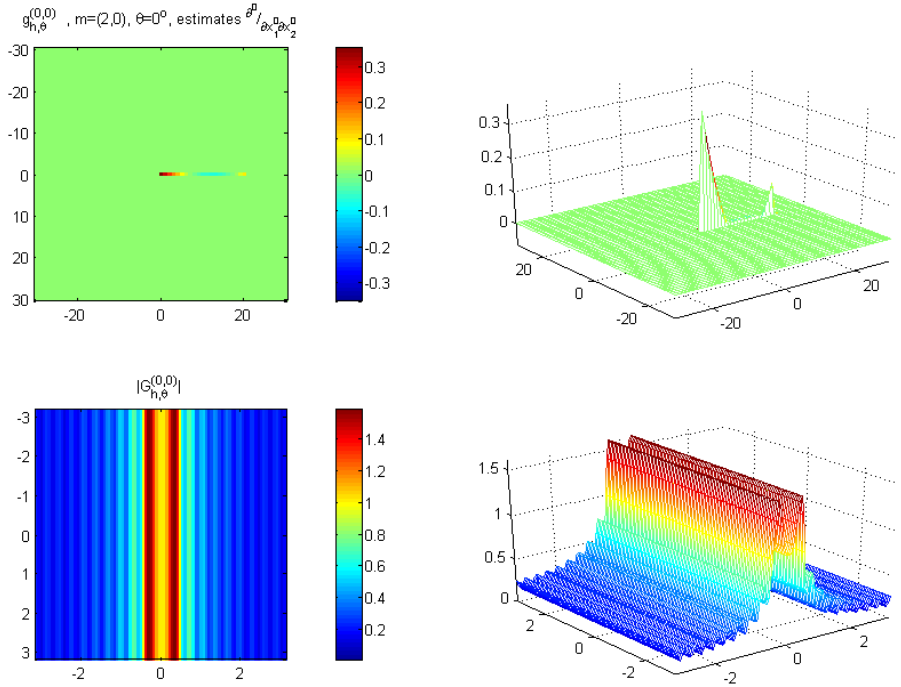


The differentiating basic kernel $g_h^{(1,0)}$, $\theta = 0$. The first order differentiating on x_1, $\partial^{(1,0)} = \partial_{x_1}$, $|G_h^{(1,0)}(0)| = 0$.

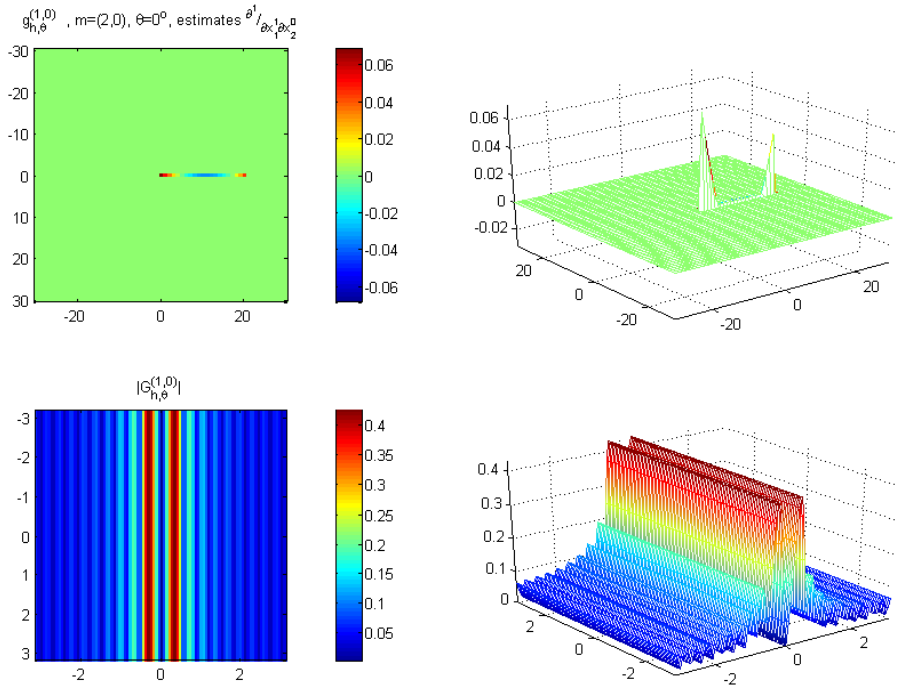


The differentiating basic kernel $g_h^{(2,0)}$, $\theta = 0$. The second order differentiating on x_1, $\partial^{(2,0)} = \partial^2_{x_1}$, $|G_h^{(2,0)}(0)| = 0$.

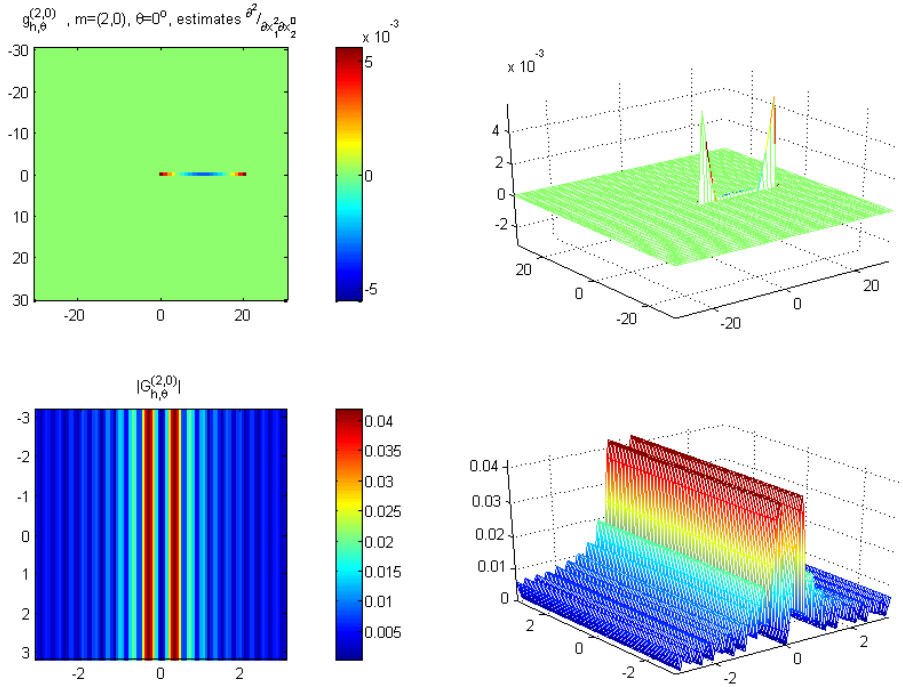
Line-wise kernels



The basic smoothing kernel $g_h^{(0,0)}$, $\theta = 0$, line-wise support, $m = (2,0)$.



The differentiating basic kernel $g_h^{(1,0)}$, $\theta = 0$. The first order differentiating on x_1, $\partial^{(1,0)} = \partial_{x_1}$, $|G_h^{(1,0)}(0)| = 0$.



The differentiating basic kernel $g_h^{(2,0)}$, $\theta = 0$. The second order differentiating on x_1, $\partial^{(2,0)} = \partial_{x_1}^2$, $|G_h^{(2,0)}(0)| = 0$.

LPA Accuracy

The LPA accuracy is characterized by an error defined as a difference between the true signal value and the estimate:

$$e_y(x,h) = y(x) - \hat{y}_h(x),$$
$$e_{y^{(r)}}(x,h) = y^{(r)}(x) - \hat{y}_h^{(r)}(x).$$

These errors are composed of the systematic (bias) and random components corresponding to the deterministic y and the random noise ε, respectively.

Bias and variance

Bias

The bias of the estimate is a difference between the true signal and the expectation of the estimate:

$$m_{\hat{y}_h}(x,h) = y(x) - E\{\hat{y}_h(x)\},$$
$$m_{\hat{y}_h^{(r)}}(x,h) = y^{(r)}(x) - E\{\hat{y}_h^{(r)}(x)\},$$

where $E\{\bullet\}$ means the mathematical expectation with respect to the random noise.

The estimates are defined by the convolutions:

$$\hat{y}_h(x) = (g_h \circledast z)(x),\ \hat{y}_h^{(r)}(x) = (g_h^{(r)} \circledast z)(x).$$

Elementary calculations give

$$E\{\hat{y}_h(x)\} = E\{(g_h \circledast z)(x)\} = (g_h \circledast E\{z\})(x) = (g_h \circledast y)(x),$$
$$E\{\hat{y}_h^{(r)}(x)\} = E\{(g_h^{(r)} \circledast z)(x)\} = (g_h^{(r)} \circledast E\{z\})(x) = (g_h^{(r)} \circledast y)(x).$$

Then, the bias of the estimates is

$$m_{\hat{y}_h}(x,h) = y(x) - (g_h \circledast y)(x),\ m_{\hat{y}_h^{(r)}}(x,h) = y^{(r)}(x) - (g_h^{(r)} \circledast y)(x).$$

Variance

The random estimation errors are

$$e_y^0(x,h) = -(g_h \circledast \varepsilon)(x),$$

$$e_{y^{(r)}}^0(x,h) = -(g_h^{(r)} \circledast \varepsilon)(x).$$

Assuming that the random ε are $i.i.d.$ with the variance σ^2 we obtain for the variance of the estimate $\hat{y}_h(x)$:

$$\sigma_{\hat{y}_h}^2(x,h) = E\{(e_y^0(x,h))^2\} = E\{(\sum_{l\in\mathbb{Z}^d} g_h(l\Delta)\varepsilon(x-l\Delta))^2\} =$$

$$\sigma^2 \sum_{k\in\mathbb{Z}^d} g_h^2(k\Delta).$$

A similar formula can be derived for the derivative estimate $\hat{y}_h^{(r)}(x)$.

It proves that the variance of the smoothing and differentiating estimates can be given as follows

$$\sigma^2_{\hat{y}_h}(x,h) = \sigma^2 \|g_h\|^2,$$

$$\sigma^2_{\hat{y}^{(r)}_h}(x,h) = \sigma^2 \|g^{(r)}_h\|^2,$$

where the Euclidian norm of the kernels means

$$\|g_h\|^2 = \sum_{k \in \mathbb{Z}^d} g^2_h(\Delta k), \; \|g^{(r)}_h\|^2 = \sum_{k \in \mathbb{Z}^d} (g^{(r)}_h(\Delta k))^2.$$

In this particular case the variances $\sigma^2_{\hat{y}_h}(x,h)$ and $\sigma^2_{\hat{y}^{(r)}_h}(x,h)$ do not depend on x. However, for generality, we keep the argument x in the variance notation as in further chapters we use estimates with the variance depending on x.

Accuracy results

($H1$) y is deterministic with bounded derivative of the order $m+1$;

($H2$) The order of the LPA is equal to m. The sampling intervals are the same for all variables, $\Delta_i = \Delta$, $i = 1, \ldots, d$.

Proposition. Let the hypotheses $H1 - H2$ hold, the noise be white with variance σ^2 and h, Δ, Δ/h, $\Delta^d/h^d \to 0$.

Then, the accuracy of the multivariate discrete LPA estimates is defined as following:

for the bias

$$m_{\hat{y}_h}(x,h) \sim (-1)^m h^{m+1} \sum_{|k|=m+1} y^{(k)}(x) \frac{1}{k!} \int_{\mathbb{R}^d} g(u) u^k du,$$

$$m_{\hat{y}_h^{(r)}}(x,h) \sim (-1)^m h^{m+1-|r|} \sum_{|k|=m+1} y^{(k)}(x) \frac{1}{k!} \int_{\mathbb{R}^d} g^{(r)}(u) u^k du,$$

for the variance

$$\frac{1}{\sigma^2}\sigma^2_{\hat{y}_h}(x,h) \sim \frac{\Delta^d}{h^d}\int_{\mathbb{R}^d} g^2(u)du,$$

$$\frac{1}{\sigma^2}\sigma^2_{\hat{y}_h^{(r)}}(x,h) \sim \frac{\Delta^d}{h^{d+2|r|}}\int_{\mathbb{R}^d} (g^{(r)}(u))^2 du.$$

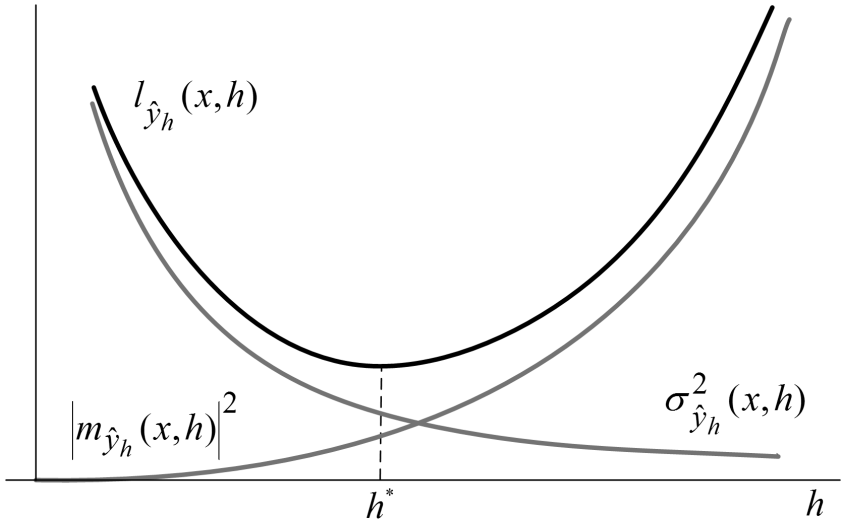
For the quadratic risks we have

$$l_{\hat{y}_h}(x,h) = m^2_{\hat{y}_h}(x,h) + \sigma^2_{\hat{y}_h}(x,h),$$

$$l_{\hat{y}_h^{(r)}}(x,h) = m^2_{\hat{y}_h^{(r)}}(x,h) + \sigma^2_{\hat{y}_h^{(r)}}(x,h).$$

Bias-variance trade-off

Ideal scale



In the asymptotic analysis the systematic error for $d = 1$ is evaluated as

$$m_{\hat{y}_h}(x,h) = y(x) - \int g(u)y(x-hu)du.$$

The variance of the random components is defined as

$$\sigma^2_{\hat{y}_h}(x,h) = \sigma^2 \frac{\Delta}{h} B_g,\ B_g = \int g^2(u)du,$$

where Δ is a small sampling period and σ^2 is the variance of the noise.

We may conclude that the main term of the systematic error is

$$|m_{\hat{y}_h}(x,h)| = h^{m+1}|y^{(m+1)}(x)|A_g,$$

where m is the order of the polynomial approximation and
$A_g = |\int g(u)u^{m+1}du|/(m+1)!$.

This optimal h can be found by minimizing the mean squared error

$$l_{\hat{y}_h}(x,h) \triangleq E\{e_y^2(x,h)\} = m_{\hat{y}_h}^2(x,h) + \sigma_{\hat{y}_h}^2(x,h).$$

We have the mean squared risk function

$$l_{\hat{y}_h}(x,h) = (h^{m+1}A_g y^{(m+1)}(x))^2 + \sigma^2 \frac{\Delta}{h} B_g$$

convex on h.

The equation $\partial l_{\hat{y}_h}(x,h)/\partial h = 0$ gives for the optimal scale

$$h^* = \left(\frac{\sigma^2 B_g \Delta}{2(m+1)(A_g y^{(m+1)}(x))^2}\right)^{1/(2m+3)}.$$

Bias-variance balance

It can be seen that

$$l_{\hat{y}_h}(x,h^*) = m^2_{\hat{y}_h}(x,h^*) + \sigma^2_{\hat{y}_h}(x,h^*) =$$

$$\sigma^2_{\hat{y}_h}(x,h^*)(1+\gamma^2),$$

where

$$\gamma^2 = \frac{m^2_{\hat{y}_h}(x,h^*(x))}{\sigma^2_{\hat{y}_h}(x,h^*(x))}$$

and

$$\gamma^2 = \frac{1}{2(m+1)}.$$

The ideal balance between the standard deviation and the bias of estimation can be represented in the form of the following inequalities

$$|m_{\hat{y}_h}(x,h)| \begin{cases} < \gamma \cdot \sigma_{\hat{y}_h}(x,h) \text{ if } h < h^*(x), \\ > \gamma \cdot \sigma_{\hat{y}_h}(x,h) \text{ if } h > h^*(x). \end{cases}$$

The ICI rule uses the estimate of the signal and the variance of the estimate in order to tests the hypotheses $h \lessgtr h^*(x)$ and to select in this way the adaptive scale close to the ideal $h^*(x)$.

Derivative estimation

It can be shown that

$$l^*_{\hat{y}_h^{(r)}}(x) = l_{\hat{y}_h^{(r)}}(x, h_r^*(x)) = \sigma^2_{\hat{y}_h^{(r)}}(x, h_r^*(x))(1 + \gamma^2_{|r|}).$$

where

$$\gamma^2_{|r|} = \frac{(m_{\hat{y}_h^{(r)}}(x, h_r^*(x)))^2}{\sigma^2_{\hat{y}_h^{(r)}}(x, h_r^*(x))}$$

and

$$\gamma^2_{|r|} = \frac{1 + 2|r|}{2(m + 1 - |r|)}$$

It can be checked that

$$|m_{\hat{y}_h^{(r)}}(x,h)| \begin{cases} < \gamma_{|r|} \cdot \sigma_{\hat{y}_h^{(r)}}(x,h) \text{ if } h < h_r^*(x), \\ > \gamma_{|r|} \cdot \sigma_{\hat{y}_h^{(r)}}(x,h) \text{ if } h > h_r^*(x). \end{cases}$$

ICI Adaptive Scale

The problem of varying scale pointwise adaptation has received a powerful impetus in connection with a number of novel ideas developed in mathematical statistics.

These methods known under a generic name Lepski$'$s approach and developed by Lepski O., Nemirovski A., Goldenshluger A., Spokoiny V.

Overall, the algorithm searches for the largest local vicinity of the point of estimation x where the LPA assumptions fit well to the data. The estimates are calculated for a few scales and compared. The adaptive scale is defined as the largest one of those for which the estimate does not differ significantly from the estimates corresponding to the smaller scales.

