TAMPERE UNIVERSITY OF TECHNOLOGY
Institute of Signal Processing – Transforms and Spectral Methods Group

Spatially adaptive local approximations in signal and image processing: varying-scale polynomials, anisotropic adaptation, shape-adaptive transforms

Alessandro Foi

www.cs.tut.fi/~foi

EUSIPCO 2007 - Poznań, Poland 3/9/2007

Outline:

PART 1

preliminaries (best approximations, reconstruction formulas, dual frames, etc.)
Local Polynomial Approximation (moving least-squares)
LPA kernels
varying scale estimates
ideal scales
adaptive scales: the Intersection of Confidence Intervals rule
LPA-ICI algorithm
Directional LPA
Anisotropic LPA-ICI

PART 2

Shape-Adaptive DCT
Pointwise Shape-Adaptive DCT filters: denoising, deblocking, deblurring
"Shape-Adapted" Transforms
Non-local approach: Block-Matching 3D filtering algorithm

PART 1

Best approximation in a closed subspace $\mathcal{M}$ of a Hilbert space $\mathcal{H}$

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

Solution: $\hat{\varphi}$ is the orthogonal projection of f onto $\mathcal{M}$

Let $\{\phi_n\}_n$ be a system of generators for $\mathcal{M}$, then

$$\hat{\varphi} = \sum_n \langle f, \phi_n \rangle_{\mathcal{H}} \, \phi_n \quad \text{for } \{\phi_n\}_n \text{ orthonormal system, or}$$

$$\hat{\varphi} = \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \, \phi_n \quad \text{for } \{\phi_n\}_n \text{ frame and } \{\breve{\phi}_n\}_n \text{ dual frame.}$$

Window / weight function, associated measure and space

window function $w \rightsquigarrow \mathcal{H} = L^2\left(\mathbb{R}^d, \mu\right)$ $(w \geq 0$ density of $\mu)$

$$\begin{aligned} \langle f, g\rangle_{\mathcal{H}} &= \int f\overline{g}d\mu = \int f(v)\,\overline{g(v)}w(v)\,dv \\ \|f\|_{\mathcal{H}}^2 &= \langle f, f\rangle_{\mathcal{H}} = \int |f|^2\,d\mu = \int |f(v)|^2\,w(v)\,dv \end{aligned}$$

Best approximation in a closed subspace $\mathcal{M} \subset L^2\left(\mathbb{R}^d, \mu\right) = \mathcal{H}$

$$\hat{\varphi} = \underset{\varphi \in \mathcal{M}}{\operatorname{argmin}} \|\varphi - f\|_{\mathcal{H}} = \underset{\varphi \in \mathcal{M}}{\operatorname{argmin}} \int |\varphi - f|^2 \, d\mu = \underset{\varphi \in \mathcal{M}}{\operatorname{argmin}} \int |\varphi(v) - f(v)|^2 \, w(v) \, dv$$

Let $\{\phi_n\}_n$ be a frame for $\mathcal{M}$ and $\{\breve{\phi}_n\}_n$ its dual frame, then

$$\hat{\varphi} = \sum_n \int f \overline{\breve{\phi}_n} d\mu \phi_n = \sum_n \int f(v) \overline{\breve{\phi}_n(v)} w(v) \, dv \phi_n$$

Matrix form:

$$\hat{\varphi} = \underset{\varphi \in \mathcal{M}}{\operatorname{argmin}} \|\varphi - f\|_{\mathcal{H}} = \underset{\varphi \in \mathcal{M}}{\operatorname{argmin}} \left((\varphi - f)^T \mathbf{w} (\varphi - f)\right)$$

$$\hat{\varphi} = \boldsymbol{\phi} \breve{\boldsymbol{\phi}}^T \mathbf{w} f,$$

where $\mathbf{w} = \operatorname{diag} w$ is the weight matrix, $\boldsymbol{\phi}$ is the frame matrix, $\breve{\boldsymbol{\phi}} = \boldsymbol{\phi} \boldsymbol{\Phi}^{\dagger}$ is the dual frame matrix, and $\boldsymbol{\Phi} = \boldsymbol{\phi}^T \mathbf{w} \boldsymbol{\phi}$ is the Gramian matrix.

This is equivalent to a standard weighted least-squares problem.

Pointwise estimation $\rightsquigarrow$ kernel estimates

$$\begin{aligned}\hat{\varphi}(0) &= \sum_n \langle f, \breve{\phi}_n \rangle_{\mathcal{H}} \phi_n(0) = \left\langle f, \textstyle\sum_n \breve{\phi}_n \overline{\phi_n(0)} \right\rangle_{\mathcal{H}} = \\ &= \int f(v) \textstyle\sum_n \overline{\breve{\phi}_n(v)} \phi_n(0)\, w(v)\, dv = \int f(v)\, g_{\mathcal{M}}(v)\, dv\end{aligned}$$

The kernel $g_{\mathcal{M}}$ depends only on $\mathcal{M}$ (i.e. on $\mathcal{M}$ and w).

"Moving" least-squares method $\rightsquigarrow$ convolution estimates

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

$\hat{y}(x) = \hat{\varphi}_x(0)$, where $\hat{\varphi}_x = \operatorname{argmin}_{\varphi \in \mathcal{M}} \|\varphi - \tilde{z}_x\|_{\mathcal{H}}$ and $\tilde{z}_x = z(x - \cdot)$.

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

LPA (local polynomial approximation)

$\mathcal{M}$ is a subspace of real polynomials

There are only two design parameters for $\mathcal{M}$:

polynomial order O_m : $\mathcal{M} = \left\{\varphi : \varphi(v) = \sum_{o \in O_m} c_o v^o\right\}$

window function w : $\mathcal{M} \subset \mathcal{H}, \quad \langle f, g \rangle_{\mathcal{H}} = \int f g d\mu = \int f(v)\, g(v)\, w(v)\, dv$

Clearly, $\hat{\varphi}$ does not depend on the frame used to construct $\mathcal{M}$. Nevertheless, it is very convenient to use a basis whose generators are monomes, and in particular an ideal choice is to use the standard Taylor basis,

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

where $o! = o_1!\cdots o_d!$, $o = o(n) \in \mathbb{N}^d$, $\phi_n(v) = \frac{v^{o(n)}}{o(n)!}$. Indeed, since $\phi_n(0) = \delta(n-1)$, the *function-estimation kernel* is

$$g_{\mathcal{M}}(v) = w(v)\sum_{l,n}\phi_l(v)\,\mathbf{\Phi}^{\dagger}(l,n)\,\phi_n(0) = w(v)\sum_{l}\phi_l(v)\,\mathbf{\Phi}^{-1}(l,1). \tag{2}$$

The use of Taylor basis allows to define the *derivative-estimation kernels* as

$$g_{\mathcal{M}}^{(o(n))}(v) \triangleq w(v)\sum_{l}\phi_l(v)\,\mathbf{\Phi}^{-1}(l,n). \tag{3}$$

The convolution against them yields an estimate for the function and all its derivatives (for all orders $o \in O_m$)

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

where $\hat{y}^{(o)}(x) = \left(D^{(o)}\widetilde{\hat{\varphi}_x}\right)(x)$,with $\widetilde{\hat{\varphi}_x} = \hat{\varphi}_x(\cdot - x)$ being the best approximation of z in the space $\tilde{\mathcal{M}}_x$. Thus, $\hat{y}^{(o)}(x)$ is in fact an estimate of the o-th (partial) derivative of y.

Vanishing moments

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

$$\int g_{\mathcal{M}}(v)\,dv = 1, \tag{4}$$

$$\int v^o g_{\mathcal{M}}(v)\,dv = 0, \quad o \in O_m \setminus \{0\}. \tag{5}$$

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

Zero-order LPA

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

$$g_{\mathcal{M}} = w\phi_1 \left(\int \phi_1(v)\,\phi_1(v)\,w(v)\,dv \right)^{-1} = \frac{w}{\int w(v)\,dv}. \tag{6}$$

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

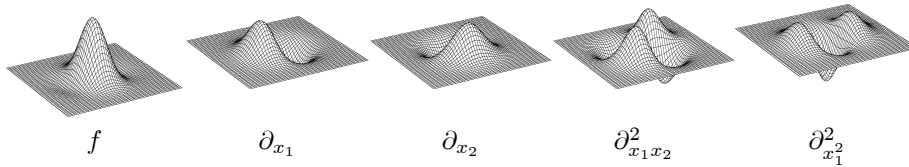


Figure 2: Function and derivative estimation *LPA* kernels obtained for $m = (2, 1)$ using a Gaussian window function w.

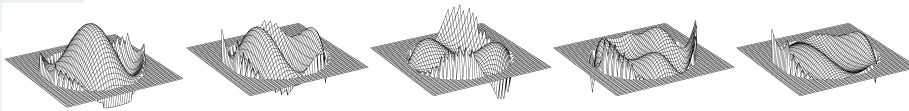


Figure 3: The function estimation *LPA* kernel and some of the derivative estimation *LPA* kernels obtained for $m = (5, 3)$ using the characteristic function of a disk as the window function w.

Scale parameter h

$$w_h(\cdot) \triangleq w(\cdot/h), \qquad h \in \mathbb{R}^+,$$
$$g_h(\cdot) = h^{-d} g_{\mathcal{M}}(\cdot/h).$$

h controls the "size" of the kernel support

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

LPA kernels as smoothers

In frequency domain, we have

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

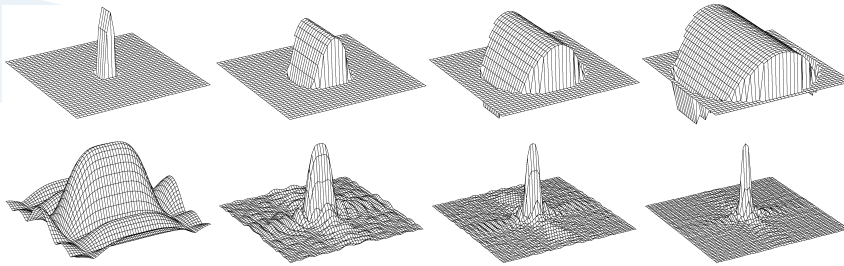


Figure 4: Function estimation *LPA* kernels obtained for $m = (2, 1)$ using the characteristic function of disk as the window function w_h. Kernels are shown for four different values of h, from small (left) to large (right). The absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure. To improve the visualization, the vertical scale of the space-domain plots is larger for the kernels of a larger scale.

How to choose h?

Accuracy analysis of the LPA: bias and variance of the *LPA* estimates

Let the observations be of the form

$$z = y + \sigma\eta, \qquad \eta \text{ iid noise, } \eta(\cdot) \sim \mathcal{N}(0,1).$$

and consider the *LPA* estimate

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

The bias and variance (i.e. the deterministic and stochastic error) of $\hat{y}_h(x)$ are

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

$$var\{\hat{y}_h(x)\} = \sigma^2_{\hat{y}_h(x)} = E\left\{(E\{\hat{y}_h(x)\} - \hat{y}_h(x))^2\right\} =$$

$$= E\left\{(((y-z) \circledast g_h)(x))^2\right\} = \left(\sigma_z^2 \circledast g_h^2\right)(x) = \sigma^2 \int g_h^2(v)\,dv = \sigma^2 \|g_h\|_2^2.$$

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

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

Our aim is to determine h in such a way that $l_{\hat{y}_h(x)}$ is minimized: find the *optimal scale*.