These methods are from a class of quality-of-fit statistics based on the multiple statistical hypotheses testing.

We present this methods in the form known as the intersection of confidence intervals (ICI) rule.

Foundations

The ICI rule for the adaptive scale selection is derived from the accuracy analysis of the LPA estimates.

Signal estimation

Let $y(x)$ and $\hat{y}_h(x)$ be the signal and its LPA estimate, respectively, with the estimate defined according to as

$$\hat{y}_h(x) = \sum_s g_h(x, X_s) z_s.$$

The estimation error e_y can be represented as

$$|e_y(x,h)| = |y(x) - \hat{y}_h(x)| \leq |m_{\hat{y}_h}(x,h)| + |e^0(x,h)|.$$

where $m_{\hat{y}_h}(x,h)$ is the bias.

The random estimation error $e_y^0(x,h) = -\sum_s g_h(x, X_s)\varepsilon_s$.

Assuming that the noise ε_s is Gaussian the random error $e_y^0(x,h)$ is subject to the Gaussian probability density $N(0, \sigma^2_{\hat{y}_h}(x,h))$.

The following inequality holds with the probability $p = 1 - \beta$

$$|e_y^0(x,h)| \le \chi_{1-\beta/2} \cdot \sigma_{\hat{y}_h}(x,h),$$

where $\chi_{1-\beta/2}$ is $(1-\beta/2)-th$ quantile of the standard Gaussian distribution $N(0,1)$.

It has been shown that there exists an optimal balance between the bias and the standard deviation of the random error having a form of the inequality:

$$|m_{\hat{y}_h}(x,h)| \begin{cases} < \gamma \cdot \sigma_{\hat{y}_h}(x,h) \text{ if } h < h^*(x), \\ > \gamma \cdot \sigma_{\hat{y}_h}(x,h) \text{ if } h > h^*(x). \end{cases}$$

Using the first inequality we obtain that for $h \le h^*(x)$

$$|e_y(x,h)| \le \gamma \cdot \sigma_{\hat{y}_h}(x,h) + \chi_{1-\beta/2}\sigma_{\hat{y}_h}(x,h) \le \Gamma \cdot \sigma_{\hat{y}_h}(x,h),$$

$$\Gamma = \gamma + \chi_{1-\beta/2}.$$

This inequality is equivalent to

$$|y(x) - \hat{y}_h(x)| \leq \Gamma \cdot \sigma_{\hat{y}_h}(x,h)$$

and can be rewritten in the form

$$\hat{y}_h(x) - \Gamma \cdot \sigma_{\hat{y}_h}(x,h) \leq y(x) \leq \hat{y}_h(x) + \Gamma \cdot \sigma_{\hat{y}_h}(x,h),$$

if $h \leq h^*(x)$.

Define the confidence interval $Q(h)$ of the estimate $\hat{y}_h(x)$ as

$$Q(h) = [\hat{y}_h(x) - \Gamma \cdot \sigma_{\hat{y}_h}(x,h),\ \hat{y}_h(x) + \Gamma \cdot \sigma_{\hat{y}_h}(x,h)].$$

Provided $h < h^*(x)$ we can claim that with the probability p

$$y(x) \in Q(h).$$

Let

$$H = \{h_1, \ldots, h_J\},$$

where $h_j < h_i$, if $j < i$.

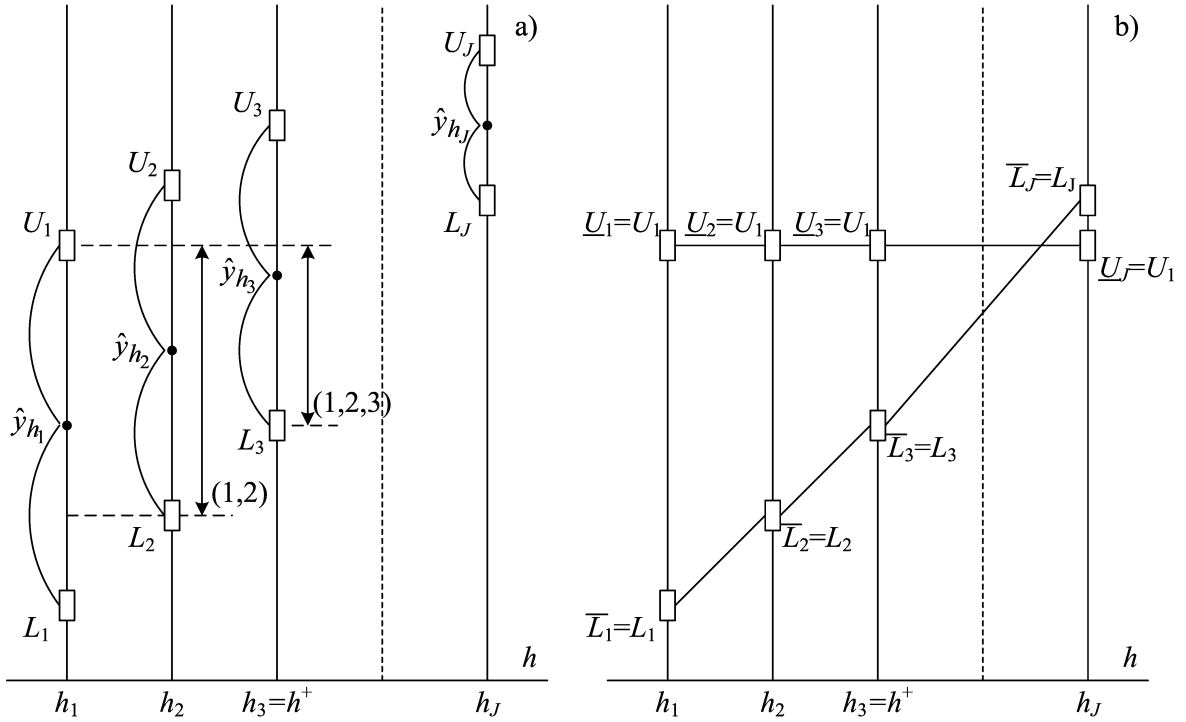
For $h_j \in H$ we define a sequence of the confidence intervals $Q_j = Q(h_j)$

$$Q_j = [\hat{y}_{h_j}(x) - \Gamma \cdot \sigma_{\hat{y}_h}(x, h_j),\ \hat{y}_{h_j}(x) + \Gamma \cdot \sigma_{\hat{y}_h}(x, h_j)].$$

and $y(x) \in Q_j$ for all h_j provided that $h_j < h^*(x)$.

The *ICI* rule

Consider the intersection of the intervals $Q_j,\ 1 \le j \le i,$ with increasing i, and let i_+ be the largest of those i for which the intervals $Q_j,\ 1 \le j \le i,$ have a point in common. This i_+ defines the adaptive scale and the adaptive LPA estimate as follows $\hat{y}^+(x) = \hat{y}_{h^+(x)}(x),\ h^+(x) = h_{i_+}$.



ICI rule calculations

Determine the sequence of the upper and lower bounds of the confidence intervals Q_i

$$Q_i = [L_i,\ U_i],$$
$$U_i = \hat{y}_{h_i}(x) + \Gamma \cdot \sigma_{\hat{y}_h}(x, h_i),$$
$$L_i = \hat{y}_{h_i}(x) - \Gamma \cdot \sigma_{\hat{y}_h}(x, h_i).$$

Let

$$\bar{L}_{i+1} = \max\{\bar{L}_i,\ L_{i+1}\},\ \underline{U}_{i+1} = \min\{\underline{U}_i,\ U_{i+1}\},$$
$$i = 1, 2, \ldots, J-1,\ \bar{L}_1 = L_1,\ \underline{U}_1 = U_1.$$

According to these formulas $\bar{L}_{i+1}$ and $\underline{U}_{i+1}$ are nondecreasing and nonincreasing sequences, respectively.

Find the largest i when

$$\bar{L}_i \le \underline{U}_i,\ i = 1,2,\ldots,J.$$

Denote this largest value i_+. This i_+ is the largest of those i for which the confidence intervals Q_i have a point in common as it is discussed above and the ICI adaptive scale is $h^+ = h_{i_+}$.

$LPA - ICI$ algorithm

Input variables of the ICI algorithm for signal estimation:

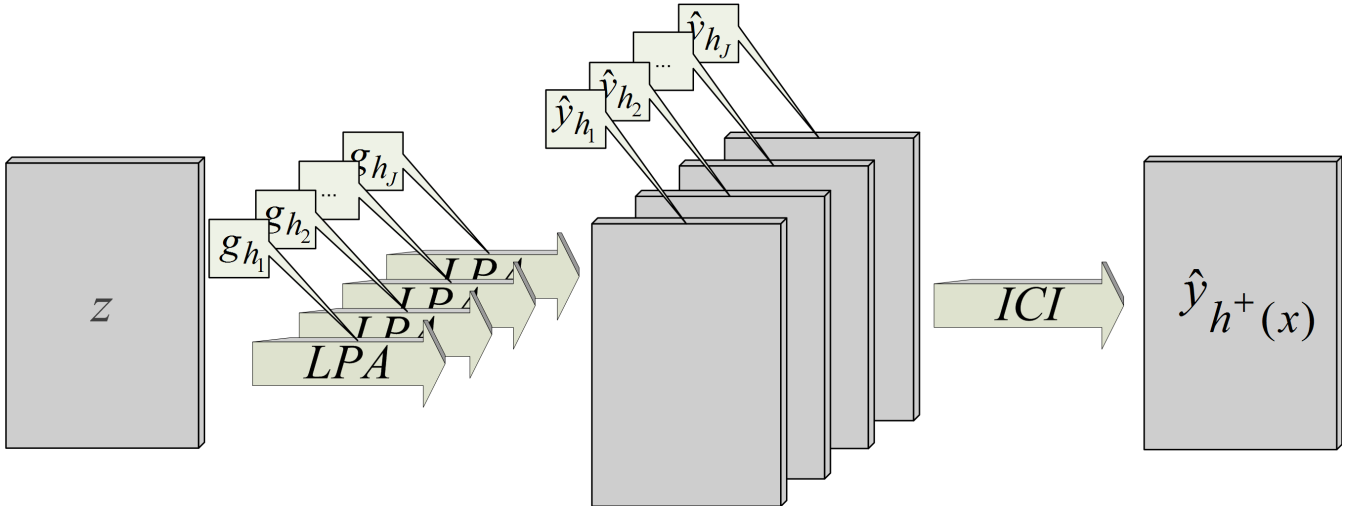
1. Noise variance σ^2;
2. The estimate $\hat{y}_h(x)$ given for all $h \in H$ and $x \in X$;
3. The variance $\sigma^2_{\hat{y}_h}(x,h)$ of the estimates given for all $h \in H$.

Algorithm organization

The algorithm includes the following basic steps:

1. The LPA filter design defines the bank of the linear filters $g_h(x)$, $h \in H$;
2. Calculation of the variance of observation noise;
3. Filtering an input signal with the results given for all $h \in H, x \in X$;
4. Calculation of the lower and upper bounds of the confidence intervals.
5. Selection of the ICI adaptive scales $h^+(x)$ and the corresponding adaptive estimates $\hat{y}_{h_+}$ for all $x \in X$.

A layout of the adaptive scale $LPA - ICI$ algorithm.



Differentiation

The above algorithm is applicable for derivative estimation if the kernels g_h are replaced by the corresponding differentiating kernel $g_h^{(r)}$.

Complexity

The calculation of the estimate $\hat{y}_h(x)$ for a given h is a linear convolution requiring $N_{conv} \sim n \log n$ operations. If more sophisticated procedures are used for convolution, then $N_{conv} \sim n \log n_{h_J}$, where n_{h_J} is the maximum size of the support of the kernel g_{h_J}.

The ICI algorithm is implemented as a loop on J different scales from the set H. Its complexity is proportional to J. Thus, the overall complexity of calculations of the ICI adaptive estimates is proportional to $J \cdot N_{conv}$.

In this way the ICI adaptive scale algorithm belongs to the class of fast algorithms with a number of operations proportional to $n \log n$ or $n \log n_{h_J}$.

Convergence

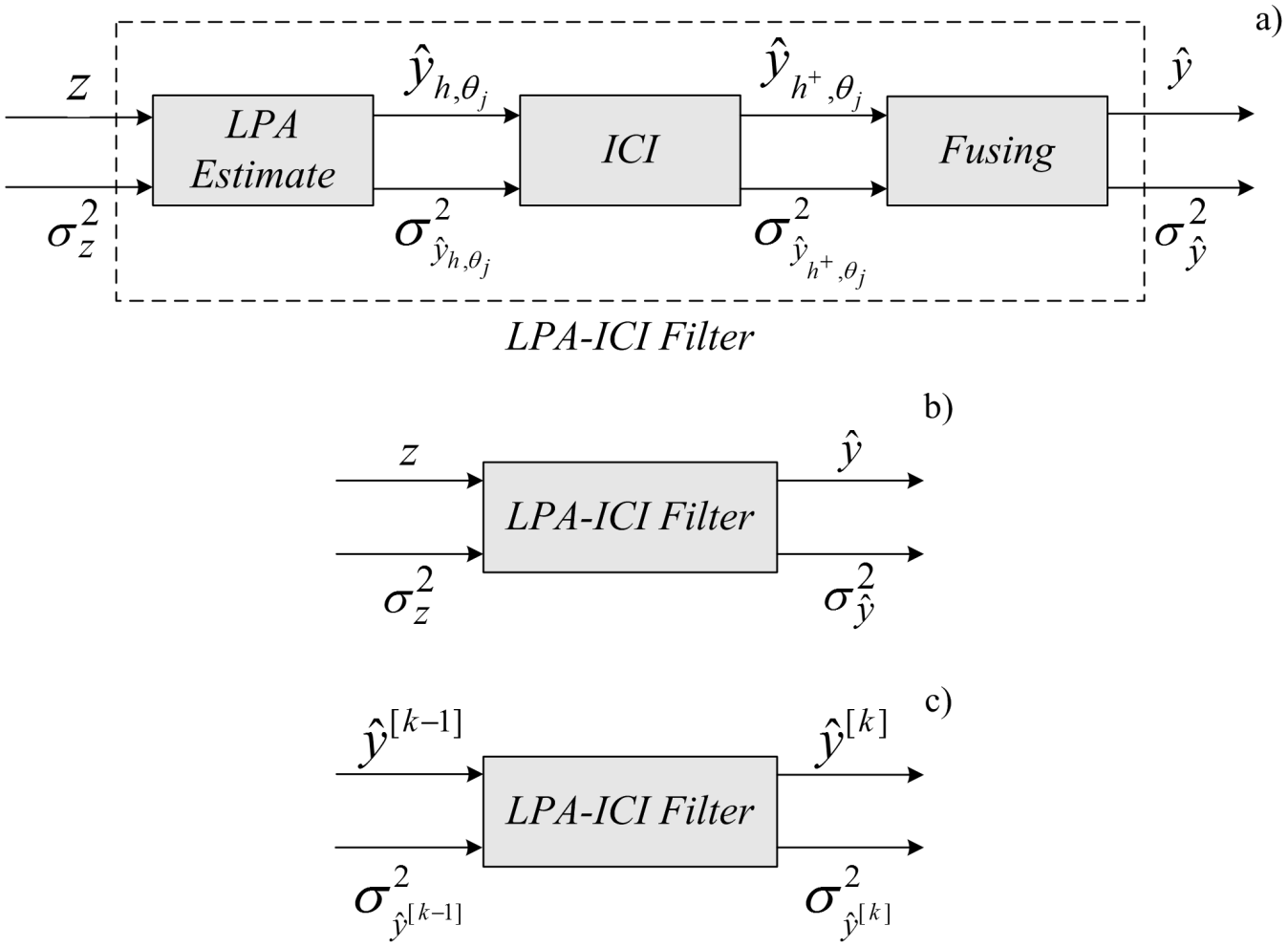
It is assumed in this asymptotic that the signal is defined on the regular grid with a small enough sampling interval Δ and with a large number of observations $n \sim 1/\Delta^d$. It is assumed also that the threshold Γ is growing as $\log n$. The set of scales H is dense enough starting with a small $h_1 = o(\log n/n)$ and $h_i - h_{i-1} = o(\log n/n)$.

Proposition. Let the assumptions of the accuracy proposition hold, the estimates $\hat{y}_{h^+}(x)$ and $\hat{y}^{(r)}_{h^+}(x)$ be scale adaptive according to the ICI algorithm. Then for n large enough and the threshold Γ growing proportionally to $\log n$ the mean square errors of the estimates are of the following orders:

$$E\{(y(x) - \hat{y}_{h^+}(x))^2\} = O((\frac{\log n}{n})^{\frac{2(m+1)}{2m+d+2}}),$$

$$E\{(y^{(r)}(x) - \hat{y}^{(r)}_{h^+}(x))^2\} = O((\frac{\log n}{n})^{\frac{2(m+1-|r|)}{2m+2+d}}).$$

Image Denoising



Estimate aggregation/fusing

Using K sectors we obtain K independently derived estimates for each x. These estimates are exploited to obtain the unique final estimate $\hat{y}(x)$ from the partial directional ones.

We use two different methods for calculation of the final estimate:

(1) Multi – window (fused) *estimate* are weighted means with the data-driven adaptive weights.

With the inverse variances as the weights for the linear fusing it gives

$$\hat{y}(x) = \sum_{j=1}^{K} \lambda_j \hat{y}_{\theta_j}(x),\ \hat{y}_{\theta_j}(x) = \hat{y}_{h,\theta_j}(x)|_{h=h^+(x,\theta_j)},$$

$$\lambda_j = \sigma_j^{-2}(x) / \sum_{i=1}^{K} \sigma_i^{-2}(x),$$

where

$$\hat{y}_{h,\theta_j}(x) = (g_{h,\theta_j} \circledast z)(x),\ \sigma_j^2(x) = \sigma^2_{\hat{y},\theta_j}(x,h)|_{h=h^+(x,\theta_j)},$$

and the variance of the adaptive scale estimate is calculated as

$$\sigma^2_{\hat{y},\theta_j}(x,h) = \sigma^2 \sum_x g^2_{h,\theta_j}(x).$$

The weights λ_j are data-driven adaptive as $\sigma^2_j(x)$ depend on the adaptive pointwise $h^+(x,\theta_j)$.