Asymptotic error analysis

Let x be a fixed estimation point. We can rewrite the expression for the variance as an explicit function of the scale parameter,

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

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

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

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

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

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

Asymptotic error analysis (continued)

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

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

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

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

$$\bar{m}_{\hat{y}_h(x)} = rh^{\alpha} \|g_{\mathcal{M}}\|_1 = ah^{\alpha}, \tag{7}$$

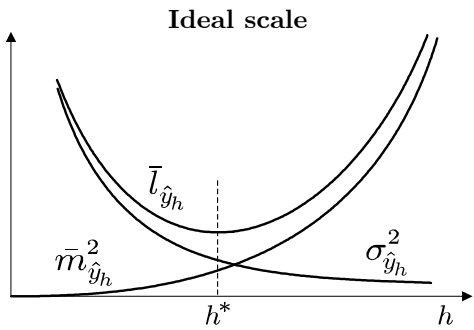
and its variance

$$\sigma^2_{\hat{y}_h(x)} = \sigma^2 h^{-d} \|g_{\mathcal{M}}\|_2^2 = b^2 h^{-2\beta}. \tag{8}$$

The upper bound of the asymptotic risk is the *ideal risk*

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

Here b and β depend only on the kernel (variance of the estimate is not affected by the estimated function), a depends on the uniform bound L, and α is influenced by the order of the *LPA* (i.e. by the polynomial components of the remainder).



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

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

and can be found by solving

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

It gives

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

and the ideal scale h^* as

$$h^* = \left(\frac{\beta b^2}{\alpha a^2}\right)^{\frac{1}{2\alpha+2\beta}}. \tag{9}$$

Let us use this ideal scale into the bias and variance expressions (7) and (8). We obtain,

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

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

$$\frac{\bar{m}^2_{\hat{y}_{h^*}(x)}}{\sigma^2_{\hat{y}_{h^*}(x)}} = a^2 \left(\frac{\beta b^2}{\alpha a^2}\right)^{\frac{\alpha}{\alpha+\beta}} b^{-2} \left(\frac{\beta b^2}{\alpha a^2}\right)^{\frac{\beta}{\alpha+\beta}} = a^2 b^{-2} \frac{\beta b^2}{\alpha a^2} = \frac{\beta}{\alpha} = \gamma^2, \tag{10}$$

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

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

$$\bar{l}_{\hat{y}_{h^*}(x)} = \sigma^2_{\hat{y}_{h^*}(x)} \left(1 + \gamma^2\right). \tag{11}$$

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

$$\bar{m}^2_{\hat{y}_h(x)} \begin{cases} < \gamma^2 \sigma^2_{\hat{y}_h(x)} & \forall h < h^* \\ > \gamma^2 \sigma^2_{\hat{y}_h(x)} & \forall h > h^* \end{cases}, \tag{12}$$

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

Adaptive scale: Intersection of Confidence Intervals (ICI) rule

The *ICI* rule (Goldenshluger&Nemirovski, 1997) is a practical method that selects an adaptive scale $h^+(x)$ whose corresponding estimate $\hat{y}_{h^+(x)}$ is close to the ideal $\hat{y}_{h^*(x)}$. This is can be done without knowledge on smoothness of the signal, i.e. without knowing L or a. Note that the ideal scale depends on these parameters:

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

Intersection of Confidence Intervals (ICI) rule: the idea (sketch of the proof)

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

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

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

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

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

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

From the inequalities (12) we obtain, for $h \leq h^*(x)$,

$$|y(x) - \hat{y}_h(x)| \leq (\gamma + \chi_{1-\lambda/2})\sigma_{\hat{y}_h(x)} = \Gamma\sigma_{\hat{y}_h(x)}, \quad \Gamma = (\gamma + \chi_{1-\lambda/2}). \tag{13}$$

Equivalently, we can express the above inequality as

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

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

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

for the estimate $\hat{y}_h(x)$: for $h \leq h^*$, with probability p, we have $y(x) \in \mathcal{D}(h)$. According to (8), the width of the confidence intervals $\mathcal{D}(h)$ is a monotonically decreasing function of h. We may say that the confidence intervals "shrink" as h increases.

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

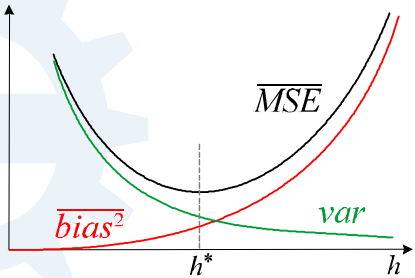
Let j^+ be the largest of those j for which all $\mathcal{D}(h_i)$ with $i \leq j$ have a point in common. Observe that $h_{j^+} \geq h^{*-} \triangleq \max\{h_j : h_j \leq h^*(x)\}$, i.e. all $\mathcal{D}(h_j)$ with $h_j \leq h^{*-}$ have nonempty intersection. This condition, together with the shrinking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

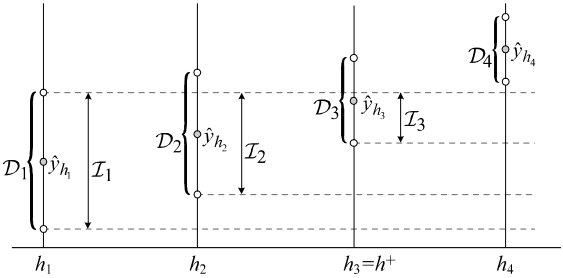
Indeed, $y(x) \in \bigcap_{h_j \leq h^{*-}} \mathcal{D}(h_j)$, hence $|y(x) - \hat{y}_{h^{*-}}(x)| \leq \Gamma\sigma_{\hat{y}_{h^{*-}}(x)}$. Similarly, since $\mathcal{D}(h^{*-}) \cap \mathcal{D}(h^+) \neq \varnothing$, $|\hat{y}_{h^{*-}}(x) - \hat{y}_{h^+}(x)| \leq \Gamma\sigma_{\hat{y}_{h^{*-}}(x)}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

$$|y(x) - \hat{y}_{h^+}(x)| \leq 2\Gamma\sigma_{\hat{y}_{h^{*-}}(x)} + \Gamma\sigma_{\hat{y}_{h^+}(x)} \leq 3\Gamma\sigma_{\hat{y}_{h^{*-}}(x)}. \tag{14}$$

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

Intersection of confidence intervals (*ICI*) rule





The estimates $\hat{y}_h(x)$ are calculated for a set $H = \{h_j\}_{j=1}^J$ of increasing scales. The *ICI* rule yields a pointwise adaptive estimate $\hat{y}_{h^+}(x)$, where for every x an adaptive scale $h^+(x) \in H$ is used. We have that $\hat{y}_{h^+}(x) \approx \hat{y}_{h^*(x)}(x)$, in practice one can think that $h^+(x) \approx h^*(x)$.

The *ICI* rule is as follows: *Consider the intersection of confidence intervals*

$$\mathcal{I}_j = \bigcap_{i=1}^{j} \mathcal{D}_i, \quad \text{where} \quad \mathcal{D}_i = \left[\hat{y}_{h_i}(x) - \Gamma\sigma_{\hat{y}_{h_i}}, \hat{y}_{h_i}(x) + \Gamma\sigma_{\hat{y}_{h_i}}\right]$$

and $\Gamma > 0$ *is a threshold parameter, and let* j^+ *be the largest of the indexes* j *for which* $\mathcal{I}_j$ *is non-empty,* $\mathcal{I}_{j^+} \neq \varnothing$ *and* $\mathcal{I}_{j^+ + 1} = \varnothing$. *Then,* h^+ *is defined as* $h^+ = h_{j^+}$ *and the adaptive estimate is* $\hat{y}_{h^+}(x)$.

Pseudo-code of the ICI adaptive-scale selection algorithm

$h^+ \equiv h_1$
$\hat{y}_{h^+} = \hat{y}_{h_1}$ } / initialization of adaptive scale and of corresponding estimate and variance

$U = \hat{y}_{h_1} + \Gamma\sigma_{\hat{y}_{h_1}}$
$L = \hat{y}_{h_1} - \Gamma\sigma_{\hat{y}_{h_1}}$ } / initialization of upper and lower bounds of intersection

`for` $j = 2, \ldots, J$ / loop on j (scale index)

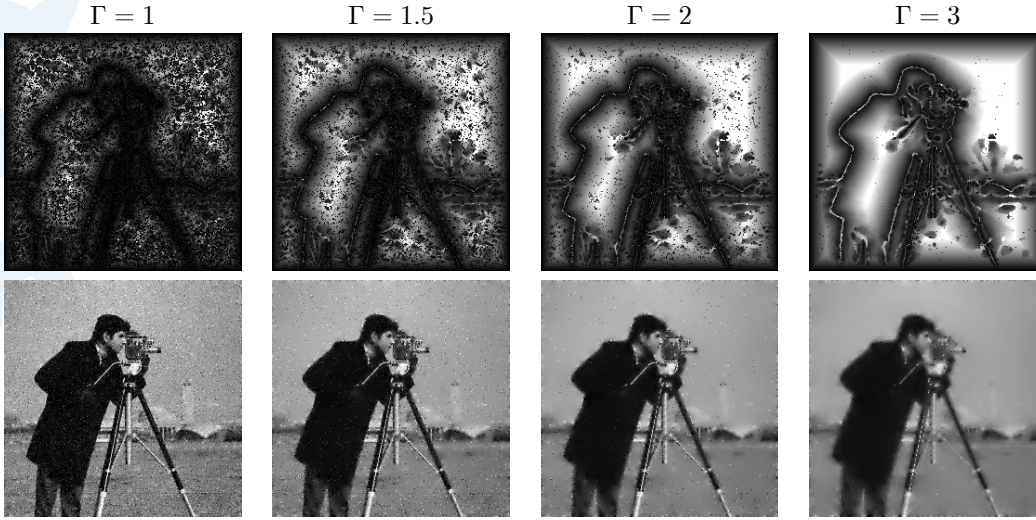
$U = \min\{U, \hat{y}_{h_j} + \Gamma\sigma_{\hat{y}_{h_j}}\}$
$L = \max\{L, \hat{y}_{h_j} - \Gamma\sigma_{\hat{y}_{h_j}}\}$ } / update bounds of intersection

$T = U \geq L$ / test for non-empty intersection

$h^+ = h_j T + h^+ \mathrm{NOT}(T)$
$\hat{y}_{h^+} = \hat{y}_{h_j} T + \hat{y}_{h^+} \mathrm{NOT}(T)$ } / update adaptive scale and adaptive-scale estimate

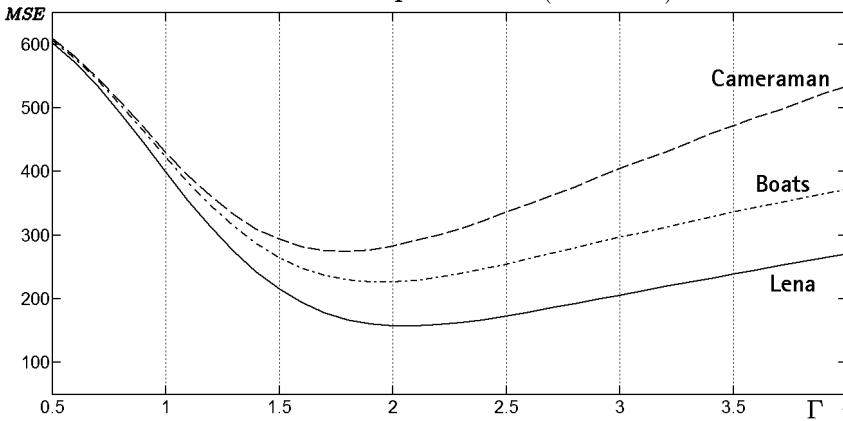
`end` / end loop on j (scale index)

ICI: choice of Γ parameter



Adaptive scales and adaptive-scale estimates obtained for different values of Γ. The adaptive scales are represented using a darker shade of gray for the smaller scales, black being the smallest scale (which corresponds to a Dirac-delta estimate), and white being the maximum scale (corresponding to a kernel whose support is a disc of radius 35 pixels).

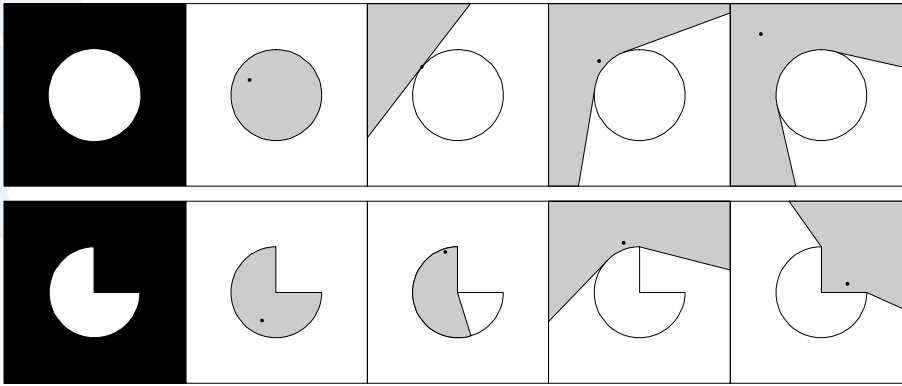
ICI: choice of Γ parameter (continued)



"*MSE* vs. Γ" plots for three different images ($\sigma = 25$). The same kernels used for the experiment shown in the previous figure are used.

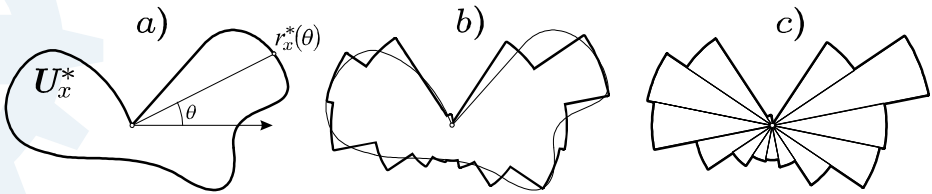
The plots show that for the three images the best found values of Γ are all close to 2, and that variations of ± 0.2 around these best found values do not affect the objective quality of reconstruction. Therefore, with this family of kernels, $\Gamma = 2$ can be used for all the images achieving a performance close to the one achieved with an "oracle" Γ.

Anisotropy: motivation



In some cases the geometry of symmetric kernels is not sufficient to adapt to the image structure. Goal: adapt to the image using approximations of starshaped supports.

Anisotropic estimator based on directional adaptive-scale: idea



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

Directional *LPA*

The window is characterized by a direction θ and is denoted as w_θ.

The polynomials are expressed with respect to a θ-rotated coordinate system:

$$\mathcal{M} = \left\{ \varphi : \varphi(u_1, u_2) = \sum_{i,j}^{m} c_{i,j} u_1^i u_2^j \right\},$$

$$(u_1, u_2) = (v_1 \cos\theta + v_2 \sin\theta, v_2 \cos\theta - v_1 \sin\theta) = \mathbf{U}_\theta v.$$

Typically, w_θ is obtained by rotating a "basic" window $w = w_0$ through an angle θ, $w_\theta = w(\mathbf{U}_\theta v)$. When also a scale parameter h is exploited, the resulting estimates and kernels are denoted as $w_{h,\theta}$, $g_{h.\theta}$, respectively.

Directional *LPA*: examples of directional kernels

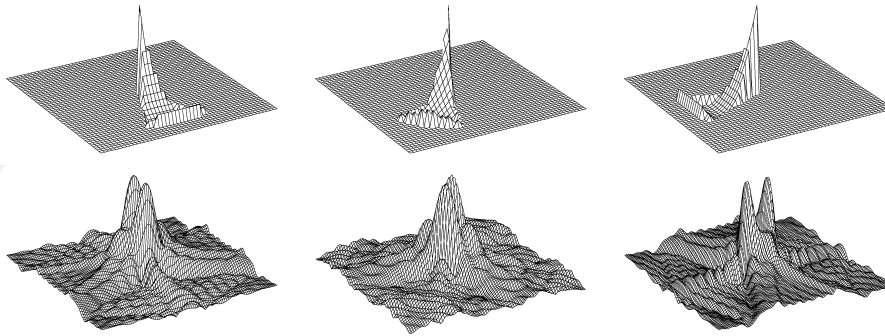
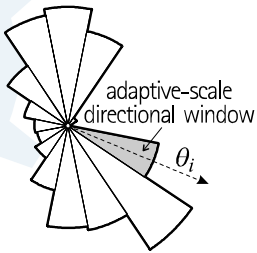
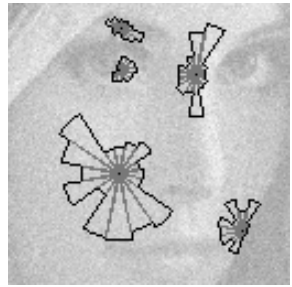
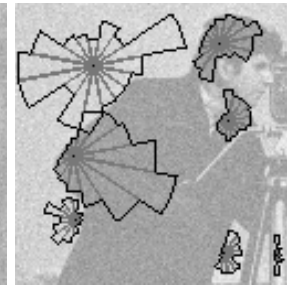


Figure 5: Function estimation directional *LPA* kernels, $o = (0, 0)$, obtained for $m = (2, 1)$ using the characteristic function of a conical sector as the window function w_θ. Kernels are shown for three different direction θ; the absolute value of the kernels' Fourier transforms are shown in the bottom row of the Figure.

Anisotropic *LPA-ICI*

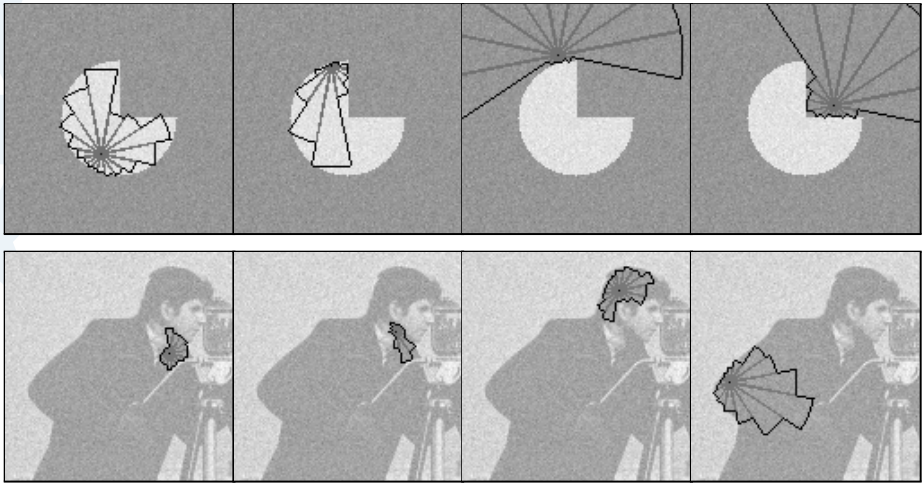






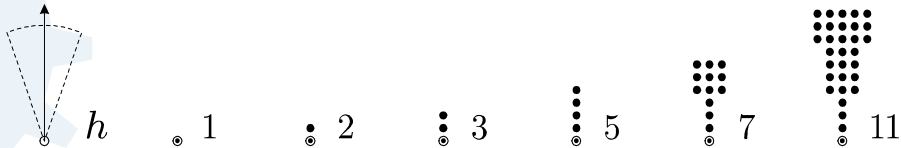
For each specified direction θ_i, the *ICI* rule is exploited in order to find the adaptive-scale directional-*LPA* estimates $\hat{y}_{h^+(x,\theta_i),\theta_i}(x)$.

Anisotropic *LPA-ICI*: examples

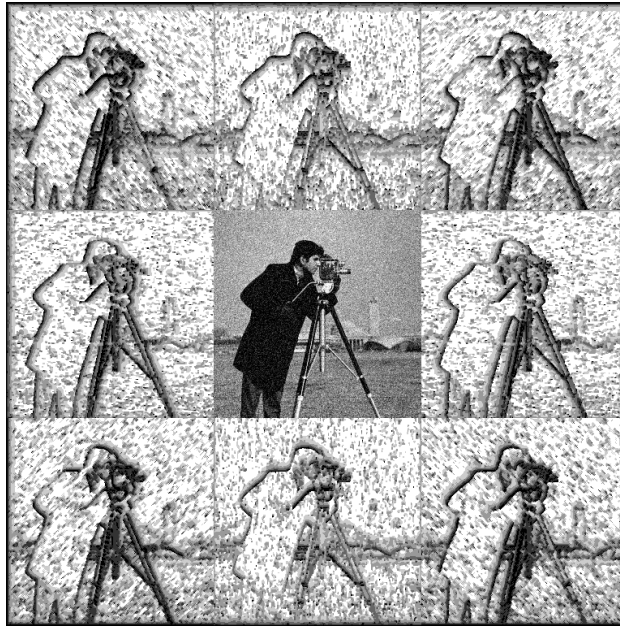


Cheese and *Cameraman* (detail): adaptive anisotropic neighborhoods $\tilde{U}_x^+$ obtained through *ICI* using sectorial kernels.

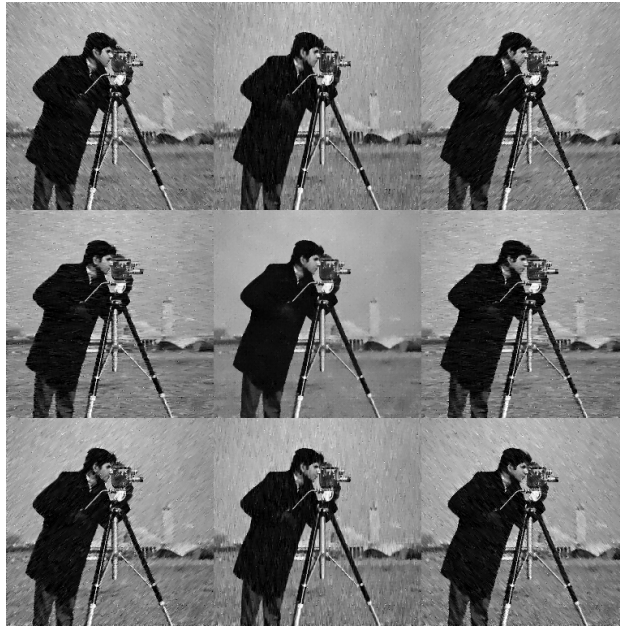
Anisotropic *LPA-ICI*: kernel supports used in the practice



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



Smaller scales are represented using a darker shade of gray.



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

Anisotropic *LPA-ICI*: fusing

The adaptive-scale directional estimates $\hat{y}_{h^+(x,\theta_i),\theta_i}(x)$ are "fused" into the final *anisotropic estimate* $\hat{y}$ by the convex linear combination

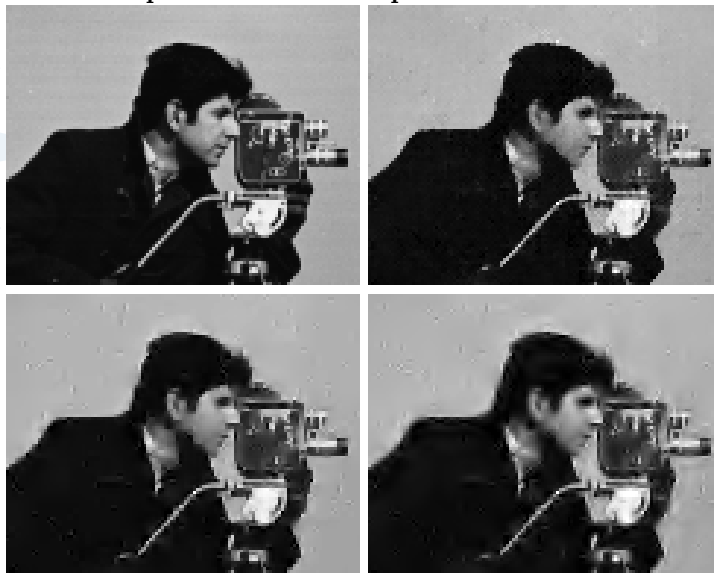
$$\begin{aligned} \hat{y}(x) &= \sum\nolimits_i \lambda(x,\theta_i)\, \hat{y}_{h^+(x,\theta_i),\theta_i}(x), \\ \lambda(x,\theta_i) &= \sigma^{-2}_{\hat{y}_{h^+(x,\theta_i),\theta_i}(x)} \Big/ \sum\nolimits_j \sigma^{-2}_{\hat{y}_{h^+(x,\theta_j),\theta_j}(x)}, \end{aligned} \tag{15}$$

where the inverse of the variance of the adaptive estimates is used as the weighting factor.