Assuming that the supports of the kernels $g_{\theta_j,h}$ are not overlapping and neglecting that the kernels have the point x as a common one we obtain for the variance of the fused estimate

$$\sigma^2_{\hat{y}}(x) = \sum\nolimits_{j=1}^{K} \lambda_j^2 \sigma_j^2(x) = \frac{1}{\sum_{j=1}^{K} \sigma_j^{-2}(x)}.$$

(2) $Combined - window\ estimate.$

Let h_j^+ be an adaptive scale found for $j-th$ estimate. Denote the corresponding window function by w_{j,h_j^+} and its support by $supp(w_{j,h_j^+})$.

The support of the combined – window estimate is a union of these adaptive supports

$$U(x) = \cup_{j \in I} supp(w_{j,h_j^+}).$$

The LPA technique is used in order to calculate the corresponding combined-window estimate in the support $U(x)$.

For $m = 0$ the combined-window estimate is the mean of all observations covered by the supports $supp(w_{j,h_j^+})$.

Recursive LPA_ICI algorithms (demo_RecursiveDenoisingGaussian.m)

The above calculations can be exploited in standard or recursive modes.

Denoting the calculations imbedded in this algorithm as a LI operator the input-output equation can be written as

$$(\hat{y},\, \sigma_{\hat{y}}) = LI\{z, \sigma_z\}.$$

In the recursive mode the $LPA-ICI$ multidirectional algorithm is applied repeatedly to the data obtained as outputs of the previous filtering:

$$(\hat{y}^{[k]},\, \sigma_{\hat{y}}^{[k]}) = LI\{z^{[k-1]}, \sigma_z^{[k-1]}\}, \quad k = 1, 2, \dots \ ,$$

where the index $^{[k]}$ numbers the iterations.

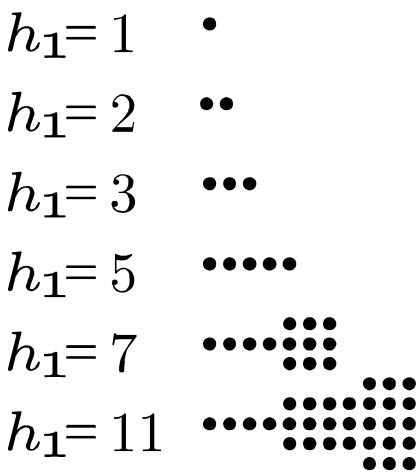
Simulation results (demo_DenoisingGaussian.m)

A test signal is the 256×256 "Cameraman" image (8 bit gray-scale) corrupted by an additive zero-mean Gaussian noise. The image intensity takes integer values $0 \leq y \leq 255$ with the noise standard deviation $\sigma = 25.5$ ($SNR = 14.39\ dB$).

We use narrow nonoverlapping sectorial kernels of the order $m = 0$ with a uniform window function w.

The kernel supports are defined by the two-dimensional scale $h = (h_1, h_2)$ with $h \in H$

$$H = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 5 \\ 1 \end{pmatrix}, \begin{pmatrix} 7 \\ 2 \end{pmatrix}, \begin{pmatrix} 11 \\ 3 \end{pmatrix} \right\}.$$



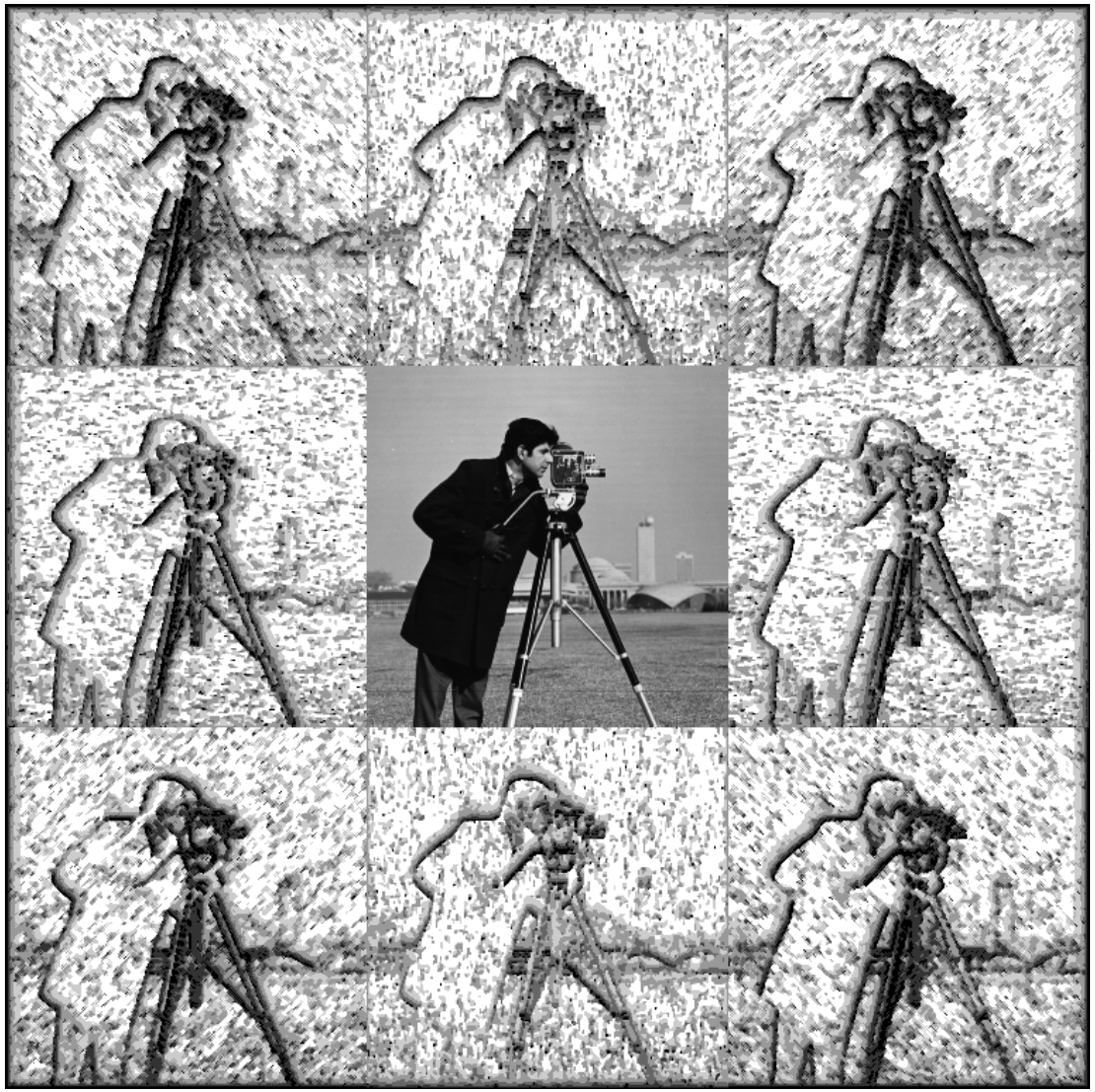
The estimates are calculated as the sample means of observations included in the kernel supports.

The ICI rule is used with the threshold $\Gamma = 1$.

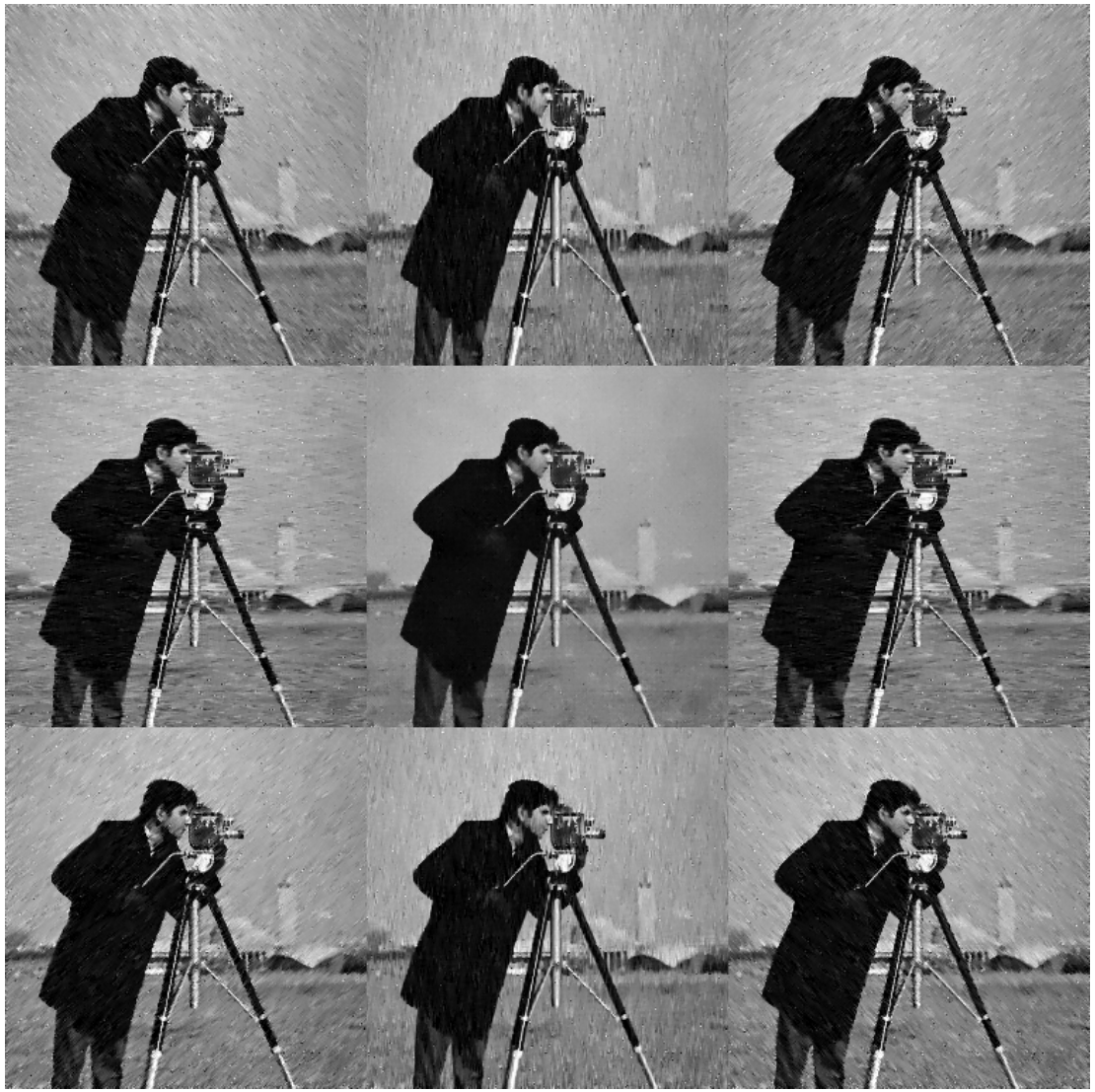
The estimates and the adaptive scales $h_j^+(x)$ are found for eight directions

$$\theta_j = (j-1)\pi/4,\ j = 1,\ldots,8.$$

These ICI adaptive directional estimates are fused in the final one using the multi-window method.



ICI adaptive directional scales $\hat{h}_j^+(x)$, $\theta_j = (j-1)\pi/4, j = 1,\ldots,8$.



$LPA-ICI$ adaptive scale directional estimates $\hat{y}_{j,\hat{h}_j^+(x)}(x)$.

Cameraman image: criteria values for the eight directional and final multi-window estimates.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	Final
ISNR, dB	4.13	3.57	4.08	3.55	4.11	3.44	4.07	3.55	8.07
SNR, dB	18.5	17.9	18.4	17.94	18.50	17.83	18.45	17.94	22.46
PSNR, dB	24.1	23.5	24.0	23.53	24.08	23.41	24.04	23.52	28.05
RMSE	15.8	16.9	15.9	16.98	15.93	17.20	16.01	16.98	10.10
MAE	10.6	1.54	10.7	11.59	10.68	11.70	10.81	11.57	6.66
MAXDIF	133.	114.	124.	117.0	112.6	142.4	114.4	125.8	85.26

Cameraman image: criteria values for the four quadrant and final multi-window estimates.

	Q_1	Q_2	Q_3	Q_4	Final Est
ISNR, dB	4.55	4.58	4.55	4.46	7.25
SNR, dB	18.94	18.97	118.94	18.85	21.64
PSNR, dB	24.52	24.55	24.52	24.44	27.22
RMSE	15.14	15.09	15.14	15.29	11.09
MAE	9.48	9.42	9.44	9.51	6.65
MAXDIF	132.99	120.59	119.71	112.25	98.93

ICI-pilot DCT algorithm

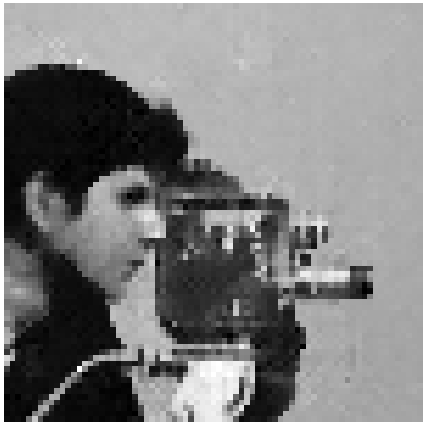
1. The square symmetric uniform window w and the zero order LPA define the estimator as a sample mean of the observations in this window.
2. The ICI is applied in order to find the corresponding adaptive pointwise window sizes $h^{+}(x)$.
3. The discrete cosine transform (DCT) is applied to filter observations in these adaptive varying size subsets. The filtering is produced by the standard hard-thresholding procedure.
4. The DCT filter gives the estimates for all points of the adaptive size subsets. As these subsets have different sizes and overlapping we obtain a various number of estimates for each x.
5. The final estimate for a given x is calculated as a mean of the DCT filtered observations available for each x.

Cameraman image: criterion values for adaptive ICI-pilot DCT, fixed size DCT, $LPA-ICI$ recursive and $LPA-ICI$ eight directional multi-window estimates.

	DCT	DCT	LPA – ICI	LPA – ICI
	ICI-pilot	(8×8)	(recursive)	(8 sectors)
ISNR, dB	8.60	8.03	8.47	8.07
SNR, dB	23.0	22.42	22.86	22.46
PSNR, dB	28.58	28.01	28.44	28.05
RMSE	9.49	10.14	9.65	10.09
MAE	6.49	6.55	6.12	6.44
MAXDIF	87.00	85.55	111.10	85.26

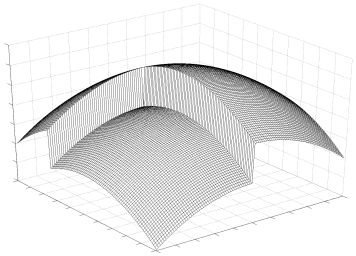
DCT (8×8)

ICI-pilot DCT

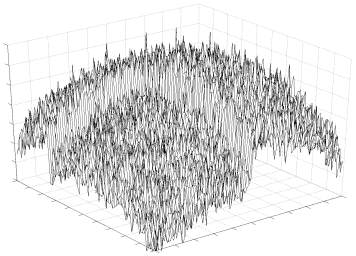


LPA – ICI

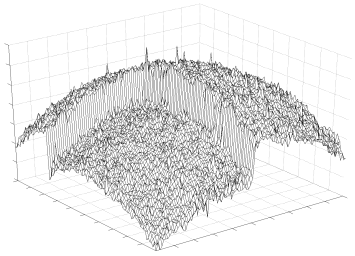
Recursive



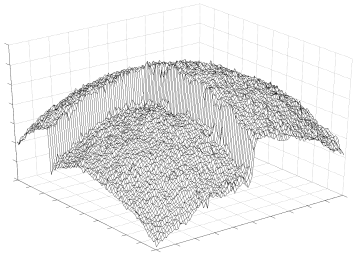
a) True function



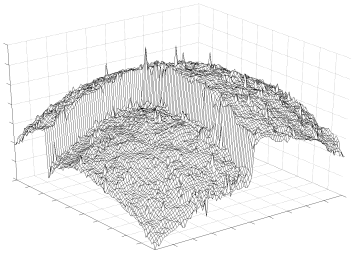
b) Noisy function



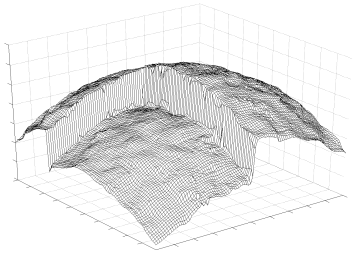
c) Multi-window



d) Combined window



e) Rec. multi-window

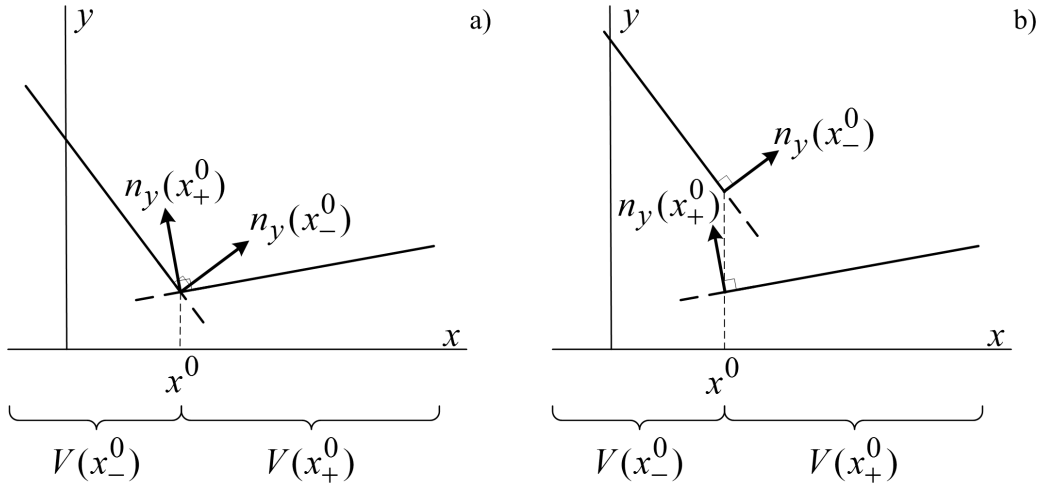


f) Rec. combined

Piece-wise quadratic function: data and estimates.