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

Criteria values for the denoising of the *Cameraman* image using 8 directional adaptive estimates. The fused estimate is much better than each of the directional ones.

Anisotropic *LPA-ICI*: comparison with TI wavelets



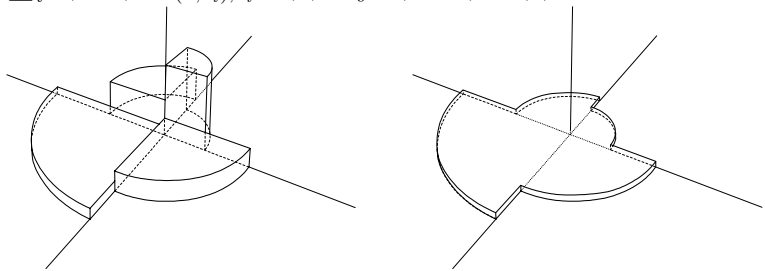
Noisy image, not shown, $\sigma = 255/10$. Clockwise from top-left: original image, anisotropic *LPA-ICI* estimate, *ISNR*=8.1dB, translation-invariant (TI) Daubechies Db4 wavelets (oracle threshold), *ISNR*=7.4dB, TI Haar wavelets (oracle threshold), *ISNR*=7.8dB.

Fusing: why σ^{-2}?

1. for uniform kernels, the fused estimate is exactly the average of the signal over the anisotropic adaptive neighborhood $\tilde{U}_x^+$ (as shown in figure below);
2. assuming that the directional adaptive estimates are independent and unbiased, such fusing is the maximum-likelihood estimate of $y(x)$ given $\left\{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)\right\}_{k=1}^{K}$.

(1.) Uniform anisotropic kernel

The anisotropic estimate can also be obtained as the integration against the anisotropic kernel $g_x^+ = \sum_i \lambda(x,\theta_i) g_{h^+(x,\theta_i),\theta_i}$, $\hat{y}(x) = \int z(x-v)\, g_x^+(v)\, dv$. This is **not** a convolution.



Directional *LPA* (uniform w and $m=0$) adaptive kernels $g_{h^+(x,\theta_i),\theta_i}$ and the resulting fused anisotropic kernel g_x^+.

Fusing: why σ^{-2}? (2.) ML fusing

Let us assume that the directional adaptive estimates $\hat{y}_{h^+(x,\theta_k),\theta_k}(x)$, $k = 1, \ldots, K$, are independent and unbiased. This latter condition is generally never satisfied. However, when combining multidirectional estimates, the choice of the Γ parameter (in ICI) is such that the resulting bias is much smaller than the variance. Independency is guaranteed if the kernels supports are non-overlapping.

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

$$\begin{aligned}
\text{Log-likelihood} \quad L &= \ln \prod_k \left(2\pi\sigma_k^2\right)^{-\frac{1}{2}} e^{-\frac{1}{2\sigma_k^2}\left(\hat{y}_{h^+(x,\theta_k),\theta_k}(x) - y(x)\right)^2} = \\
&= \sum_k -\frac{1}{2}\sigma_k^{-2}\left(\hat{y}_{h^+(x,\theta_k),\theta_k}(x) - y(x)\right)^2 + \ln\left(\left(2\pi\right)^{-1/2}\sigma_k^{-1}\right), \\
&\qquad \text{where} \quad \sigma_k^2(x) \triangleq \sigma^2_{\hat{y}_{h^+(x,\theta_k),\theta_k}(x)}.
\end{aligned}$$

ML fusing (continued)

Differentiating L with respect to y, we obtain

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

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

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

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

Deconvolution (deblurring): observation model

We wish to recover an image y from blurred and noisy observations

$$z = (v \circledast y) + \sigma\eta,$$

where v is the point-spread function (PSF) of the optical system. For simplicity, it is assumed that the PSF is known and that the noise η is standard Gaussian.

In the frequency domain the observation equation has the form

$$Z = YV + \sigma\eta, \tag{16}$$

where capital letters are used for the discrete Fourier transform of the corresponding variables. We assume an orthonormal Fourier transform.

Deconvolution (deblurring): observation example

Original *Cameraman* image (left) and noisy blurred observation (9×9 boxcar v, BSNR=40dB) (right)

Anisotropic LPA-ICI RI-RWI deconvolution algorithm: directional estimates

The considered technique is based on the following regularized inversion (RI) and regularized Wiener inversion (RWI) algorithms, using the directional-LPA kernels g_{h,θ_k}:

$$\hat{Y}^{RI}_{h,\theta_k} = \frac{\overline{V}G_{h,\theta_k}}{|V|^2+\varepsilon_1^2}Z, \qquad (RI), \tag{17}$$

$$\hat{Y}^{RWI}_{h,\theta_k} = \frac{\overline{V}|Y|^2G_{h,\theta_k}}{|VY|^2+\varepsilon_2^2\sigma^2}Z, \qquad (RWI), \tag{18}$$

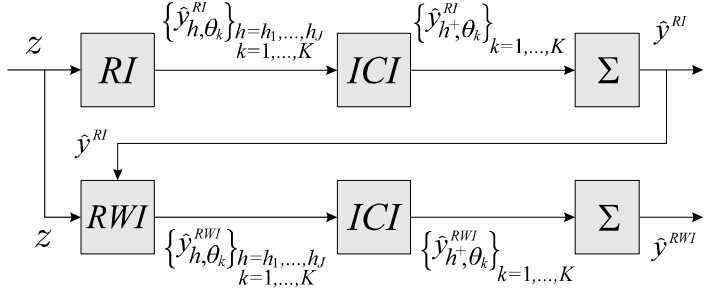
where $\varepsilon_1, \varepsilon_2 > 0$ are regularization parameters. Equivalently, using the standard convolutions in space domain:

$$\hat{y}^{RI}_{h,\theta_k} = \mathcal{F}^{-1}\left(\frac{\overline{V}Z}{|V|^2+\varepsilon_1^2}\right) \circledast g_{h,\theta_k}, \qquad (RI),$$

$$\hat{y}^{RWI}_{h,\theta_k} = \mathcal{F}^{-1}\left(\frac{\overline{V}|Y|^2Z}{|VY|^2+\varepsilon_2^2\sigma^2}\right) \circledast g_{h,\theta_k}, \qquad (RWI).$$

The final estimate of y is given as an anisotropic RWI estimate which uses an anisotropic RI estimate as the reference signal Y in (18). Thus, we arrive to a two steps procedure (see figure in the next slide).

Anisotropic LPA-ICI RI-RWI deconvolution algorithm: flowchart



(1) Estimates $\{\hat{y}^{RI}_{h,\theta_k}\}_{h\in H}$ are calculated according to (17) for a set of scales H and *ICI* selects the best scales for each direction and for each pixel. In this way, we obtain the directional adaptive-scale estimates $\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}$, $k = 1, \dots, K$, which are fused into anisotropic $\hat{y}^{RI}$ according to (15) (i.e., inverse variances).

(2) The $\hat{y}^{RI}$ serves as the reference signal in the *RWI* procedure. The adaptive *RWI* algorithm is similar and gives the *ICI* adaptive varying scales estimates $\hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}$ for each direction and x. Then, the final estimate $\hat{y}^{RWI}$ is obtained by fusing these directional ones again according to (15).

Anisotropic LPA-ICI RI-RWI deconvolution algorithm: ICI

ICI rule requires the standard deviations of the individual varying scale directional estimates $\{\hat{y}^{RI}_{h,\theta_k}\}_{h\in H}$ and $\{\hat{y}^{RWI}_{h,\theta_k}\}_{h\in H}$. These standard deviations can be easily calculated by the l^2-norm of the frequency response of the corresponding filters:

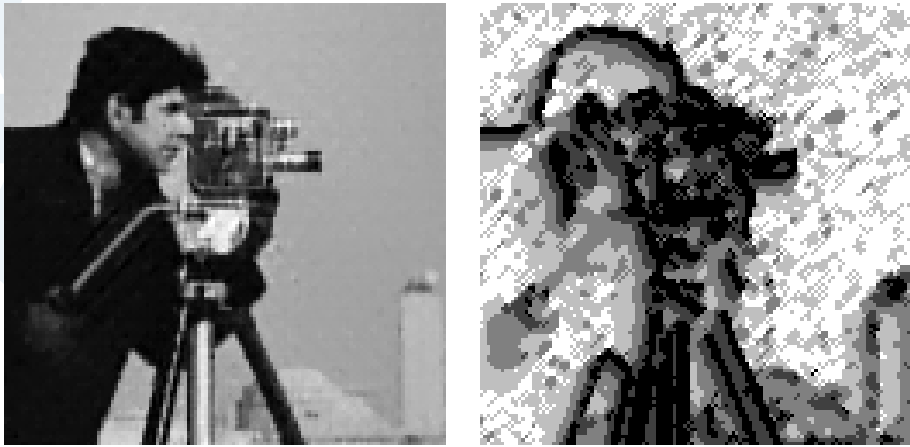
$$\begin{aligned}\sigma_{\hat{y}^{RI}_{h,\theta_k}} &= \sigma \left\| \frac{\overline{V} G_{h,\theta_k}}{|V|^2+\varepsilon_1^2} \right\|_2, \\ \sigma_{\hat{y}^{RWI}_{h,\theta_k}} &= \sigma \left\| \frac{\overline{V}|Y|^2 G_{h,\theta_k}}{|VY|^2+\varepsilon_2^2\sigma^2} \right\|_2 .\end{aligned}$$

Fusing: RI

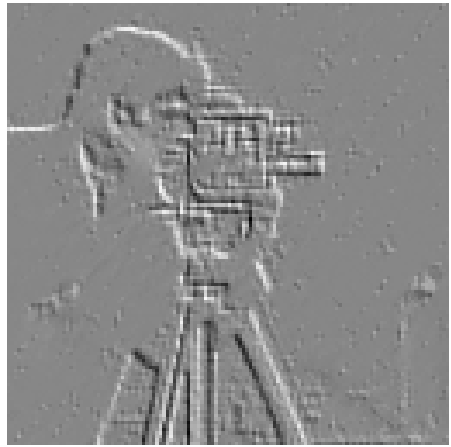
$$\begin{aligned}\hat{y}^{RI}(x) &= \sum\nolimits_k \lambda_k^{RI}(x)\hat{y}^{RI}_{h^+(x,\theta_k),\theta_k}(x), \\ \lambda_k^{RI}(x) &= \sigma_k^{RI-2}(x)/\textstyle\sum_i \sigma_i^{RI-2}(x), \\ \sigma_i^{RI-2}(x) &= 1/\sigma^2_{\hat{y}^{RI}_{h^+(x,\theta_i),\theta_i}}(x).\end{aligned}$$

Fusing of RWI estimates $\hat{y}^{RWI}_{h^+(x,\theta_k),\theta_k}$ is analogous.