Anisotropic Gradient

Idea



A signal $y(x)$ is piece-wise linear composed from two linear fragments. There are two gradients and two normal vectors for left- and right-hand side neighborhoods of the point x: $a)$ continuous signal, $b)$ discontinuous signal.

Definition

A signal y is differentiable at x if there exists a vector $\nabla y(x)$ such that

$$y(x+v)-y(x)-v^T\nabla y(x)=o(|v|),\ v\in \mathrm{B}_h,$$

where ∇y is the gradient vector of the signal y and v is a vector-deviation from the point x.

The anisotropic gradient concept allows the existence of a few neighborhoods V_l at the pixel x with the corresponding a few possible different vectors $(\nabla y(x))_l$ such that

$$y(x+v)-y(x)-v^T(\nabla y(x))_l=o(|v|),\ v\in V_l$$

The ICI adaptive anisotropic differentiation is aimed to estimate simultaneously both the gradients $(\nabla y(x))_l$ and the neighborhoods V_l.

Directional estimate of the anisotropic gradient

Consider a sectorial partition of the spherical neighborhood (unit ball) of the point x:

$$\mathrm{B}_h=\{v:\|x-v\|\le h\}.$$

Let $g_{h,\theta}^{(1,0)}$ be differentiating kernels defined on these sectors with the

corresponding estimates $\hat{y}^{(1,0)}_{h,\theta}$.

Assuming that the signal y is differentiable at x with a vector-gradient

$$\nabla y = (\partial_{x_1} y, \partial_{x_2} y)^T$$

the directional derivative (on the definition $\partial_\theta y \triangleq \partial_{u_1} y$) can be written as

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

and can be treated as an indirect observation of the gradient components $\partial_{x_1} y$ and $\partial_{x_2} y$.

The estimate of $\partial_\theta y$ is the directional derivative $\hat{y}^{(1,0)}_{h,\theta_i}$, then the gradient estimate ∇y can be found by minimizing the weighted mean square criterion

$$J = \sum_{i=1}^{K} \frac{1}{\sigma^2_{\theta_i}} [\hat{y}^{(1,0)}_{h,\theta_i} - (\partial_{x_1} y \cdot \cos\theta_i + \partial_{x_2} y \cdot \sin\theta_i)]^2,$$

where the weights $\sigma^2_{\theta_i}$ are the variances of the estimates $\hat{y}^{(1,0)}_{h,\theta_i}$.

In the vector-matrix notation J can be rewritten as

$$J = (\hat{y}_h^{(1,0)} - B\nabla y)^T \Lambda (\hat{y}_h^{(1,0)} - B\nabla y),$$

where $\hat{y}_h^{(1,0)} = (\hat{y}_{h,\theta_1}^{(1,0)}, \ldots, \hat{y}_{h,\theta_K}^{(1,0)})^T$ is a vector of the estimates, $\Lambda = diag\{1/\sigma_{\theta_1}^2, \ldots, 1/\sigma_{\theta_K}^2\}$ is a diagonal matrix, and $B = (B_i)_{K\times 2}$, $B_i = (\cos\theta_i, \sin\theta_i)$.

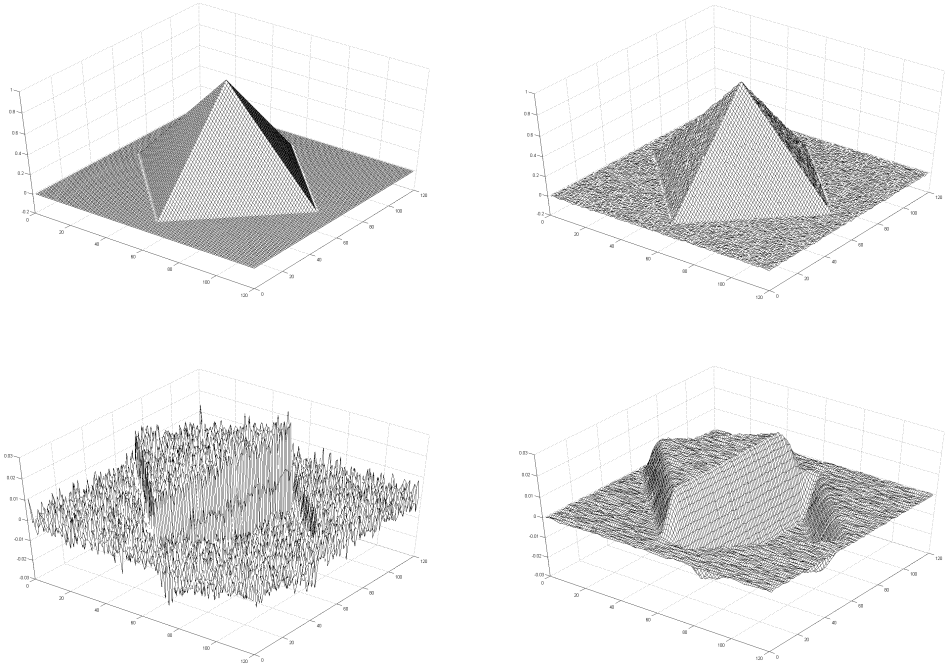
Minimization of J gives the estimate of the gradient as

$$\hat{\nabla} y = (B^T \Lambda B)^{-1} B^T \Lambda \hat{\mathbf{y}}_h^{(1,0)}.$$

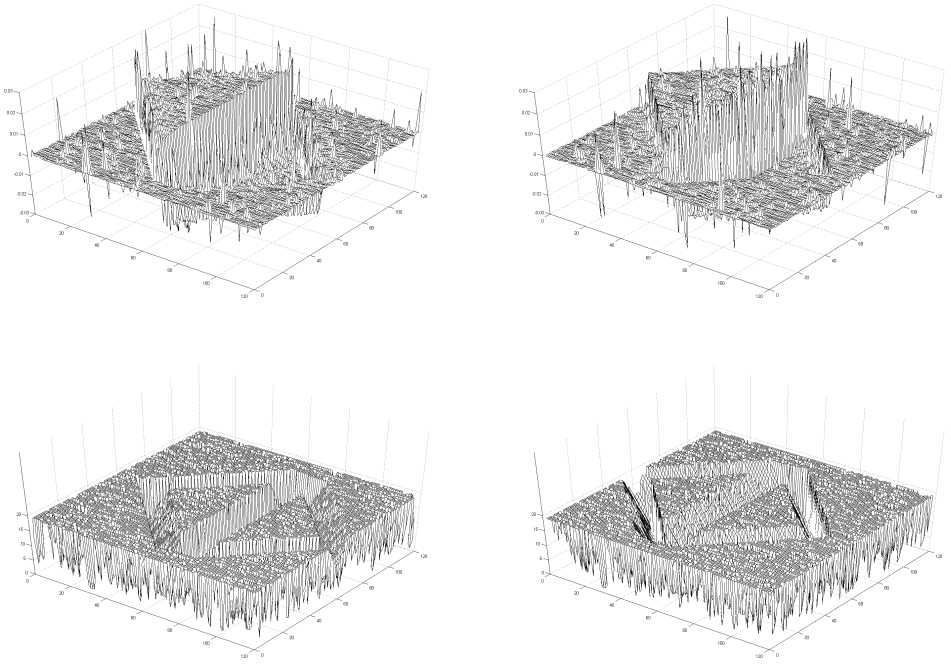
The union of the support of these adaptive estimates is a star-shaped set which allows the following interesting interpretation. It is a largest neighborhood of x where the signal y is differentiable.

Illustrations

Horizontal derivative for pyramid

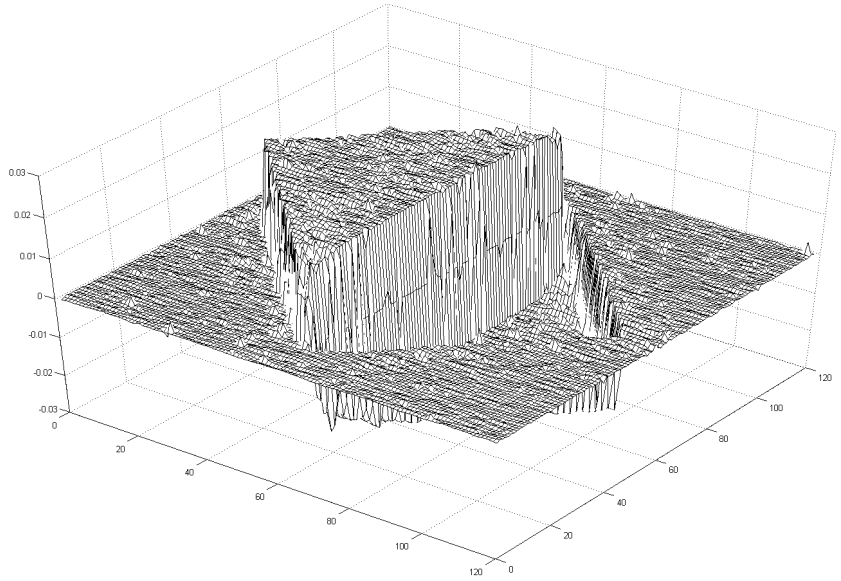


Non-adaptive derivative estimation. A short differentiating kernel of the length 3 gives is an extremely noisy result (bottom left). A larger kernel of the length 11 reduces the noise but blurs the sharp transitions in the derivative(bottom right).



Adaptive nonsymmetric derivative estimation: left and right derivatives (first row); the corresponding ICI adaptive scales h^+ are shown in the second row.

We apply the described above anisotropic procedure for the derivative calculation using the left- and right-hand side adaptive directional derivative estimates.



The estimate of the derivative ∂_{x_1} obtained using the ICI adaptive nonsymmetric left- and right- differentiating kernels, $\theta_1 = 0$, $\theta_2 = \pi$.

Optical flow estimation

A relative motion of objects and camera is an obvious source of temporal variations in recorded image sequences.

The image intensity can be considered as a function of the spatial x and time t variables with notation $y(x,t)$. For a sequence of time-instants $\{t_j\}$ we have a sequence of the intensities $\{y(x,t_j)\}$ where for each pixel x the signal $y(x,t_j)$ gives a variation of the intensity in time:

$$\{y(x,t_j),\ x \in X \text{ and } j = 1,\ldots,K\},$$

where K is a size of the sequence and X is a grid of the image pixels.

Optical flow equation

Let s be a parameter along a motion trajectory then the invariance of y along the trajectory means that

$$\partial_s y = 0.$$

Using the chain rule for differentiation we obtain

$$\partial_s y = (\partial_x y)^T \cdot \partial_s x + \partial_t y \cdot \partial_s t = 0.$$

and

$$(\partial_x y)^T \cdot \partial_t x + \partial_t y = 0.$$

Then the optical flow equation can be given in the standard form

$$\boxed{v^T \partial_x y + \partial_t y = 0.}$$

If the gradient $\partial_x y$ and time derivative $\partial_t y$ are given then the only unknown in the flow-equation are the component of the vector-velocity v.

Velocity estimation

Let u_1 and u_2 be variables defined in the rotated coordinates. The link between the unmoving and rotated variables is defined by the formula

$$u = U(\theta)x,\ x = U^T(\theta)u, U(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

Using the chain rule for the vector differentiation we obtain for the gradient of $y_\theta(u)$ calculated on u

$$\partial_u y_\theta(u) = U(\theta)\partial_x y(x)$$

and

$$\partial_x y(x) = U^T(\theta)\partial_u y_\theta(u).$$

The derivatives in the different directions are linked through the rotation matrix $U(\theta)$.

Inserting the gradient $\nabla y(x) \triangleq \partial_x y(x)$ we obtain the optical flow equation in the directional form

$$\boxed{(U(\theta)v)^T \partial_u y_\theta(u) + \partial_t y = 0,}$$

where $U(\theta)v$ is a projection of the vector-velocity v on the axes u_1, u_2.

The directional derivative estimates $\hat{y}^{(r)}_{h,\theta}(x)$ for $r = (1,0)$ and $r = (0,1)$ gives the estimates of the components of the vector $\partial_u y_\theta$, i.e.

$$\partial_{u_1} y_\theta \simeq \hat{y}_{h,\theta}^{(1,0)}(x),\ \partial_{u_2} y_\theta \simeq \hat{y}_{h,\theta}^{(0,1)}(x).$$

Denote this vector estimate as

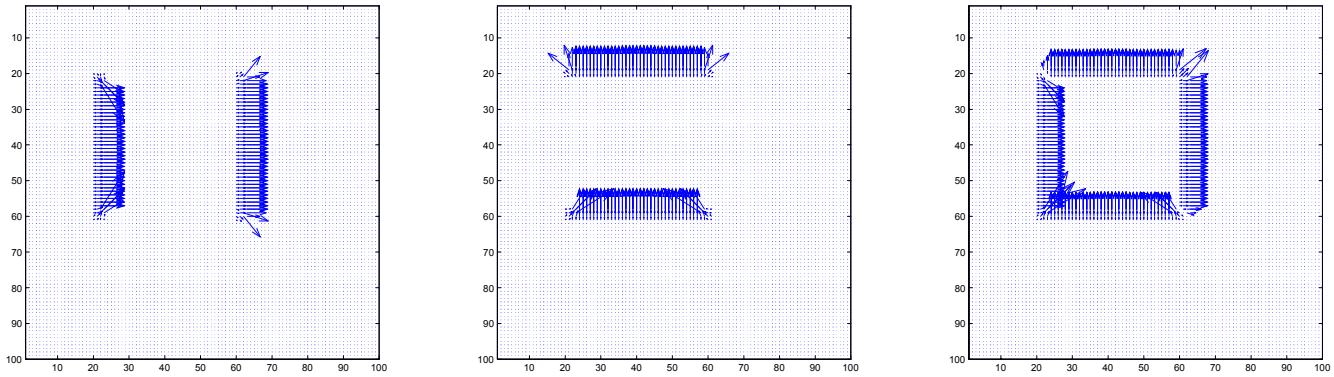
$$\boxed{\partial_u \hat{y}_{h,\theta}(x) = (\hat{y}_{h,\theta}^{(1,0)}(x),\ \hat{y}_{h,\theta}^{(0,1)}(x))^T.}$$

We can find the estimate of the vector-velocity by minimizing the quadratic criterion

$$\boxed{J = \sum_{\theta_i} [(\partial_u \hat{y}_{h,\theta_i}(x))^T U(\theta_i) v + \partial_t y]^2.}$$

These directional derivative estimates can be exploited with the ICI adaptive scales.

Illustrations



"Square": optical flow for horizontal, vertical and diagonal movements.

Akyoi: optical flow for head and body movement.

Image Deblurring

Modeling and inverse

We wish to reconstruct an image $2D$ intensity function y, but we are able to observe only the convolution $(y \circledast v)(x)$, where v is a given PSF.

It is assumed that the convolution is discrete and noisy, so we observe z given as

$$z(x) = (y \circledast v)(x) + \varepsilon(x),$$

where x is defined on the regular $n_1 \times n_2$ lattice X and ε is zero mean white Gaussian noise with variance σ^2.

In the frequency domain

$$Z(f) = Y(f)V(f) + \varepsilon(f),$$

where $Z(f), Y(f), V(f)$ and $\varepsilon(f)$ are the DFT of the corresponding signals and $f = (f_1, f_2), f \in F$.

A unbiased estimate of $Y(f)$ can be obtained as a straightforward "naive inverse" solution of the equation

$$\hat{Y}(f) = \frac{Z(f)}{V(f)}.$$

However, in the cases of interest, the convolution operator v is not invertible, in the sense that the inverse of v does not exist as a bounded linear operator. Such inverse problems are called ill – posed or ill – conditioned.

It is now a standard to approach the inverse problem by the method of regularization:

$$\hat{Y}(f) = \frac{Z(f)V^*(f)}{|V(f)|^2 + \alpha^2},$$

where $\alpha^2 > 0$ is a regularization parameter.

LPA deblurring

The basic idea of the proposed deblurring algorithm is to estimate the smoothed image intensity y_h instead of the original one y

$$y_h(x) = (g_h \circledast y)(x).$$

Applying the kernel operator g_h to the both sides of the observation equation yields -

$$z_h(x) = (g_h \circledast (y \circledast v))(x) + \varepsilon_h(x) = (y_h \circledast v)(x) + \varepsilon_h(x),$$

where

$$\varepsilon_h(x) = (g_h \circledast \varepsilon)(x).$$

In the frequency domain

$$Z_h(f) = Y_h(f)V(f) + \varepsilon_h(f),$$

where $Z_h(f),\ Y_h(f)$ and $\varepsilon_h(f)$ stand for DFT of the corresponding signals.

Regularized inverse

Let us look for a solution $\hat{y}_h$ for y_h based on the criterion

$$R_\psi^2 = R^2 + \alpha^2 \psi, \ \alpha > 0,$$

$$R^2 = \sum_x r^2(x), \ r(x) = z_h(x) - \hat{z}_h(x), \ \hat{z}_h(x) = (y_h \circledast v)(x),$$

$$\psi = \sum_x r_d^2(x), \ r_d(x) = (y_h \circledast d)(x).$$

The parameter α defines a weight of this penalty in the criterion.

Using the Parseval formula we have

$$R_\psi^2 = \frac{1}{n_1 n_2}\|Z_h(f) - V(f)Y_h(f)\|^2 + \frac{\alpha^2}{n_1 n_2}\|Y_h(f)D(f)\|^2,$$

where $D(f)$ is the DFT for $d(x)$.

The minimum conditions for R_ψ^2 give the solution in the frequency domain as

$$\boxed{\hat{Y}_h(f) = \frac{G_h(f)V^*(f)}{|V(f)|^2+\alpha^2}Z(f).}$$

Wiener inverse

In the Wiener filtering we are looking for the optimal linear estimate of the smoothed signal $y_h(x)$ in the form of the linear shift-invariant operator

$$\hat{y}_h(x) = (z \circledast q_{WI})(x),$$

with the kernel q_{WI} to be found.