Anisotropic LPA-ICI RI-RWI deconvolution algorithm: result

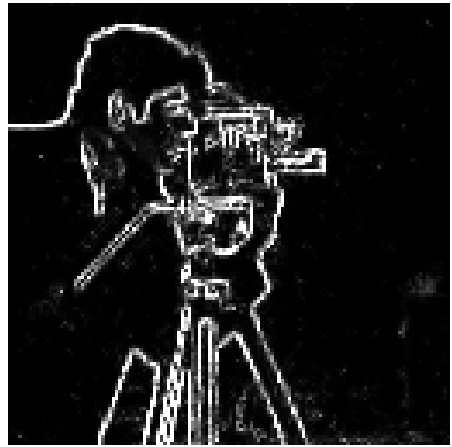


Anisotropic RWI estimate $\hat{y}^{RWI}$ (final restored image), $ISNR$=8.23dB (left) and adaptive scales scales $h^{+}(\cdot, \pi/4)$ (right). Smaller scales are darker. The ICI adaptive scales $h^{+}(\cdot, \theta_k)$ represent the distribution of image features across the direction θ_k.



Let us replace in the *RWI* stage of the algorithm (18) the function estimation (smoothing) kernels by the derivative-estimation kernels. Then, the ICI output $\hat{y}^{RWI}_{h^+,\theta_k}$ is the estimate of the directional right-hand derivative $\partial_{+\theta_k} y$. The figure shows the diagonal derivative estimate $\hat{\partial}_{\theta_2}$ calculated for $\theta = \pi/4$ as the mean of the two one-sided directional derivatives with $\theta_2 = \pi/4$ and $\theta_5 = \theta_2 + \pi = 5\pi/4$, $\hat{\partial}_{\theta_2} = (\hat{y}^{RWI}_{h^+,\theta_2} - \hat{y}^{RWI}_{h^+,\theta_5})/2$.

Edge detection from noisy blurred observations



Further, for the edge detection, we calculate the sum of the absolute values of the directional derivative estimates $\sum_{k=1}^{4} |\hat{\partial}_{\theta_k}|$.

PART 2

Pointwise Shape-Adaptive DCT filters: idea and motivation

“ Use the adaptive neighborhoods $\tilde{U}_x^+$ defined by the Anisotropic LPA-ICI as supports for some transform. ”

By demanding the local fit of a polynomial model, we are able to avoid the presence of singularities or discontinuities within the transform support. In this way, we ensure that data are represented sparsely in the transform domain, significantly improving the effectiveness of shrinkage (e.g., thresholding).

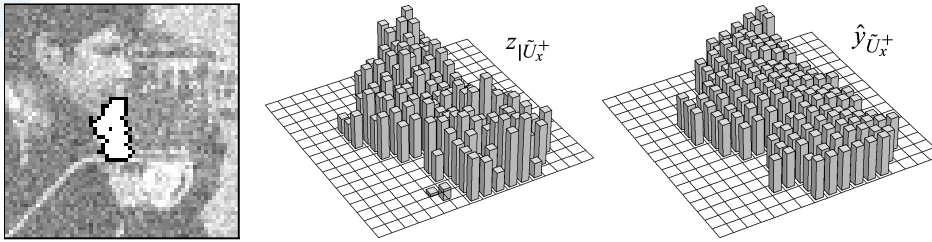
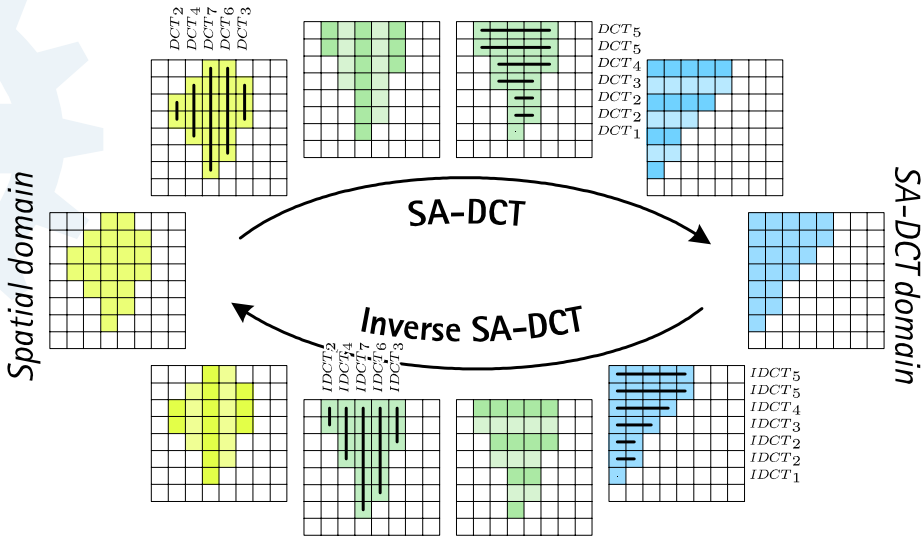


Figure 6: From left to right: a detail of the noisy *Cameraman* showing an adaptive-shape neighborhood $\tilde{U}_x^+$ determined by the Anisotropic LPA-ICI procedure, and the image intensity corresponding to this region before and after hard-thresholding in SA-DCT domain.



Shape-Adaptive Discrete Cosine Transform (SA-DCT) and its inverse. Transformation is computed by cascaded application of one-dimensional varying-length DCT transforms, along the columns and along the rows.

Shape-Adaptive Discrete Cosine Transform (SA-DCT)

- direct generalization of the classical block-DCT (B-DCT);
- on rectangular domains (e.g., squares) the SA-DCT and B-DCT coincide;
- the same computational complexity as the B-DCT (separable);
- SA-DCT is part of the MPEG-4 standard;
- efficient (low-power) hardware implementations available;

- shape must be coded separately (constitutes some overhead);
- before 2005, it had been used only for image and video compression.

Orthonormal SA-DCT

The orthonormalization of the SA-DCT is obtained by normalization of the individual one-dimensional transforms that are used for transforming the columns and rows. In terms of their basis elements, they are defined as:

$$\psi_{L,m}^{\text{1D-DCT}}(n) = c_m \cos\left(\frac{\pi m(2n+1)}{2L}\right), \quad m,n=0,\ldots,L-1, \tag{19}$$

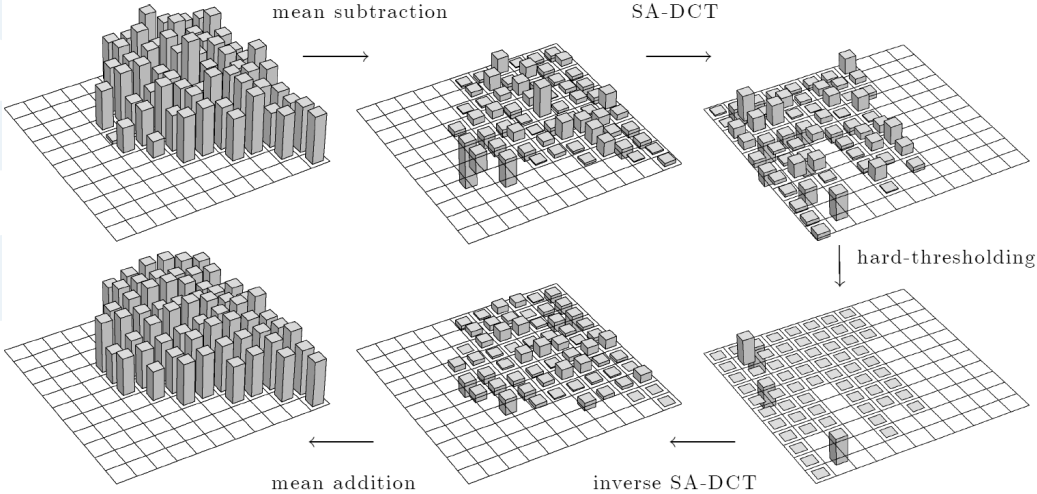
$$c_0 = \sqrt{1/L}, \quad c_m = \sqrt{2/L},\ m>0. \tag{20}$$

Here L stands for the length of the column or row to be transformed. A different normalization of the 1D transforms would produce, on an arbitrary shape, a 2D transform that is non-orthogonal.

Notation

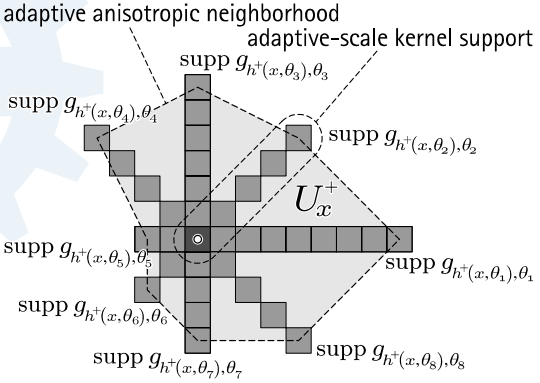
We denote by $T_U : \mathcal{U} \to \mathcal{V}_U$ the orthonormal SA-DCT transform obtained for a region $U \subset X$, where $\mathcal{U} = \{f : U \to \mathbb{R}\}$ and $\mathcal{V}_U = \{\varphi : V_U \to \mathbb{R}\}$ are function spaces and $V_U \subset \mathbb{Z}^2$ indicates the domain of the transform coefficients. Let $T_U^{-1} : \mathcal{V}_U \to \mathcal{U}$ be the inverse transform of T_U. We indicate the thresholding (or quantization) operator as Υ. Thus, the SA-DCT-domain processing of the observations z on a region U can be written as $\hat{y}_U = T_U^{-1}\left(\Upsilon\left(T_U\left(z_{|U}\right)\right)\right)$, $\hat{y}_U : U \to \mathbb{R}$.

Hard-thresholding in SA-DCT domain: DC separation



The image data on an arbitrarily shaped region is subtracted of its mean. The zero-mean data is then transformed and thresholded. After inverse transformation, the mean is added back.

(Orthonormal SA-DCT with DC separation and ΔDC compensation, Kauff et al. 1997)



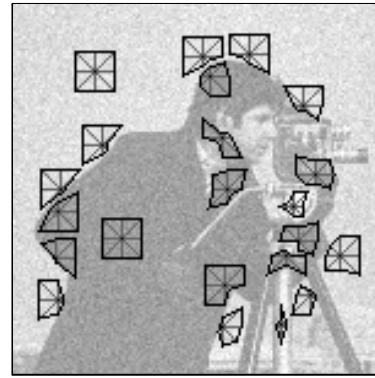


Figure 7: "Linewise" one-dimensional directional LPA kernels are used for 8 directions. The anisotropic neighborhood U_x^+ is constructed as the polygonal hull of the adaptive-scale kernels' supports (left). Thus, only the adaptive scales h^+ are needed to construct the neighborhood. Some examples of the anisotropic neighborhoods $\tilde{U}_x^+$ used for SA-DCT filtering of the noisy *Cameraman* image (right), σ=25. In our implementation we use $h \in H = \{1, 2, 3, 5, 7, 9\}$.

Pointwise SA-DCT: local estimates (hard-thresholding)

For every point $x \in X$, we take the anisotropic neighborhood $\tilde{U}_x^+$ and construct a *local* estimate $\hat{y}_{\tilde{U}_x^+} : \tilde{U}_x^+ \to \mathbb{R}$ of the signal y by thresholding in SA-DCT domain

$$\hat{y}_{\tilde{U}_x^+} = T^{-1}_{\tilde{U}_x^+}\left(\Upsilon_x\left(\varphi_{z,x}\right)\right) + m_{\tilde{U}_x^+}(z), \tag{21}$$

where the transform-domain coefficients $\varphi_{z,x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ are calculated as

$$\varphi_{z,x} = T_{\tilde{U}_x^+}\left(z_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right), \tag{22}$$

and Υ_x is a hard-thresholding operator based on the threshold

$$\sigma\sqrt{2\ln|\tilde{U}_x^+| + 1}. \tag{23}$$

This threshold is essentially Donoho's "universal" threshold.