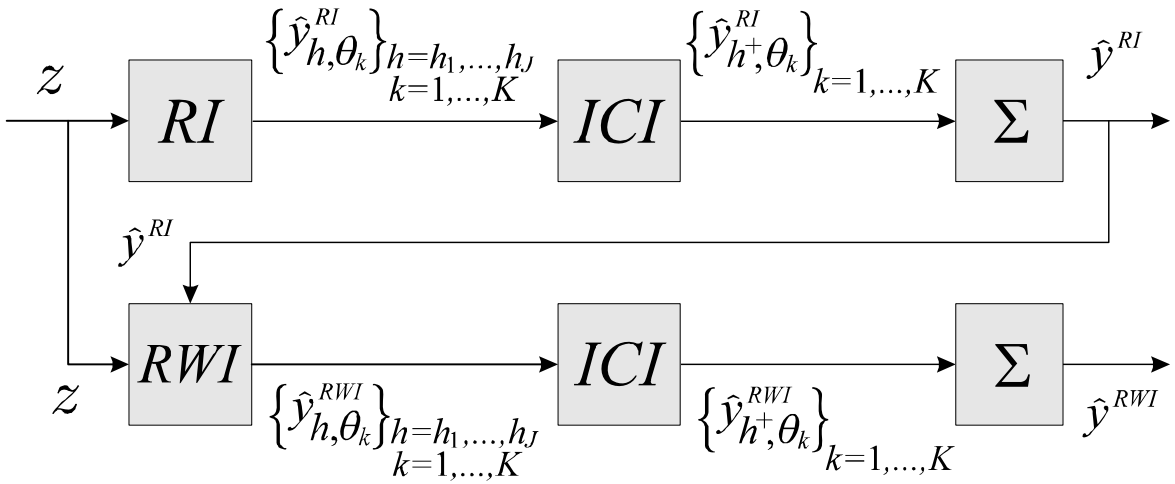
Hereafter, the index WI stays for "Wiener inverse". It gives a transfer function of the optimal Wiener filter in the form

$$Q_{WI}(f) = \frac{V^*(f)G_h(f)}{|V(f)|^2 + n_1 n_2 \sigma^2 / |Y(f)|^2}, f \in F.$$

The optimal estimate of y_h is as follows

$$\hat{Y}_h^{WI}(f) = Q_{WI}(f)Z(f).$$

LPA – ICI deblurring (demo_DeblurringGaussian.m)



In the first line of the flowchart the RI estimates are calculated for a set of scales h and directions θ, the ICI is used to obtain the pointwise adaptive scale directional estimates that are then fused into the $\hat{y}^{RI}$ 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 adaptive $\hat{y}^{RWI}$ estimate.

Performance study

The level of the noise in observations is characterized in dB by $BSNR$ (blurred SNR):

$$BSNR = 10 \log_{10} \frac{\sum_x ((v \circledast y)(x) - \frac{1}{n_1 n_2} \sum_x (v \circledast y)(x))^2}{\sum_x (z(x) - (v \circledast y)(x))^2}.$$

Table presents results for four experiments:

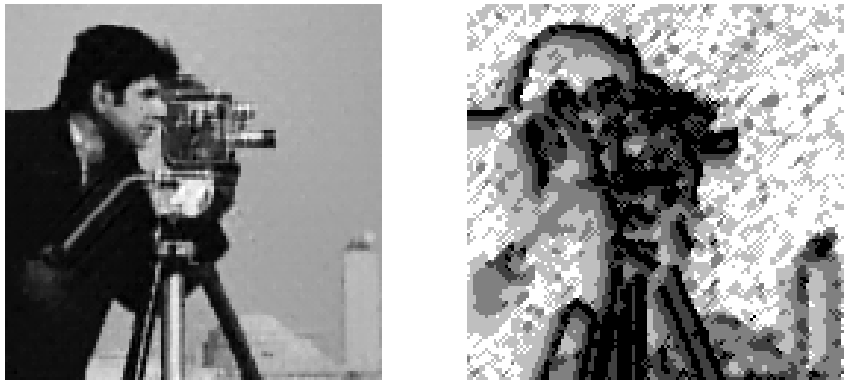
1. Cameraman image, 9×9 uniform (box-car) PSF, $BSNR = 40\ dB$ (Exp.1);
2. Cameraman image, $v(x_1, x_2) = (1 + x_1^2 + x_2^2)^{-1}, x_1, x_2 = -7, \ldots, 7$, $\sigma^2 = 2$ (Exp.2);
3. Cameraman image, $v(x_1, x_2) = (1 + x_1^2 + x_2^2)^{-1}, x_1, x_2 = -7, \ldots, 7$, $\sigma^2 = 8$ (Exp.3);
4. Lena image, v is a 5×5 separable filter with the weights $[1, 4, 6, 4, 1]/16$ in horizontal and vertical directions, $BSNR = 15.93\ dB$ (Exp.4).

The RWI algorithm demonstrates a good performance and outperforms the state-of-the-art techniques.

$ISNR$ (dB) of the $LPA-ICI$ algorithm for four experiments

		Cameraman	Cameraman	Cameraman	Lena
Method	\ Experiment	1	2	3	4
RWI		8.23	7.78	6.04	3.76
RI		7.70	7.18	5.30	–
GEM (Dias)		8.10	7.47	5.17	–
EM (Figueiredo and Nowak)		7.59	6.93	4.88	2.94
$ForWaRD$ (Neelamani et al.)		7.30	6.75	5.07	2.98

Original Cameraman image (left) and noisy blurred observations (Exp. 1) (right).

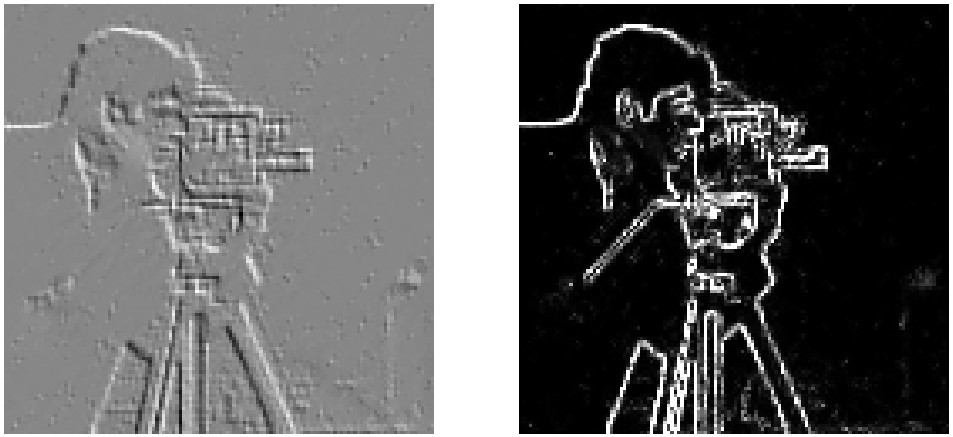


LPA – ICI algorithm performance: restored image ISNR=8.23 dB (left) and adaptive scales (right).

Differentiation

Replace in the RWI stage of the algorithm the smoothing kernels $g^{(0,0)}_{h,\theta_k}$ by the discrete differentiation kernels $g^{(1,0)}_{h,\theta_k}$ of the orders $m = (1,0)$. Then the output $\hat{y}^{RWI}_{h^+,\theta_k}$ of this two stage algorithm gives the estimate of the directional derivative $\partial_{\theta_k} y$.

For the edge detection we calculate the sum of the absolute values of these adaptive scale derivatives $\sum_{k=1}^{4} |\hat{y}^{RWI}_{h^+,\theta_k} - \hat{y}^{RWI}_{h^+,-\theta_k}|/2$.



Directional derivative (left $\theta = \pi/4$) and edge detection (right)

Motion deblurring

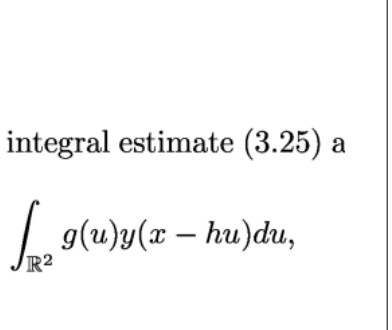
The model of the blur

$$v(x_1,x_2) = \begin{cases} 1/L, \text{ if } x_1 = x_2 \tan\beta,\ \sqrt{x_1^2 + x_2^2} \le L/2, \\ 0, \text{ otherwise,} \end{cases}$$

where the length $L = 41$.

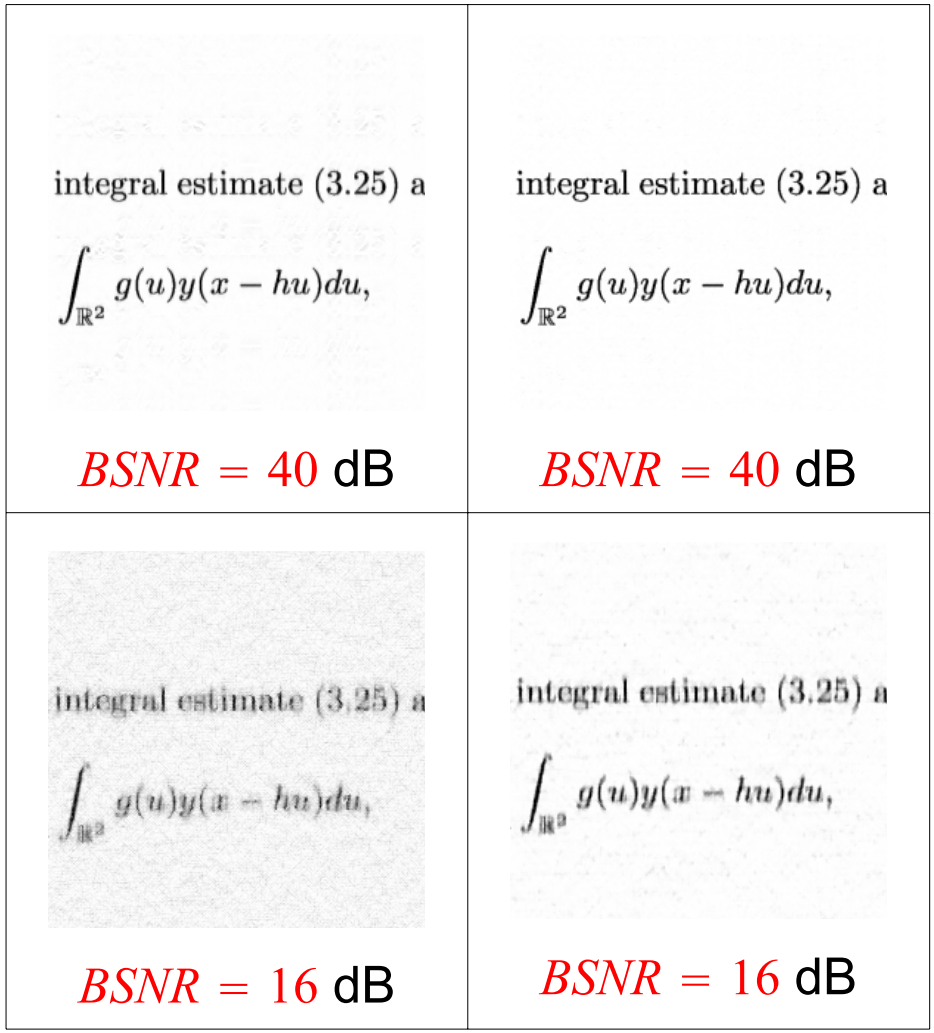
BSNR = 40 dB

BSNR = 16 dB

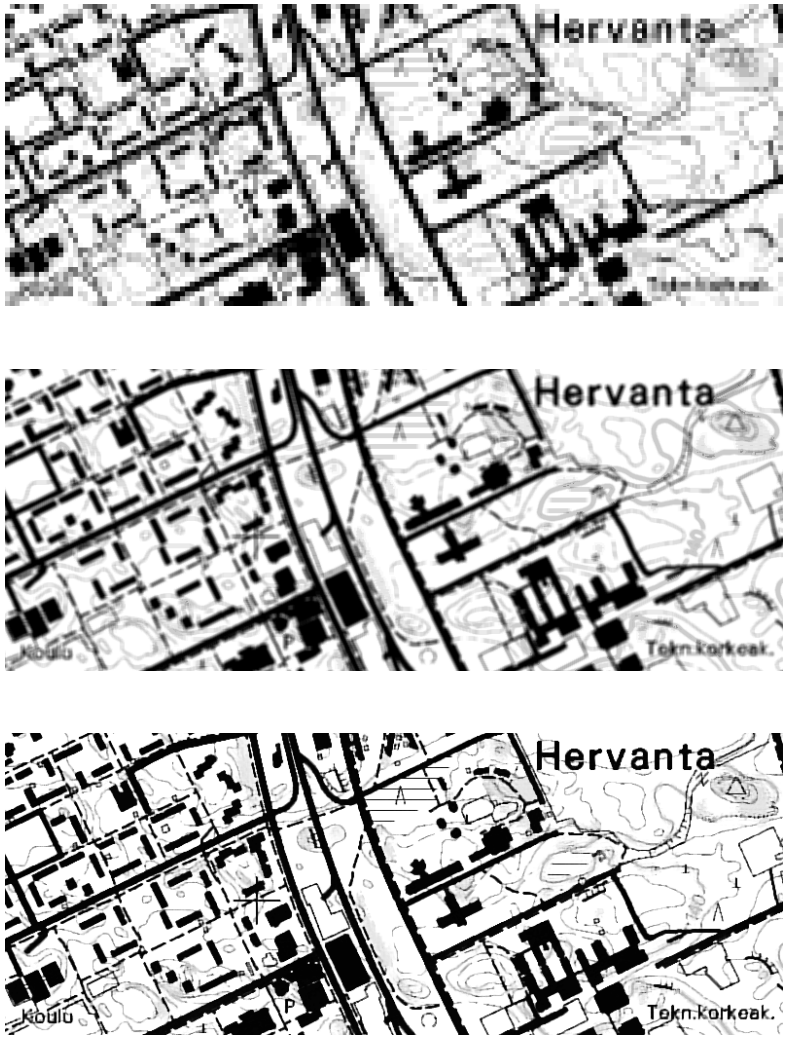


True text

True text and noisy images degraded by the vertical motion blur.



Reconstruction of the motion blurred images using the RI and RWI adaptive scale algorithms.



Super-resolution imaging

$3D$ Inverse Imaging

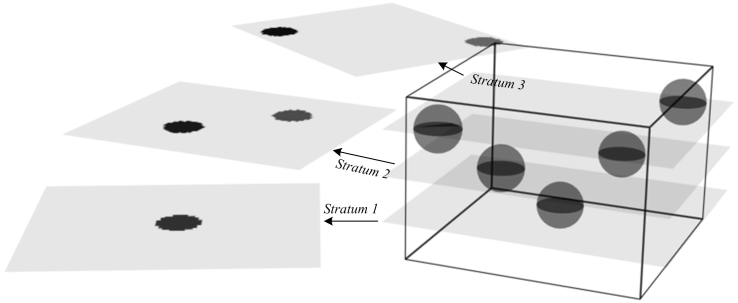
Sectioning microscopy

When one looks through the microscope, there is one plane that appears in focus. Nevertheless, these out-of-focus structures are in the field of view and thus obscure the in-focus plane.

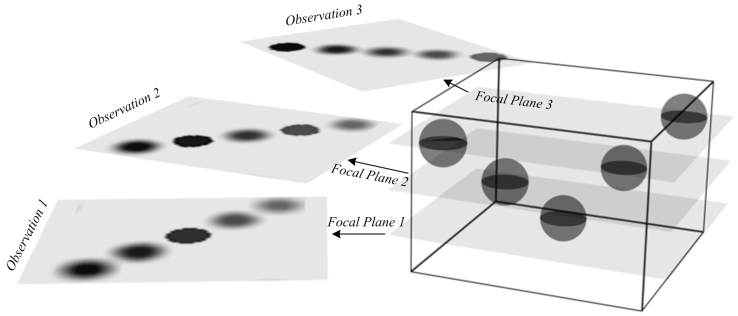
In order to obtain a $3D$ image from a specimen, it is common to use a method called optical sectioning. The microscope is focused at a given plane in the specimen and the image is recorded. Then, the microscope is refocused to another plane and another image is recorded.

Three-dimensional sectioning microscopy equipped with digital deblurring algorithms is a modern tool for visualization of specimens in biology, medicine, mineralogy, etc.

A phantom object of five balls. Three cross-sections (strata) of the object are shown.



A phantom object of five balls: optical sectioning. Three focal planes are observation slices.



Observation model

After discretization of the integral observation model we arrive to the following model given in the frequency domain

$$\mathbf{Z}(f) = \mathbf{V}(f)\mathbf{Y}(f) + \boldsymbol{\varepsilon}(f),$$

with $Z(f) = F\{z(\tilde{x})\}$, $Y(f) = F\{y(\tilde{x})\}$ and $\varepsilon(f) = F\{\varepsilon(\tilde{x})\}$ being the vectors of the respective sizes.

The PSF transfer matrix $V(f) = F\{v(\tilde{x})\}$ has a size $n_3 \times l$.

3D spatially adaptive inverse

Regularized inverse

The regularized inverse including LPA filtering is defined by the formula:

$$\hat{\mathbf{Y}}_h^{RI}(f) = \mathbf{Q}_{RI}(f,\alpha)\mathbf{Z}(f),$$

where $\hat{Y}_h^{RI}(f)$ is a vector of the regularized inverse estimates of $Y(f)$, and the transfer matrix of the regularized inverse filter is

$$\mathbf{Q}_{RI}(f,\alpha_1) = G_h(f)(\mathbf{V}^H(f)\mathbf{V}(f) + \alpha_1^2 I_{l\times l})^{-1}\mathbf{V}^H(f).$$

Here α_1^2 is a regularization parameter and $I_{l\times l}$ is the $l \times l$ identity matrix.

The scalar LPA filter $G_h(f)$ with the scale parameter h can be used independently for each component of the vector-estimate $\hat{Y}_h^{RI}(f)$.