Overcompleteness

Anisotropic neighborhoods corresponding to nearby points are usually overlapping, and since the SA-DCT is a complete system (basis) for an individual support $\tilde{U}_x^+$, the overall approach is obviously overcomplete. As a consequence, different local estimates $\hat{y}_{\tilde{U}_{x'}^+}$ and $\hat{y}_{\tilde{U}_{x''}^+}$ do not coincide where they overlap (i.e., on $\tilde{U}_{x'}^+ \cap \tilde{U}_{x''}^+$).

In order to obtain a single *global* estimate $\hat{y} : X \to \mathbb{R}$ defined on the whole image domain, all the local estimates (21) are averaged together using adaptive weights $w_x \in \mathbb{R}$ in the following convex combination:

$$\hat{y} = \frac{\sum_{x \in X} w_x \hat{y}_{\tilde{U}_x^+}|^X}{\sum_{x \in X} w_x \chi_{\tilde{U}_x^+}}, \tag{24}$$

$$w_x = \frac{\sigma^{-2}}{(1 + N_x^{\mathrm{har}})\,|\tilde{U}_x^+|}, \tag{25}$$

where N_x^{har} is the number of non-zero coefficients after thresholding (so-called "number of harmonics").

Here, $\sigma^2 \left(1 + N_x^{\mathrm{har}}\right)$ is an estimate of the total sample variance of $\hat{y}_{\tilde{U}_x^+}$, given as sum of variances of the transform coefficients which are used for reconstruction.

These weights favour estimates which correspond to sparser representations.

Pointwise SA-DCT: local estimates (Wiener filtering)

Using the same approach as for thresholding, we introduce an empirical Wiener filter in the SA-DCT domain. It assumes that an estimate $\hat{y}$ of y is known. In practice, we obtain this estimate using the above hard-thresholding technique.

For every $x \in X$, let $\varphi_{\hat{y},x} : V_{\tilde{U}_x^+} \to \mathbb{R}$ be the SA-DCT (on $\tilde{U}_x^+$) coefficients of $\hat{y}$ where the mean $m_{\tilde{U}_x^+}(z)$ of z is subtracted before applying the transform:

$$\varphi_{\hat{y},x} = T_{\tilde{U}_x^+}\left(\hat{y}_{|\tilde{U}_x^+} - m_{\tilde{U}_x^+}(z)\right). \tag{26}$$

The *local* Wiener estimate $\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}$ is defined as

$$\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+} = T^{-1}_{\tilde{U}_x^+}\left(\omega_x \varphi_{z,x}\right) + \varpi_x m_{\tilde{U}_x^+}(z), \tag{27}$$

where the SA-DCT coefficients $\varphi_{z,x}$ of z are calculated as in (22), and $\omega_x \in \mathcal{V}_{\tilde{U}_x^+}$ and $\varpi_x \in \mathbb{R}$ are respectively the Wiener attenuation factors for $\varphi_{z,x}$ and for the subtracted mean value $m_{\tilde{U}_x^+}(z)$,

$$\omega_x = \frac{\varphi^2_{\hat{y},x}}{\varphi^2_{\hat{y},x} + \sigma^2}, \qquad \varpi_x = \frac{m^2_{\tilde{U}_x^+}(\hat{y})}{m^2_{\tilde{U}_x^+}(\hat{y}) + \sigma^2 / |\tilde{U}_x^+|}. \tag{28}$$

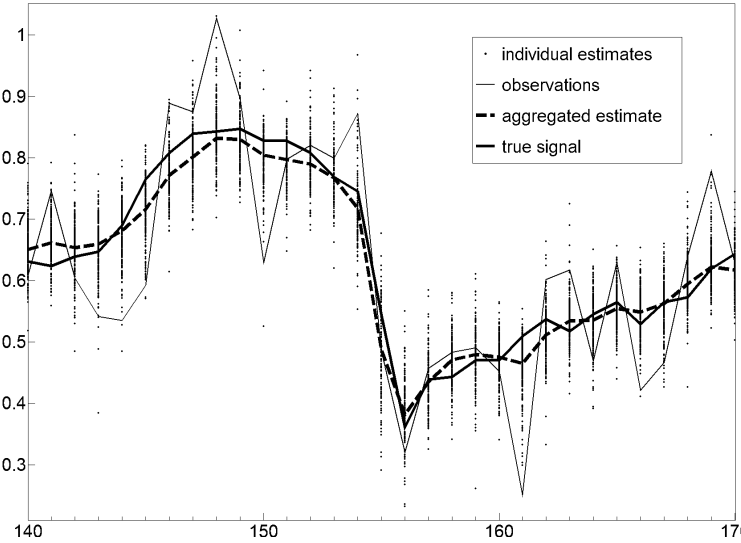
Pointwise SA-DCT: global estimate as aggregation of local Wiener estimates

The *global* estimate $\hat{y}^{\mathrm{wi}}$ can be obtained analogously as in (24), using the convex combination with the adaptive weights w_x^{wi}:

$$\begin{aligned} \hat{y}^{\mathrm{wi}} &= \frac{\sum_{x \in X} w_x^{\mathrm{wi}} \hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+} |^X}{\sum_{x \in X} w_x^{\mathrm{wi}} \chi_{\tilde{U}_x^+}}, \\ w_x^{\mathrm{wi}} &= \frac{\sigma^{-2}}{(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2)|\tilde{U}_x^+|}. \end{aligned}$$

Again, the term $\sigma^2\big(\varpi_x^2 + \sum_{V_{\tilde{U}_x^+}} \omega_x^2\big)$ in the adaptive weights corresponds to an estimate of the total sample variance of $\hat{y}^{\mathrm{wi}}_{\tilde{U}_x^+}$.

Pointwise SA-DCT: aggregation in action, an illustration



Details of a cross-section of length 31 pixels from the *Peppers* image (σ=25): the dots show all the individual estimates which are aggregated in order to obtained the final estimates at each position. For each pixel there are about 200 individual estimates.

Pointwise SA-DCT: denoising results

A fragment of *Lena*: original, noisy observation (σ=25, PSNR=20.18dB), BLS-GSM estimate (Portilla et al.) (PSNR=31.69dB), and the proposed Pointwise SA-DCT estimate (PSNR=31.66).

Pointwise SA-DCT: denoising results

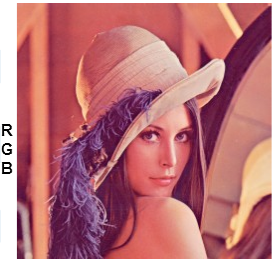
A fragment of *Cameraman*: noisy observation (σ=25, PSNR=20.14dB), BLS-GSM estimate (Portilla et al.) (PSNR=28.35dB), and the proposed Pointwise SA-DCT estimate (PSNR=29.11dB).

Pointwise SA-DCT: deblocking results

JPEG

$\hat{y}^{\mathrm{wi}}$

Pointwise SA-DCT: extension to color, motivation

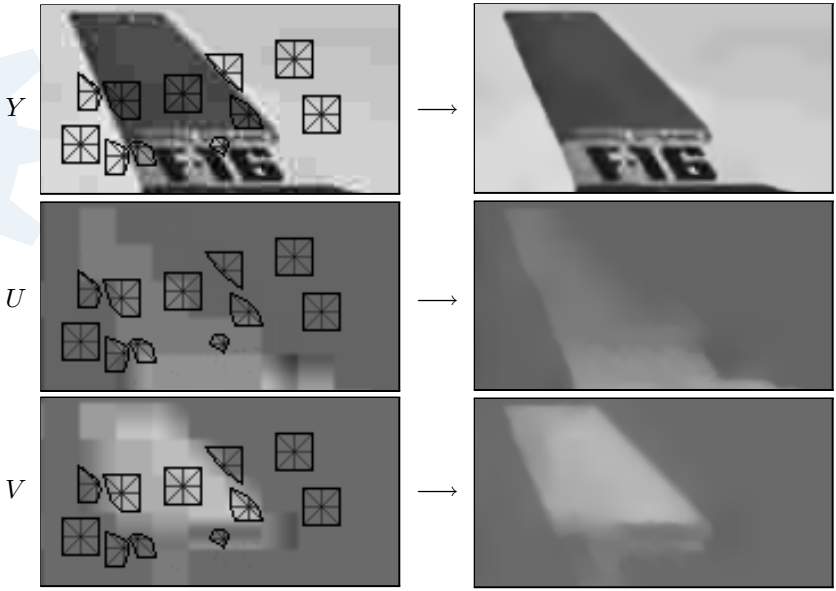


color transformation

$$\mathbf{A}_{opp} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{\sqrt{6}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{1}{3\sqrt{2}} & \frac{-\sqrt{2}}{3} & \frac{1}{3\sqrt{2}} \end{bmatrix}$$

Luminance-chrominance decompositions: "structural correlation"

Thus, separate denoising of the three channels (even luminance-chrominance channels) is often inadequate, and obviously far from optimal.



Use for all three channels the adaptive neighborhoods defined by the anisotropic LPA-ICI for the luminance channel.

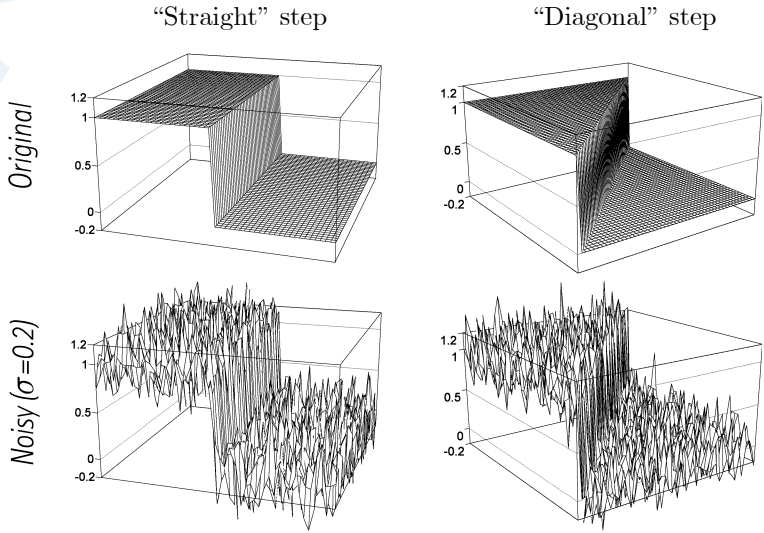
Fragments of the JPEG-compressed (Q=10, 0.25bpp, PSNR=26.87dB), and restored *F-16* color image (PSNR=28.30dB) using the proposed Pointwise SA-DCT deblocking filter in luminance-chrominance space.

Pointwise SA-DCT: denoising results

Fragments of the noisy *F-16* (σ=30, PSNR=18.59dB), of ProbShrink-MB (Pizurica et al.) estimate (PSNR=30.50dB), and of Pointwise SA-DCT estimate (PSNR=31.59dB).

Pointwise SA-DCT: block-DCT vs. shape-adaptive DCT

How important is the shape-adaptive transform?
Let us replace SA-DCT with square block-DCT with fixed or adaptive block-size.



Fixed B-DCT 16x16
Adaptive-size B-DCT
Pointwise SA-DCT
1.2
1
0.5
0
-0.2

"Shape-Adapted" transforms: Gram-Schmidt orthonormalization

- Proposed by Gilge (1989): "shape-adapted" DCT or "shape-adapted" polynomials
- Better decorrelating performance than basic SA-DCT of Sikora (1995)
- Slighly better than o.n. SA-DCT with DC separation of Kauff (1997)
- Computationally very demanding (requires orthonormalization, is not separable, implemented as matrix multiplication)
- Not obvious what to do: elements are linearly dependent, which to discard (prune), how to order? (excellent study by W. Philips, 1993)

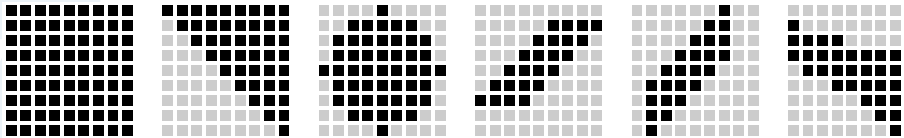
Procedure

Start with a conventional basis (e.g., DCT basis) defined on a rectangular superset of the arbitrary-shape domain of interest.
Restrict the basis elements on the domain of interest (drop samples outside domain).
Sort basis elements (e.g., zig-zag).
Prune restricted basis to remove linearly dependent elements.
Orthonormalize. E.g., Gram-Schmidt QR-decomposition: the starting basis, after restriction and pruning, is represented as a matrix A whose column vectors are the individual elements (vectorized, e.g., by raster scan of the support) and Q and R are, respectively, an orthogonal and an upper-triangular matrix, such that $QR = A$. Depending on the particular basis, there may exist faster orthonormalization procedures than G.-S. (e.g., for polynomials see W. Philips, 1997).