Regularized Wiener inverse

The regularized Wiener inverse filter including LPA filtering is defined by the formula:

$$\hat{\mathbf{Y}}_h^{RWI}(f) = \mathbf{Q}_{RWI}(f, \alpha_2)\mathbf{Z}(f),$$

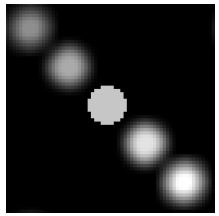
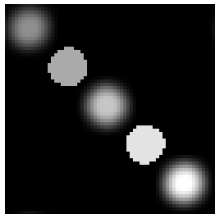
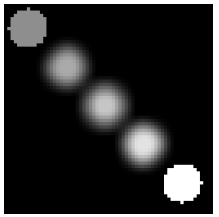
with the transfer matrix

$$\mathbf{Q}_{RWI}(f, \alpha_2) = G_h(f)\mathbf{Y}(f)\mathbf{Y}^H(f)\mathbf{V}^H(f) \times$$
$$(\mathbf{V}(f)\mathbf{Y}(f)\mathbf{Y}^H(f)\mathbf{V}^H(f) + n_1 n_2 \alpha_2^2 \boldsymbol{\sigma}^2)^{-1}, f \in F,$$

where α_2^2 is a regularization parameter and σ^2 is a diagonal matrix of the noise variances.

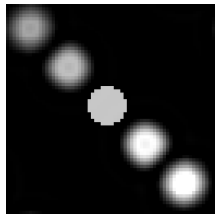
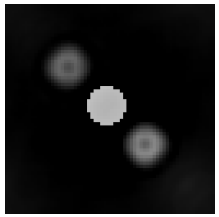
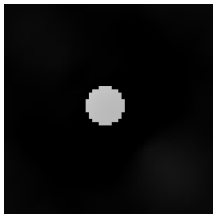
Illustrations (demo_3DInverse.m)

Observations

Reconstructions

Plane 1 Plane 2 Plane 3

Using 1 Using 1 and 2 Using 1, 2, 3

The mixed noisy observations of transparent Cameraman and Lena images obtained by focussing on different planes between these images.

The reconstruction of Lena and Cameraman images.

Multiframe Blind Deblurring

Image processing from multiple frames (channels) aim for enhanced comprehensive quality.

The basic estimation techniques involve two important concepts: channel model and inverse problem. When the blur is unknown, the image restoration is called a blind inverse problem and for multiple frames it is a multiple frame (multiple channels) blind inverse problem.

Our technique is based on the frequency domain representation of the observation model as well as of the basic algorithm implementation.

We apply the ICI rule for the blind multichannel deconvolution and show that it is able to do a good job as filter as well as the regularizator.

Observation model

The input to this system is an unknown image $y(x), x \in X$, with a finite discrete support of the size $n_1 \times n_2$. This image is distorted by unknown finite impulse response blurs modelled by the point spread functions (PSF) $v_j, j = 1, \ldots, L$:

$$z_j(x) = (y \circledast v_j)(x) + \sigma_j \eta_j(x),\ j = 1, \ldots, L.$$

The problem is to reconstruct both the image y and the $PSFs$ v_j from the observations $\{z_j(x), x \in X, j = 1, \ldots, L\}$.

Projection gradient algorithm

We exploit the following basic quadratic criterion

$$J \cdot n_1 n_2 = \sum_j \frac{1}{\sigma_j^2} \sum_{f \in F} |Z_j - YV_j|^2 +$$

$$\lambda_1 \sum_{j,i=1}^{L} \sum_{f \in F} d_{ij}^2 |Z_i V_j - Z_j V_i|^2 + \lambda_2 \sum_{f \in F} |Y|^2 + \lambda_3 \sum_{f \in F} |V_j|^2.$$

The estimates of the signal and the $PSFs$ are solutions of the following problem

$$(\hat{y}, \hat{v}) = \arg \min_{y \in Q_y, v_j \in Q_{v_j}} J,$$

where the admissible sets Q_y for y and Q_{v_j} for v_j are defined as follows

$$Q_y = \{y : 0 \le y \le 1\}$$

$$Q_{v_j} = \{v_j : \sum_x v_j(x) = 1\}.$$

The recursive projection gradient algorithm is used for solution

$$Y^{(k)} = P_{Q_y}\{Y^{(k-1)} - \alpha_k \partial_{Y^*} J(Y^{(k-1)}, V^{(k-1)})\},$$

$$V_j^{(k)} = P_{Q_y}\{V_j^{(k-1)} - \alpha_k \partial_{V_j^*} J(Y^{(k)}, V^{(k-1)})\},$$

$$k = 1, \ldots.$$

where some initiation $(Y^{(0)}, V_j^{(0)})$ is assumed.

Blind deconvolution algorithm

Main procedure

(1) Initialization: the initial guess for the $v_j^{(0)}$ and $y^{(0)}$. We use the Gaussian density for $v_j^{(0)}$ and the mean of the observed images for y with $y^{(0)} = \sum_{j=1}^{L} z_j(x)/L$.

Given $v_j^{(0)}$ and $y^{(0)}$: iterating $k = 1, \ldots$.

(2) Image estimation:

(2a) According to the projection algorithm

$$Y^{(k)} = (1 - \alpha_k) Y^{(k-1)} + \alpha_k \partial_{Y^*} J(Y^{(k-1)}, V^{(k-1)}),$$

(2b) Filter $y^{(k)}$ using $LPA - ICI$ filter

$$y^{(k)} \triangleq LI\{y^{(k)}, \sigma_{y^{(k)}}\}.$$

(2c) Project $y^{(k)}$ on the segment $[0, 1]$:

$$y^{(k)} \triangleq \max(0, \min(1, y^{(k)})).$$

(3) PSF estimation:

(3a) According to the projection algorithm

$$V_j^{(k)} = (1-\alpha_k) V_j^{(k-1)} + \alpha_k \partial_{V_j^*} J(Y^{(k)}, V^{(k-1)}).$$

(3b) Filter $V_j^{(k)}$ by the $LPA-ICI$ algorithm

$$v_j^{(k)} \triangleq LI\left\{v_j^{(k)}, \sigma_{v_j^{(k)}}\right\}, j = 1, \ldots, L.$$

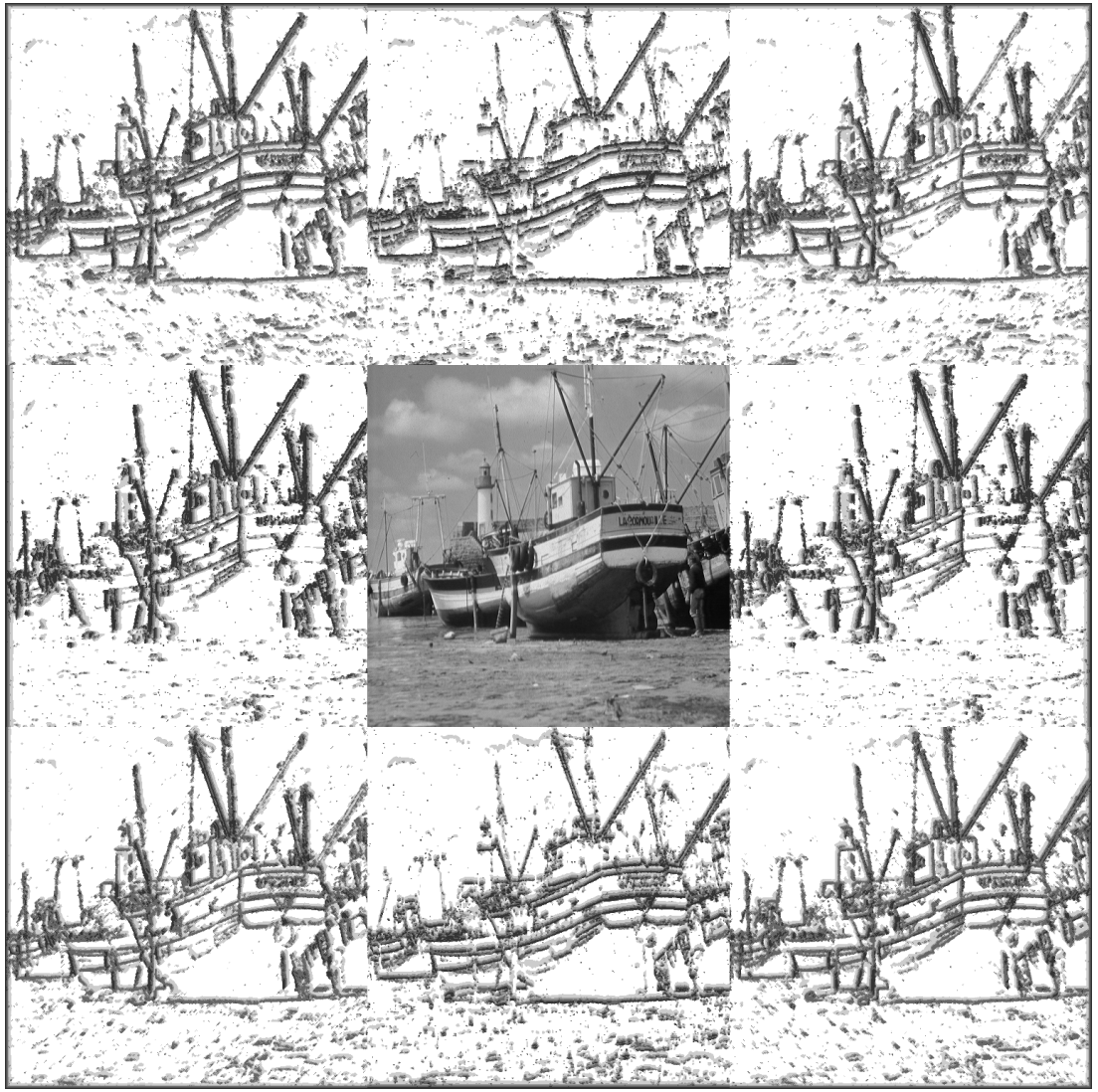
(3c) Normalize the estimates:

$$V_j^{(k)} \triangleq V_j^{(k)} / V_j^{(k)}(0).$$

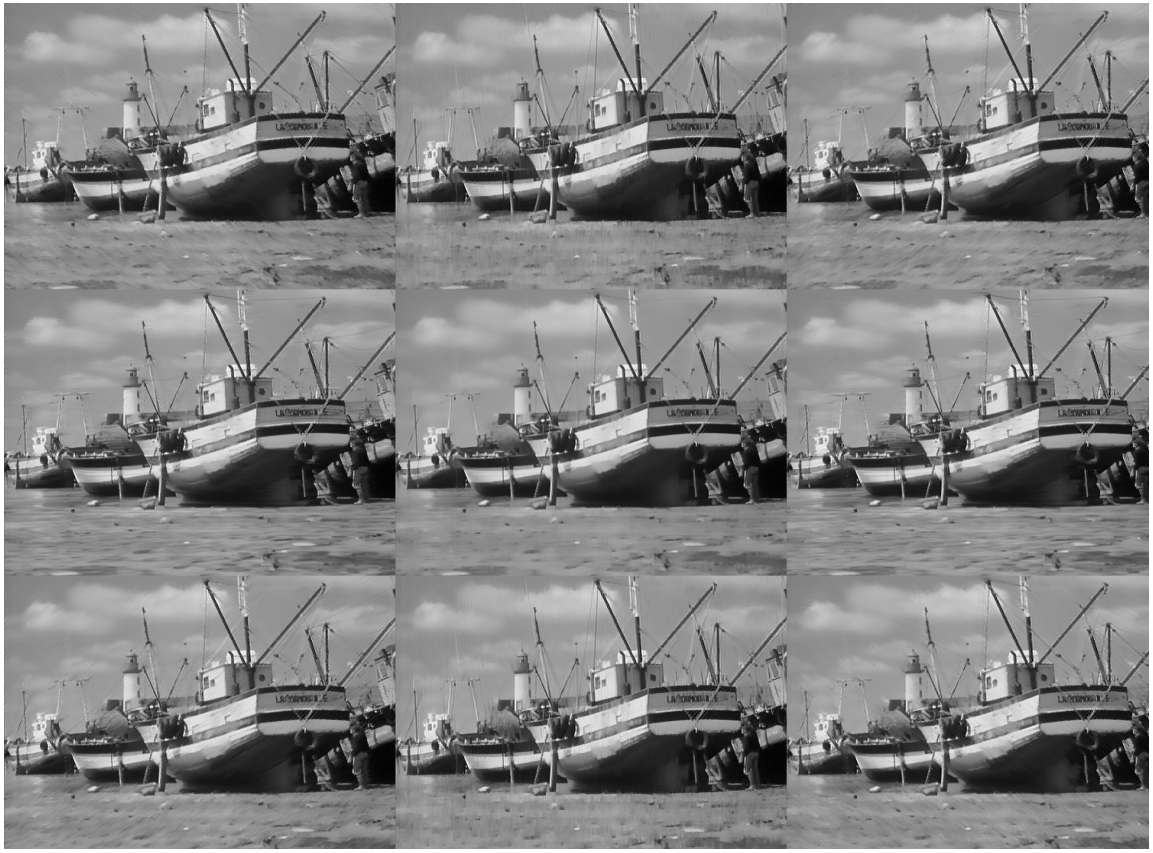
Repeat steps $(2)-(3)$ up to the convergence of the algorithm.

Illustrations

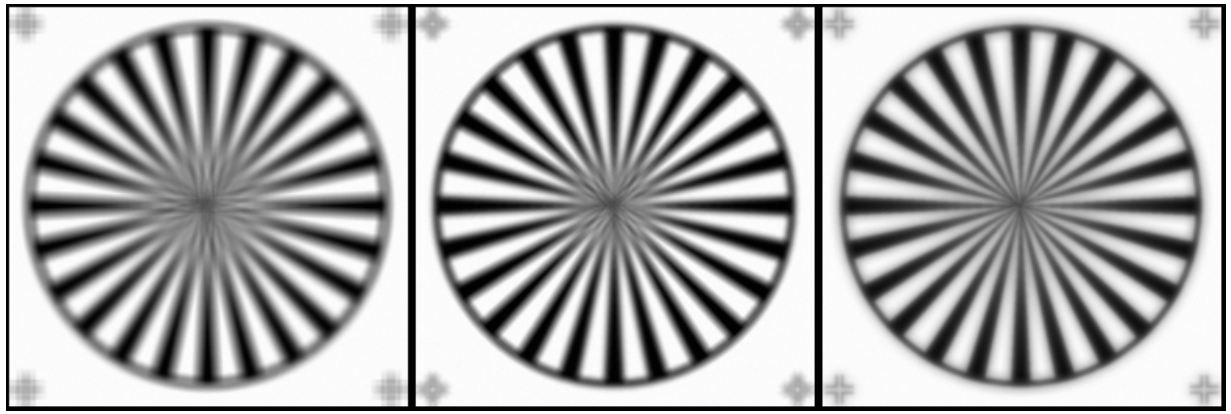
Three channel noisy observations of "Boats."



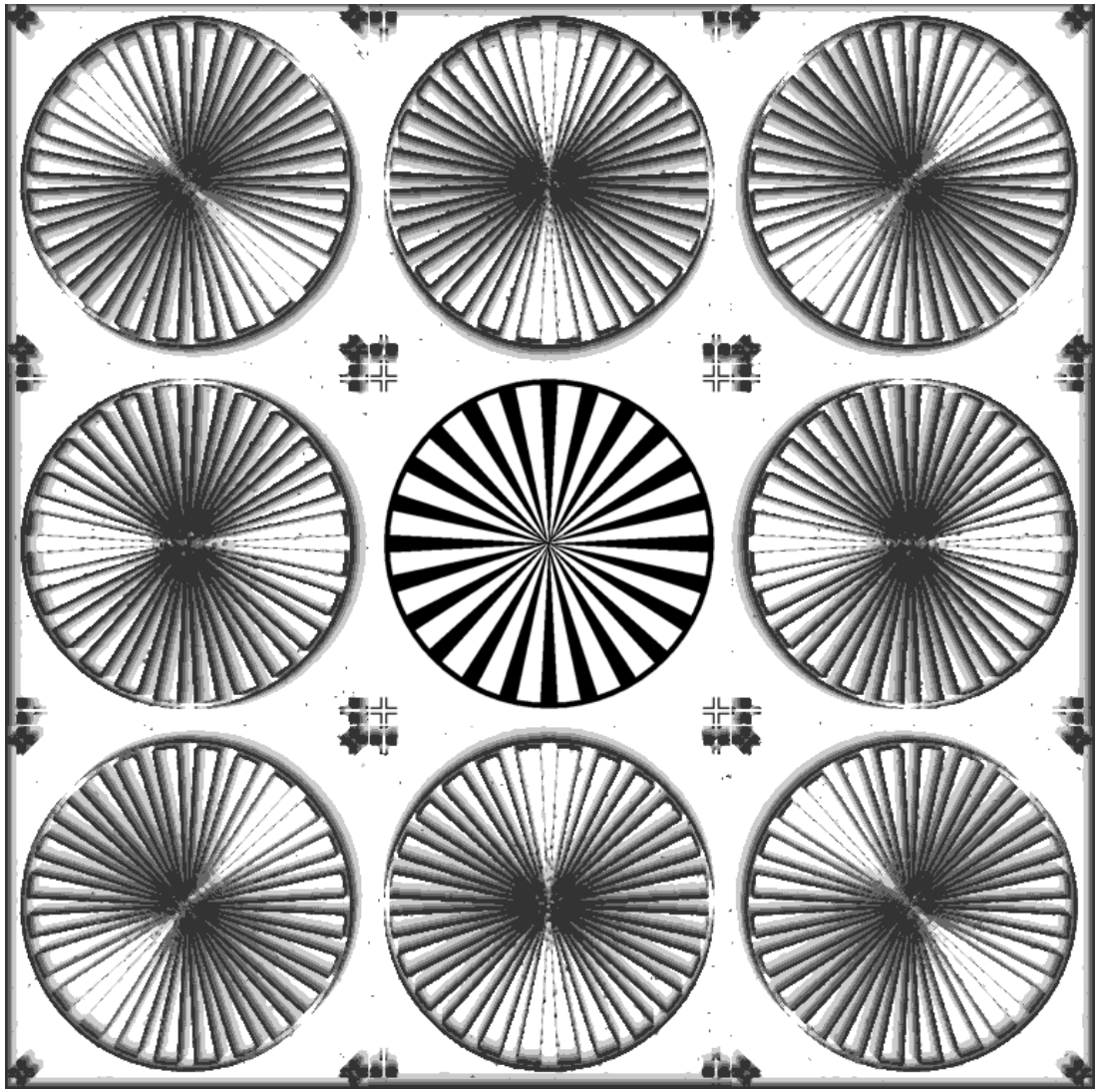
ICI adaptive directional scales $\hat{h}_j^+(x)$, $\theta_j = (j-1)\pi/4, j = 1, \ldots, 8$.



ICI adaptive directional estimates, $\theta_j = (j-1)\pi/4, j = 1, \ldots, 8$.



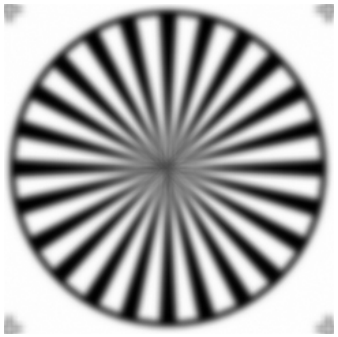
Noisy ”testpat” images.



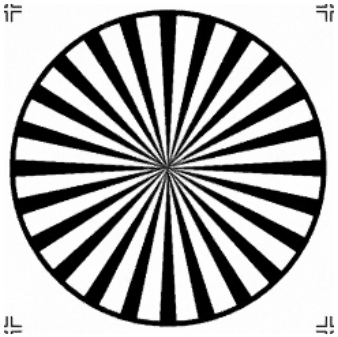
ICI adaptive directional scales $\hat{h}_j^+(x)$, $\theta_j = (j-1)\pi/4, j = 1, \ldots, 8$.

Initial guess, estimate and estimation errors

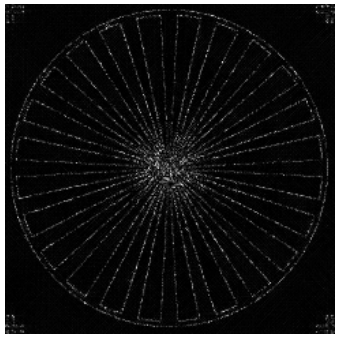
INITIAL GUESS



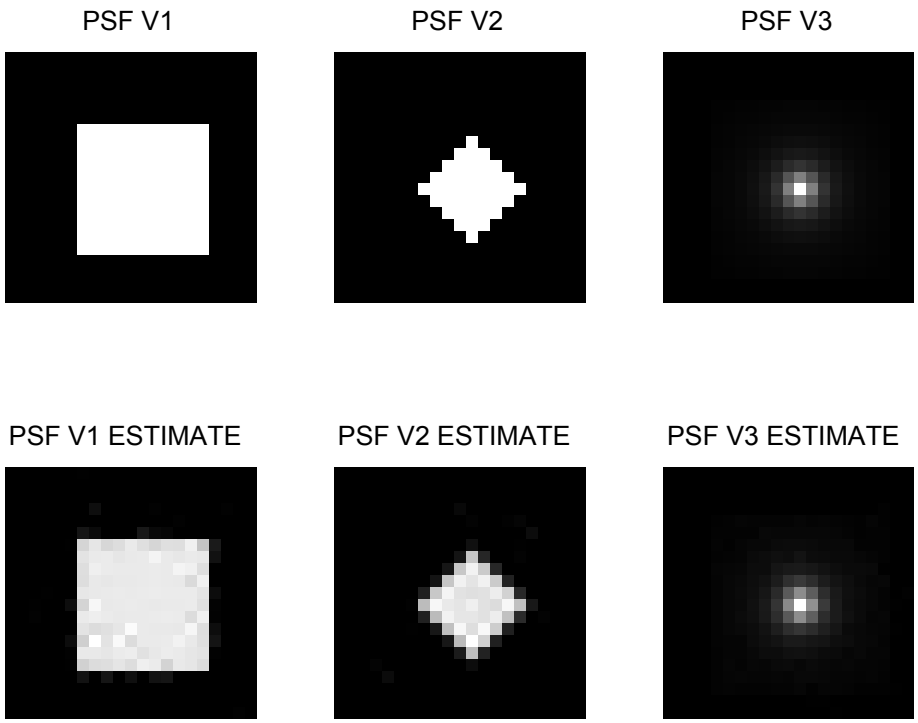
ESTIMATE



ERRORS



The estimates and true *PSFs* of the three channel imaging system.



Local Maximum Likelihood

Parametric ML

Assume that the signal y depends on the argument x and provided a fixed $x = X_s$ an observed z_s is random with a probability density f.

This stochastic model can be written in the following general form

$$z_s \sim f(z, y_s),$$

where $f(z, y)$ is a conditional $p.d.f.$ provided that x is fixed and the signal y is a parameter of this $p.d.f.$ with $y_s = y(X_s)$.

Let observations be given by the pairs $\{z_s, X_s\}$, $s = 1,...,n$. The dependence on X_s for z_s goes through $y_s = y(X_s)$. The estimation problem, we are interested in, is a reconstruction of the signal $y(x)$ from the observations $\{z_s, X_s\}$, $s = 1, ...,n$.

Denote the parametric model of y by $y_C(x)$:

$$y_C(x) = C^T \phi(x),$$

where $\phi \in \mathbb{R}^M$ is a vector of basis functions.

Assume that the observations $\{z_s, X_s\}$ are independent for different s.

The joint $p.d.f.$ of the observation pairs is a product of the densities for each pair. Replace possible values of z in this joint $p.d.f.$ by the observed z_s. The value obtained in this way is called a likelihood. Considered as a function of the parameter C the likelihood is represented in the form

$$l(C) = \Pi_s f(z_s, y_C(X_s)).$$

The log – likelihood is defined as the logarithm of the likelihood

$$L(C) = \sum_s \ln f(z_s, y_C(X_s)).$$

The ML estimate can be defined as

$$\hat{C} = \arg\max_{C} L(C).$$

Provided some relevant regularity conditions on f the ML estimates have desirable optimality properties:

1. These estimates are asymptotically unbiased and efficient. The efficiency means that the estimator has the smallest variance, and thus the narrowest confidence interval, of all estimators;
2. They have approximate normal distributions and approximate sample variances that can be used to generate confidence bounds and hypothesis tests for the parameters.

Nonparametric maximum likelihood

General approach

The local ML is a nonparametric counterpart of the parametric ML technique.

The local likelihood and local log-likelihood use the window function w for localization of estimation in a neighborhood of the point of the interest (center) x and replace the global parametric model $y_C(x)$ by a local one.

For the polynomial model $y_h(x, v)$ this local log – likelihood criterion can be presented in the form

$$L_h(x, C) = \sum_s \ln f(z_s, y_h(x, X_s)) w_h(x - X_s),$$

$$y_h(x, X_s) = C^T \phi_h(x - X_s),\ \phi_h(x) = \phi(x/h),$$

$$w_h(x) = w(x/h)/h^d,\ C \in \mathbb{R}^M.$$

What is a difference between the local and non-local likelihood?

Following the idea of the local nonparametric estimation we use the criterion $L_h(x, C)$ for computing C and the model $y_h(x, v)$ is exploited in the pointwise manner for $x = v$ only.

In this way we arrive to the following local ML version of the LPA:

$$. \hat{C} = \arg\max_{C \in \mathbb{R}^M} L_h(x, C),$$

$$\hat{y}_h(x) = \hat{C}^T \phi(0),\ \hat{y}_h^{(r)}(x) = (-\frac{1}{h})^{|r|} \hat{C}^T \phi^{(r)}(0).$$

All ideas of the standard LPA concerning the window w (shape, anisotropy, directionality, etc.), the scaling h (scalar, multivariate), estimation of the signal and derivatives are naturally valid in this nonparametric pointwise estimation.

The estimate $\hat{C}$ is a solution of the equation $\partial_C L_h(x, C) = 0$.

Additive i.i.d. noise

First of all the observation model is changed from the global $y_C(x)$ to the local one:

$$z_s = y_h(x, X_s) + \varepsilon_s,\ s = 1, \ldots, n,$$
$$y_h(x, X_s) = C^T \phi_h(x - X_s),$$

where the random ε_s are i.i.d. with the probability density f.

The probability density of z_s is $f(z_s - y_h(x, X_s))$ and for the set of observations we have the local log-likelihood in the form:

$$L_h(x, C) = \sum_s w_h(x - X_s) \ln f(e_s),\ e_s = z_s - y_h(x, X_s).$$

Denote

$$\psi(v) = -\ln f(v).$$

Then $\sum_s \ln f(e_s) = -\sum_s \psi(e_s)$ and the local log-likelihood estimates are

found from the problem

$$\hat{C} = \arg \min_{C \in \mathbb{R}^M} \sum_s \psi(e_s) w_h(x - X_s),\ e_s = z_s - y_h(x, X_s).$$

where ψ is a loss function of the residuals.

In this way we derive the local nonparametric M-estimates with the loss function ψ defined through the probability density f.

Gaussian noise

Let the observation noise be Gaussian with the invariant variance, $\varepsilon \sim N(0, \sigma^2)$. Omitting the invariant part $\ln(\sqrt{2\pi}\,\sigma)$ in $\ln f = -\ln(\sqrt{2\pi}\,\sigma) - v^2/(2\sigma^2)$ we can define the loss function as $\psi(v) = v^2/\sigma^2$ with the weighted mean squared residual criterion

$$\sum_s w_h(x - X_s)e_s^2,\ e_s = z_s - y_h(x, X_s).$$

Then, the local ML estimate in the form

$$\hat{C} = \arg\min_{C \in \mathbb{R}^M} \sum_s w_h(x - X_s)e_s^2,$$

gives the standard linear LPA.

Laplacian noise

Let the observation noise in the additive noise model be independent Laplacian, i.e. $f(v) = \frac{1}{2a}\exp(-|v|/a)$. Then $\ln f(v) = \ln(1/(2a)) - |v|/a$ and the loss function is defined as $\psi(v) = |v|$ with the modulus residual criterion

$$\sum_s |e_s| w_h(x - X_s),\ e_s = z_s - y_h(x, X_s).$$

In this way we derive the M-estimates with the loss function $\psi(x) = |x|$ as the local ML estimate.

Theory for local ML

Proposition. Let p.d.f. $f(z,y)$ be twice differentiable on y with the finite Fisher information I_y. Under the assumptions of the accuracy proposition the accuracy of the multivariate discrete local ML estimates is defined by the following expressions:

for the estimation bias

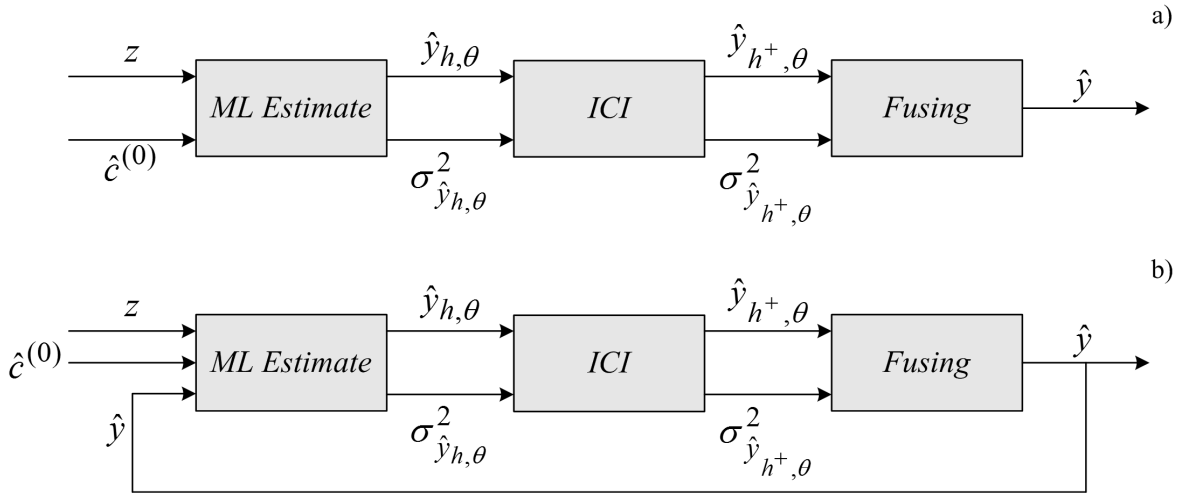
$$m_{\hat{y}_h^{(r)}}(x,h) \sim (-1)^m h^{m+1-|r|} \sum_{|k|=m+1} y^{(k)}(x) \frac{1}{k!} \int_{\mathbb{R}^d} g^{(r)}(u) u^k du, \; m \geq |r|,$$

for the estimation variance:

$$\sigma^2_{\hat{y}_h^{(r)}}(x,h) \sim \frac{1}{I_y(x)} \frac{\Delta^d}{h^{d+2|r|}} \int_{\mathbb{R}^d} (g^{(r)}(u))^2 du,$$

$$I_y(x) = E\{(\partial_y \ln f(z,y(x)))^2\} = -E\{\partial_y^2 \ln f(z,y(x))\}.$$

LPA – ICI algorithms



Flow charts of the *ML* algorithm: a) open loop algorithm; b) feedback algorithm.

The open loop algorithm includes the following blocks:

- ML estimation. The inputs are the observations z and the initial estimate of the LPA parameter $\hat{C}$. The outputs are the directional estimates $\hat{y}_{h,\theta}$, $h \in H$, and the corresponding variances $\sigma^2_{\hat{y}_{h,\theta}}$.
- ICI rule. The inputs are the estimates $\hat{y}_{h,\theta}$ and their variances $\sigma^2_{\hat{y}_{h,\theta}}$. The outputs are the adaptive scale estimates $\hat{y}_{h^+,\theta}$ and their variances $\sigma^2_{\hat{y}_{h^+,\theta}}$.
- Fusing. The inputs are the adaptive scale estimates $\hat{y}_{h^+,\theta}(x)$ and their variances $\sigma^2_{\hat{y}_{h^+,\theta}}(x,h)$. The outputs are the fused estimates $\hat{y}(x)$ and their variances $\sigma^2_{\hat{y}}(x)$.

Local Quasi-Likelihood

There are many practical circumstances in which even though the full likelihood is unknown, one can specify the relationship between the mean and the variance. In this situation estimation of the mean (regression, local regression) can be achieved by replacing the log-likelihood, $\ln f(z,y)$, by a quasi-likelihood function $Q(z,y)$.

The quasi-likelihood has been proposed as a distribution free method (note that contrary to it the ML is a distribution based method).

In many applications the noise that corrupts the signal is signal dependent, the most widely encountered types being multiplicative, Poisson, film-grain, and speckle noise. Their common feature is that the power of the noise is related to the magnitude of the signal.

Parametric quasi-likelihood

One assumes that we have independent observations z with the expectations $y = E\{z\}$ and the variance of these observations is modeled as

$$\sigma^2(y) = \sigma_0^2 \rho(y),$$

where σ^2 is unknown and ρ is a given positive function of y called a variance function.

The quasi-likelihood function $Q(z,y)$ is defined by the relation:

$$\partial_y Q(z,y) = (z-u)/\sigma^2(y).$$

Naturally,

$$Q(z,y) = \int_{-\infty}^{y} \frac{z-u}{\sigma^2(u)} du + \text{function of } z.$$

If y allows the parametric representation

$$y_C(x) = C^T \phi(x)$$

the coefficient C is found by maximization of the quasi-likelihood

$$\hat{C} = \max_{C} \sum_{s} Q(z_s, y_C(X_s)).$$

As the maximum conditions are

$$\sum_{s} \partial_C Q(z_s, y_C(X_s)) \quad = 0$$

we do not need to know the quasi-likelihood function Q.

The calculations give the equation quasi – linear on C:

$$\sum_{s} \frac{z_s - \phi^T(X_s)C}{\sigma^2(y_C(X_s))} \phi(X_s) = 0.$$

We use the term quasi – linear in order to emphasize the fact that the equation is nonlinear on C as the weights $1/\sigma^2(y_C(X_s))$ depend on the unknown C.

Nonparametric quasi-likelihood

A local version of the quasi-likelihood model can be defined using the window function w as weights of the residuals and the LPA model $y_h(x, X_s)$ for the estimated $y(x)$.

It gives the local quasi-likelihood equations in the form

$$\sum_s \frac{z_s - y_h(x, X_s)}{\sigma^2(y_h(x, X_s))} w_h(x - X_s)\phi_h(x - X_s) = 0,$$

$$y_h(x, X_s) = C^T \phi_h(x - X_s).$$

The estimates of C are found and the variances $\sigma^2(y(X_s))$ in the weights are updated

$$\sigma^2(y_h(x, X_s)) = \sigma^2(\hat{C}^T \phi_h(x, X_s)).$$

Thus, we arrive to the estimates which are nonlinear with respect to the observations. However, each iteration requires to solve only linear equations.

The estimates of C are used for estimation of y and the derivatives in the standard LPA style

$$\hat{y}_h(x) = \hat{C}^T \phi(0),$$
$$\hat{y}_h^{(r)}(x) = (\frac{-1}{h})^{|r|} \hat{C}^T \phi^{(r)}(0).$$

Quasi-likelihood zero-order LPA – ICI

Let us use the quasi-likelihood equation for $m = 0$.