Examples and comparison of basis elements

We compare the SA-DCT basis elements with the basis elements of the "shape-adapted" DCT and "shape-adapted" polynomials.

Supports



Six different supports, each of which is a subset of the 9×9 square. The coordinates within the 9×9 square support are denoted as (n_1, n_2), $n_1, n_2 = 0, \ldots, 8$. Thus, the centre will have coordinates $(4, 4)$.

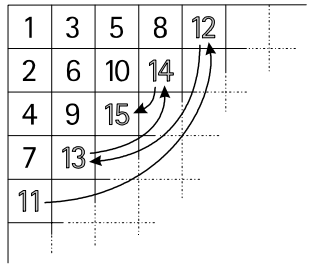
Arrangement of the figures (rows)

For each shape, the figures of the basis elements are arranged as in a long horizontal table (spanning four pages), where the rows are organized as follows:

indicators (in transform domain)
polynomial generators
cosine (DCT) generators
shape-adapted polynomials
shape-adapted DCT
shape-adaptive DCT (SA-DCT)

Arrangement of the figures (columns)

In each column, the SA-DCT basis element is obtained by applying the inverse SA-DCT on the indicator. Whereas for the "shape-adapted" bases, the indicator coincides with that of the last generator used in the linear combination that yields the orthonormalized basis element. In accordance with the "zig-zag" order, these generators are arranged column by column in the following manner (for polynomials and DCT) (next page):



Polynomial and cosine generators (sorted according to zig-zag order)

1	$n_1 - 4$	$n_2 - 4$
1	$\cos\left(\frac{\pi(2n_1+1)}{18}\right)$	$\cos\left(\frac{\pi(2n_2+1)}{18}\right)$

$(n_1 - 4)^2$	$(n_2 - 4)^2$
$\cos\left(\frac{2\pi(2n_1+1)}{18}\right)$	$\cos\left(\frac{2\pi(2n_2+1)}{18}\right)$

$(n_1 - 4)(n_2 - 4)$	$(n_1 - 4)^3$
$\cos\left(\frac{\pi(2n_1+1)}{18}\right)\cos\left(\frac{\pi(2n_2+1)}{18}\right)$	$\cos\left(\frac{3\pi(2n_1+1)}{18}\right)$

$(n_2 - 4)^3$	$(n_1 - 4)^2(n_2 - 4)$	$\ldots$
$\cos\left(\frac{3\pi(2n_2+1)}{18}\right)$	$\cos\left(\frac{2\pi(2n_1+1)}{18}\right)\cos\left(\frac{\pi(2n_2+1)}{18}\right)$	

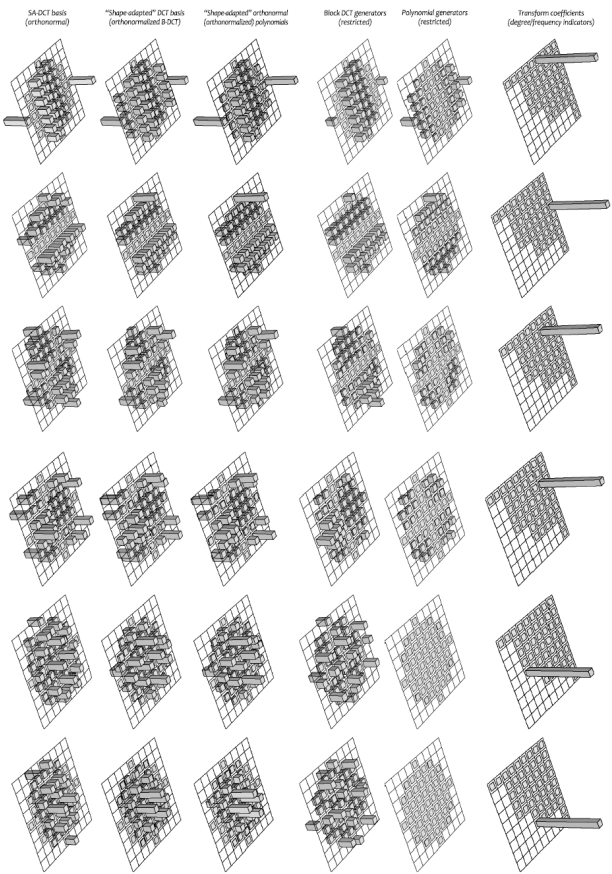
Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

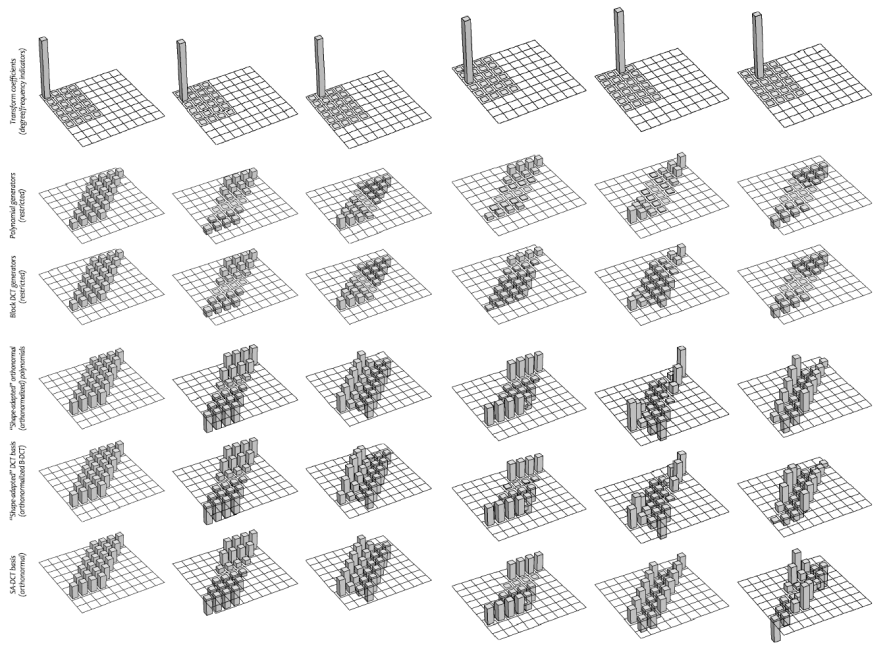
Transform coefficients
(degree/frequency indicators)
Polynomial generators
(rectDCnn)
Block DCT generators
(rectDCnn)
"Shape-adaptive" orthonormal
(orthonormalized) polynomials
"Shape-adaptive" DCT basis
(orthonormalized BADCT)
SA-DCT basis
(orthonormal)

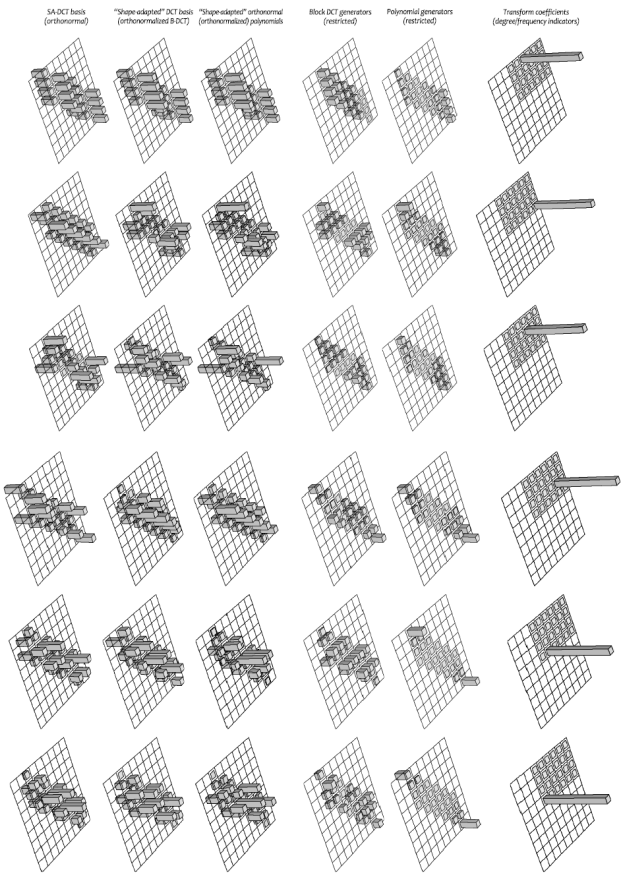
Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
SA-DCT basis
(orthonormal)

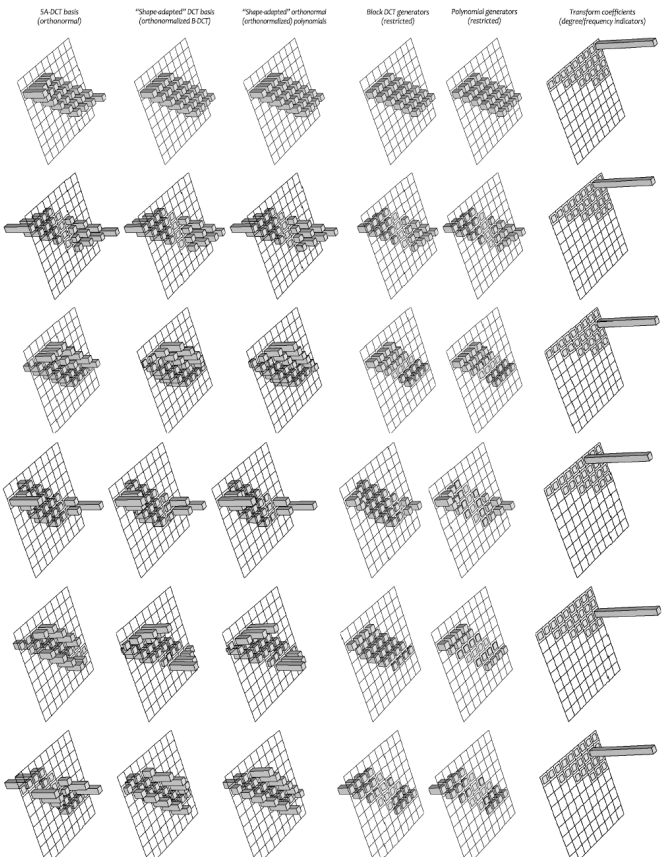
SA-DCT basis
(orthonormal)
"Shape-adapted" DCT basis
(orthonormalised B-DCT)
"Shape-adapted" orthonormal
(orthonormalised) polynomials
Block DCT generators
(restricted)
Polynomial generators
(restricted)
Transform coefficients
(degree/frequency indicators)

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adapted" orthonormal
(orthonormalised) polynomials
"Shape-adapted" DCT basis
(orthonormalised 8-DCT)
SA-DCT basis
(orthonormal)