As the estimate $\hat{y}_h(x, X_s) = \hat{C}$ is constant the variance $\sigma^2(y_h(x, X_s))$ is also constant and the solution can be given in the explicit form as the weighted mean

$$\hat{C} = \sum_s z_s g_h(x - X_s), \; g_h(x - X_s) = \frac{w_h(x - X_s)}{\sum_s w_h(x - X_s)},$$

with the regression estimate

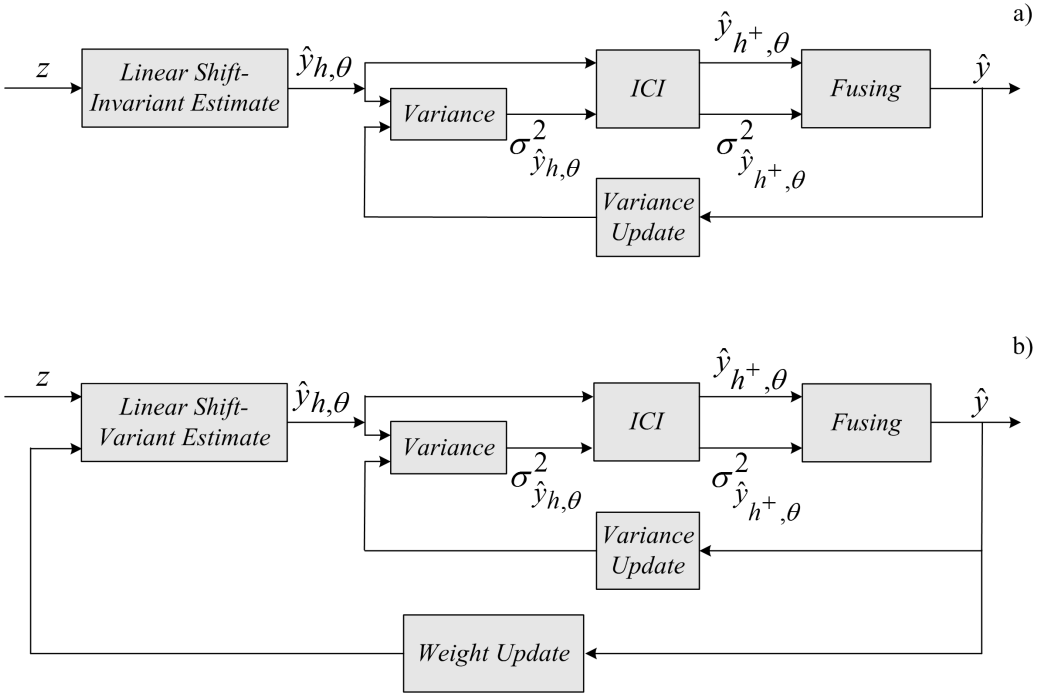
$$\hat{y}_h(x) = \hat{C}.$$

The observation variance is calculated as $\sigma^2 \rho(\hat{y}_h(x))$ with the variance for the estimate

$$\sigma^2_{\hat{y}_h}(x) = \sigma_0^2 \sum_s \rho(\hat{y}_h(X_s)) g_h^2(x - X_s) = \sigma_0^2 \cdot (\rho(\hat{y}_h) \circledast g_h^2)(x).$$

The formulas define the conventional linear Nadaraya – Watson estimate.

$LPA-ICI(AV)$ algorithm



(a) $LPA-ICI(AV)$ algorithm with the adaptive variance (AV) in the ICI rule,

(b) $LPA(AV)-ICI(AV)$ algorithm with AV used both in the LPA by updating

the weights of the estimator and in the ICI rule.

Photon-Limited Imaging

Modeling

In many imaging systems recorded observations have a physical meaning of numbers of detected photons. The photons are counted at different spatial locations and in this way form an image of object. The Poisson distribution is a conventional probabilistic model used for a random number of photons appeared during an exposure time.

The direct observation model has a form

$$z_s \sim \mathrm{P}\,(y(X_s)).$$

It means that

$$P\{z = k\} = \exp(-y)\frac{y^k}{k!},\ k = 0, 1, \ldots,$$

$$E\{z\} = y,\ var\{z\} = y.$$

where y stays for the intensity or the mean value of the Poisson distribution.

Besides, we consider the indirect observation assuming that a measurement at the position x is obscured by effects from nearby points (tissues, bones, objects, etc.).

These effects are modelled by the convolution kernel v which can be treated as the PSF of the system.

Let v be shift-invariant and the observation mean (not observations themselves !) can be presented in the convolution form $E\{z(x)\} = (y \circledast v)(x)$.

We arrive to the model

$$z_s \sim \mathrm{P}\left((y \circledast v)(X_s)\right).$$

We have for the mean of z_s :$E\{z_s\} = (y \circledast v)(X_s)$, and for the variance of z_s: $\sigma_z^2(x) = var\{z_s\} = (y \circledast v)(X_s)$.

Direct Poisson observations

Let us start assuming that the sensor is ideal: $z(x) \sim \mathrm{P}\,(y(x))$.

The following two methods can be exploited as basic ones for nonparametric estimation of y:

- Anscombe nonlinear transform;
- Local quasi-likelihood.

Anscombe transform

Replaces the observation z by

$$\xi = 2\sqrt{z + 3/8}\,.$$

The Anscombe transform enables the variance stabilizing effect and the new observations $\xi(x)$ have an approximately invariant variance for all x. The adaptive varying scale $LPA - ICI$ filtering techniques are applicable to the transformed observations.

If the $LPA - ICI$ estimate $\hat{\xi}$ of ξ is found the final step of the algorithm is to

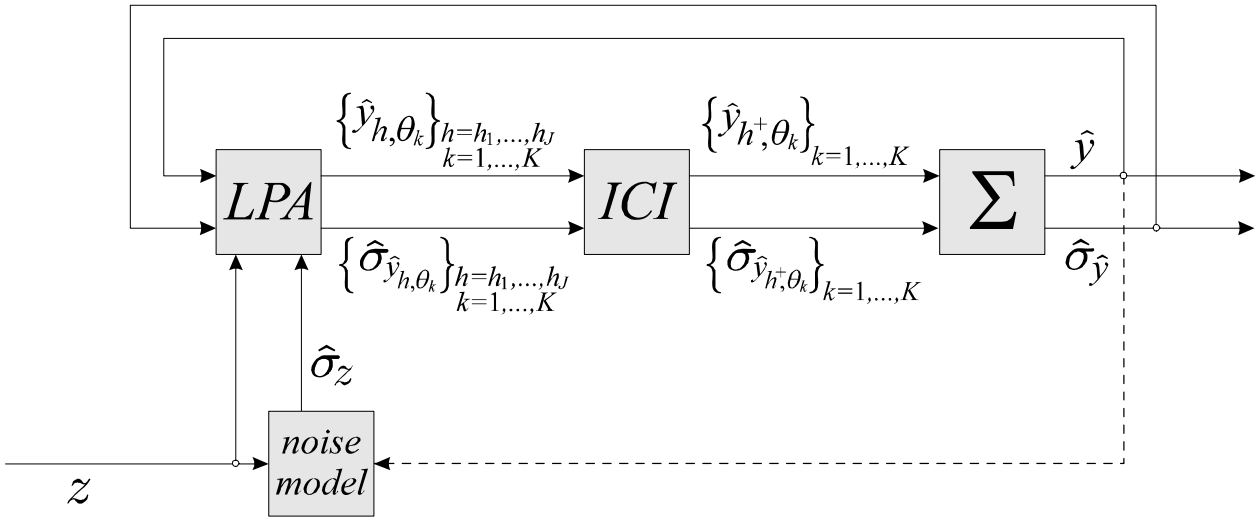
apply the inverse Anscombe transform with the estimate for y as

$$\hat{y} = \hat{\xi}^2/2 - 3/8.$$

Local quasi-likelihood

The quasi-likelihood algorithms with $\rho(y) = y$ are applicable for the direct Poisson observation problem.

Recursive LPA – ICI(AV) algorithm



Numerical experiments (demo_DenoisingSignDepNoise.m)

Observations

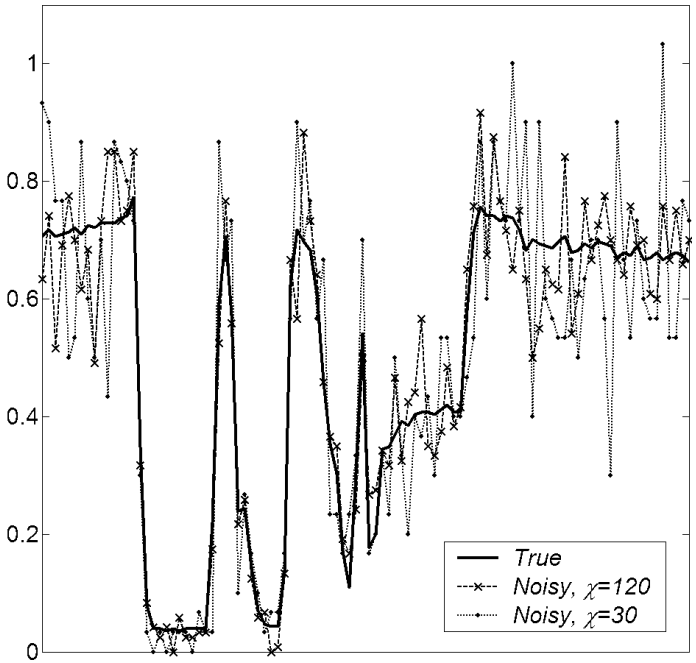
In order to achieve different levels of randomness (i.e. different SNR) in the noisy observations we multiply the true signal y (which has range $[0, 1]$) by a scaling factor $\chi > 0$:

$$z \sim \mathrm{P}(y \cdot \chi).$$

For larger χ we have a smaller relative level of the noise.

Noisy image with χ =120. Noisy image with χ =30.

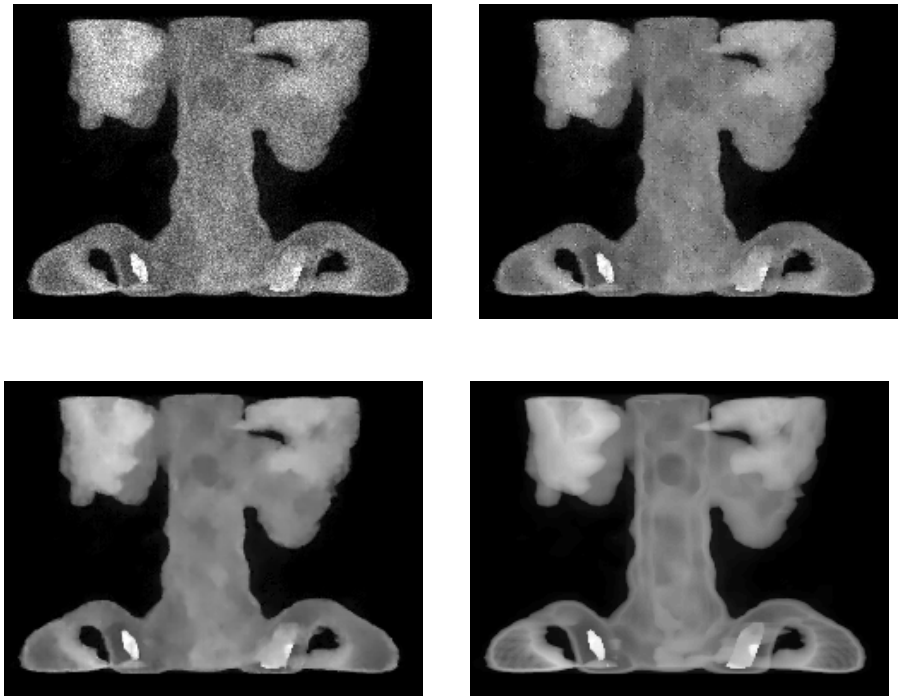
Cameraman fragment: Poisson noisy normalized images.



A cross-section of the normalized Cameraman image: true image and Poisson noisy images with $\chi = 30$ and $\chi = 120$.

Cameraman test-image: top left - noisy data $\chi = 60$ (MSE $= 23.9$); top right - first step (MSE $= 7.90$, IMSE $= 4.81$dB); second row left - fourth step (MSE $= 4.36$, ISNR $= 7.40$dB); second row right - original image.

Lena test-image: top row, left - noisy data $\chi = 60$ (MSE $= 29.0$, ISNR $= 15.3$dB); top row, right - first step (MSE $= 7.07$, ISNR $= 6.13$dB); second row, left - fourth step (MSE $= 2.62$, SNR $= 10.4$dB), second row, right - original image.



Spine test-image: data (MSE $= 14.8$, ISNR $= 15.1$dB); first row right - first step (MSE $= 3.91$, ISNR $= 5.79$dB); second row left - fourth step (MSE $= 1.56$, ISNR $= 9.77$dB); second row right - original image.

MSE comparison for Cameraman image

χ value	30	60	90	120
Noisy image	13.9	27.5	42.1	56.0
Timmerman Nowak method	2.76	7.73	14.11	21.59
Improved TN method	2.13	5.37	9.22	13.59
LPA – ICI(AV), 1^{th} step	3.75	8.04	13.4	18.8
LPA – ICI(AV), 2^{nd} step	2.28	5.30	8.96	13.2
LPA – ICI(AV), 3^{rd} step	1.79	4.50	7.79	11.8
LPA – ICI(AV), 4^{th} step	1.62	4.30	7.58	11.6
Anscombe LPA – ICI	1.78	4.49	7.75	11.7

MSE comparison for Lena image

χ value	30	60	90	120
Noisy image	14.5	29.0	43.6	58.0
Timmerman Nowak method	2.33	6.46	11.62	17.89
Improved TN method	1.98	5.32	9.35	14.03
LPA – ICI(AV), *1 step*	3.29	7.07	11.3	15.9
LPA – ICI(AV), 2 step	1.76	4.07	6.80	9.84
LPA – ICI(AV), 3 step	1.20	3.05	5.33	7.99
LPA – ICI(AV), 4 step	0.99	2.62	4.72	7.19
Anscombe LPA – ICI	1.23	3.13	5.44	8.16

Inverse Poissonian Imaging

Local ML for indirect observations

We consider indirect observations corresponding to the model where the blurring operator is given by the convolution.

Introduce a local version of the likelihood as

$$L_h(x, C) = \sum_s [-(y_h \circledast v)(X_s) + z_s \log((y_h \circledast v)(X_s))] w_h(x - X_s).$$

Assuming that the zero order LPA is used for the convolution $(y_h \circledast v)(X_s)$ we arrive to

$$L_h(x, C) = \sum_s (-C + z_s \log C) w_h(x - X_s).$$

Then the ML estimate of C has a form

$$\hat{C}_h(x) = (z \circledast g_h)(x),\ g_h(x) = \frac{w_h(x)}{\sum_x w_h(x)},$$

where $\hat{C}_h(x)$ is the nonparametric estimate for $(y_h \circledast v)(x)$.

If $\hat{C}_h(x)$ is found the estimate $\hat{y}_h$ of y_h is a solution of the equation

$$(\hat{y}_h \circledast v)(x) = \hat{C}_h(x).$$

In the frequency domain these equations give

$$V(f)\hat{Y}_h(f) = G_h(f)Z(f).$$

Thus, we arrive to the linear inverse problem. The unknown $\hat{Y}_h(f)$ is a solution of the ill-conditioned linear system. Remind that the conditioning of this system is defined by the convolution kernel v.

The adaptive methods developed for the linear inverse are fully applicable to the linear equation.

Linear inverse plus $LPA-ICI$ filtering

This algorithm uses the RI and RWI inversions developed for the linear Gaussian inverse. Adaptation to the Poisson case concerns some changes of the RWI inverse and the variance formulas exploited for the ICI rule.

Experiments (demo_DeblurringPoisson.m)

In order to achieve different level of randomness the noise-free signal $(y \circledast v)$ (which has range $[0,1]$) is multiplied by a scaling factor $\chi > 0$.

Then $E\{z\} = var\{z\} = (y \circledast v) \bullet \chi$ and the signal-to-noise ratio is calculated as

$$SNR = \sqrt{(y \circledast v) \bullet \chi}$$

For smaller χ we have a larger relative level of the noise.

The blur PSF is the uniform (box-car) 9×9 function, $\chi = 17600$. Despite so large χ, the noise is still quite an essential issue for the Poisson deblurring.

Deblurring the Poissonian Cameraman image.

From left to right: noisy blurred observations; the reconstructed image obtained by the Poissonian adaptive deconvolution algorithm ISNR $= 6.61$dB; deblurring the Poisson data with the algorithm developed for the Gaussian observations gives a poor estimate, ISNR $= 5.38$dB; after parameter's optimization, ISNR $= 6.03$dB.

Summary

We have introduce the adaptive multiscale multidirectional filter bank and demonstrated its efficiency in a number of image processing tasks.

In conclusion we wish to make some final notes.

- Heisenberg/suncertainty and the developed multidirectional multiscale filters.

  The Heisenberg/s notion of frequency and space/time uncertainty leads to the lower bound on the product of the resolution of the filter in time and frequency

$$\bar{\sigma}_x \bar{\sigma}_\lambda \geq 1/2.$$

  Here $\bar{\sigma}_x^2$ and $\bar{\sigma}_\lambda^2$ are energy spreads of the filter impulse response g(x) and its Fourier transform G(λ):

$$\bar{\sigma}_x^2 = \frac{\int_{\mathbb{R}} x^2 |g(x)|^2 dx}{\int_{\mathbb{R}} |g(x)|^2 dx}, \bar{\sigma}_\lambda^2 = \frac{\int_{\mathbb{R}} \lambda^2 |G(\lambda)|^2 d\lambda}{\int_{\mathbb{R}} |G(\lambda)|^2 d\lambda}.$$

Small $\bar{\sigma}_x^2$ and $\bar{\sigma}_\lambda^2$ mean a high concentration of the corresponding energy in the signal or transform domains.

The Heisenberg's inequality shows that the product $\bar{\sigma}_x \bar{\sigma}_\lambda$ is at least 1/2. The minimum value of this product equal to 1/2 is attained by the Gaussian g(x).

It follows that the Gaussian (smooth) kernel filters are the best in terms of the best space-frequency resolution.

In contradiction to this standard requirements in our filter-bank highly non-smooth sharp kernels are much more efficient than the smooth Gaussian ones.

It allows ones more to emphasize that the proposed filter system is nonstationary (switch, match filtering) and the conclusions like Heisenberg's uncertainty obtained for the shift-invariant stationary filtering are not applicable here.

- Limitations of the considered approach follow from the very idea of the local approximation. Small details of the image which are so small that there is no a neighborhood of the size sufficient for noise suppression cannot be restored from noisy data.

Different style methods using non-polynomial approximations in larger neighborhoods can be more efficient in this sort of situations.