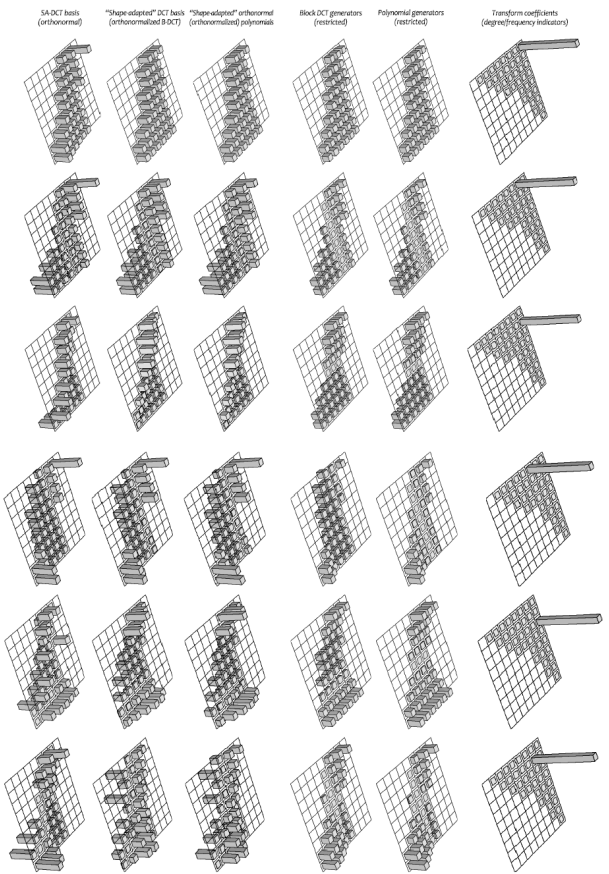

SA-DCT basis
(orthonormal)
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
Block-DCT generators
(restricted)
Polynomial generators
(restricted)
Transform coefficients
(degree/frequency indicators)

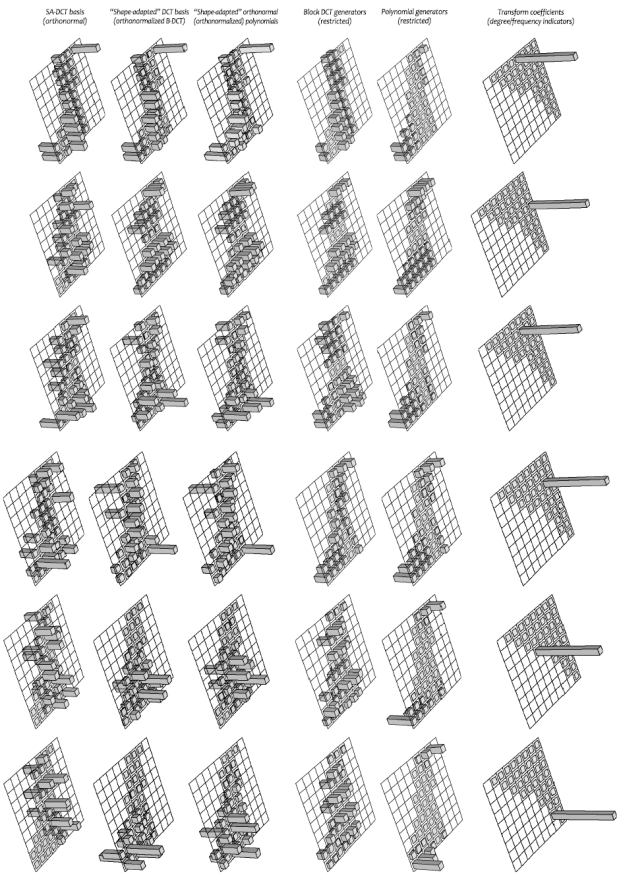

Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adaptive" orthonormal
(orthonormalized) polynomials
"Shape-adaptive" DCT basis
(orthonormalized) B-DCT
SA-DCT basis
(orthonormal)


SA-DCT basis (orthonormal)
"Shape-adapted" DCT basis (orthonormalized B-DCT)
"Shape-adapted" orthonormal (orthonormalized) polynomials
Block DCT generators (restricted)
Polynomial generators (restricted)
Transform coefficients (degree/frequency indicators)


SA-DCT basis
(orthonormal)
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
Block DCT generators
(restricted)
Polynomial generators
(restricted)
Transform coefficients
(degree/frequency indicators)

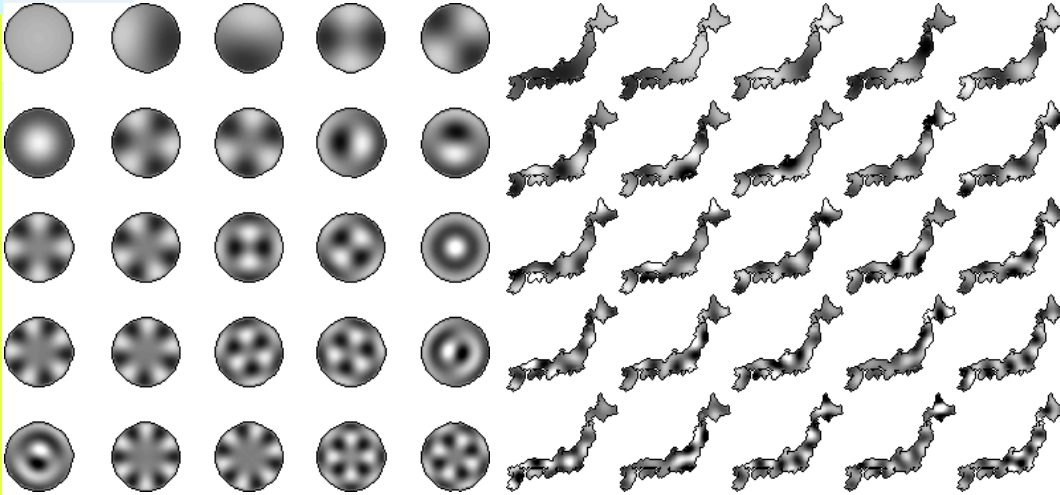
Transform coefficients
(degree/frequency indicators)
Polynomial generators
(restricted)
Block DCT generators
(restricted)
"Shape-adaptive" orthonormal
(orthonormalized) polynomials
"Shape-adaptive" DCT basis
(orthonormalized) B.DCT
SA-DCT basis
(orthonormal)


SA-DCT basis
(orthonormal)
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
Block DCT generators
(restricted)
Polynomial generators
(restricted)
Transform coefficients
(degree/frequency indicators)


SA-DCT basis
(orthonormal)
"Shape-adapted" DCT basis
(orthonormalized B-DCT)
"Shape-adapted" orthonormal
(orthonormalized) polynomials
Block DCT generators
(restricted)
Polynomial generators
(restricted)
Transform coefficients
(degree/frequency indicators)

Alternative approach: eigenfunctions of the Laplacian

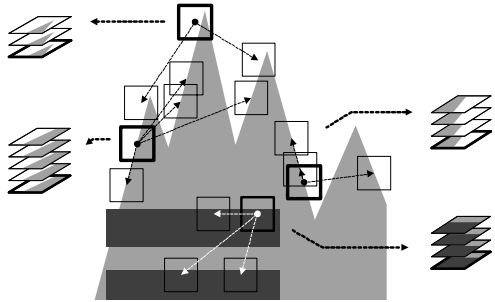
N. Saito, "Data analysis and representation on a general domain using eigenfunctions of Laplacian", submitted to *Applied and Computational Harmonic Analysis*, 27 August 2007 (available online at `http://math.ucdavis.edu/~saito/publications/lapeig.html`)



Non-local approach

Idea: *replace conventional local neighborhoods with data-driven non-local estimation domains, where the mutual similarity between different local regions determines the weights or the shape of the non-local domain.*

- Non-local means (Buades+Coll+Morel, 2005)
- Patch-based (Kervrann+Boulanger, 2006)
- Grouping and collaborative filtering (Dabov+Foi+Katkovnik+Egiazarian, 2006)



A simple example of grouping in an artificial image, where for each reference block (with thick borders), there exist perfectly matching ones, pointed to by corresponding arrows.

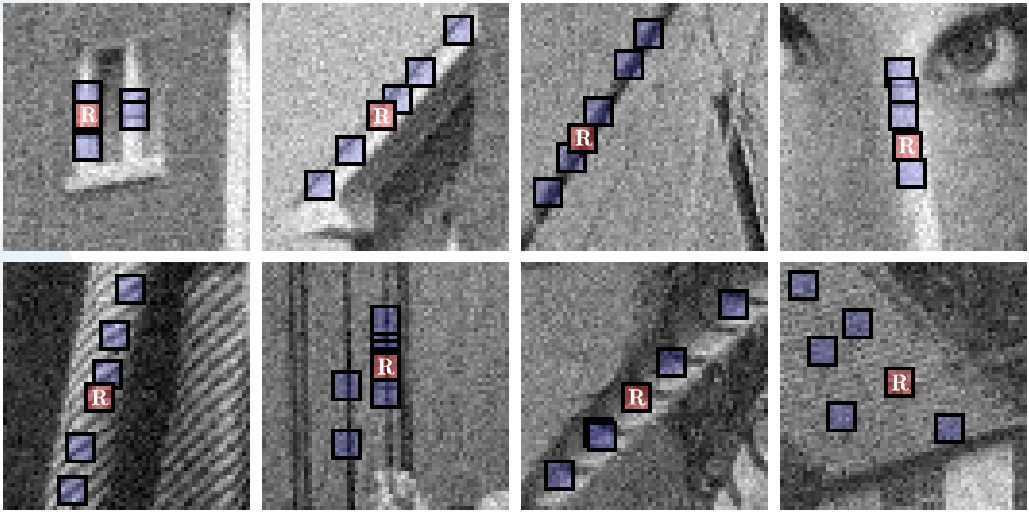
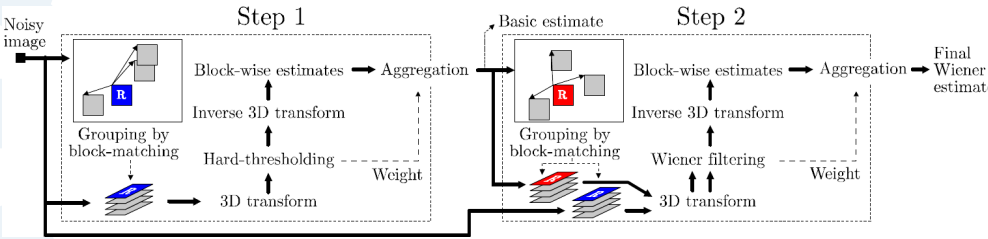


Illustration of grouping blocks from noisy natural images corrupted by white Gaussian noise with σ=15 and zero mean. Each fragment shows a reference block marked with "R" and a few of its matched ones.

Block-Matching 3D filtering (BM3D) algorithm



Flowchart of the BM3D algorithm

▷ Process overlapping blocks in a raster scan. For each such block, do the following:

1. (a) Use block-matching to find the locations of the blocks that are similar to the currently processed one. Form a 3D array (group) by stacking the blocks located at the obtained locations.
   (b) Apply a 3D transform on the formed group.
   (c) Attenuate the noise by shrinkage the 3D transform spectrum.
   (d) invert the 3D transform to produce filtered grouped blocks.

▷ Return the filtered blocks to their original locations in the image domain and compute the resultant filtered image by a weighted average of these filtered blocks (aggregation).

Block-Matching 3D filtering (BM3D) algorithm

- Collaborative filtering on the group is realized as shrinkage in 3D transform-domain.
- The algorithm structure is similar to Pointwise SA-DCT:
  - uses overcomplete local estimates aggregation;
  - two-stage scheme with hard-thresholding/empirical Wiener filtering.
- It is fast:
  - uses predictive-search for block-matching;
  - uses separable wavelets for the 3D transform (Haar/biorthogonal);
  - scalable (controlling the level of overcompleteness and predictive-search).
- Extensions to color image denoising, video denoising, as well as to joint sharpening and denoising have been reported.

To the best of our knowledge, it is currently the best performing denoising algorithm.

(Matlab software is publicly available)


(a) *Lena* (PSNR 32.08 dB)


(b) *Barbara* (PSNR 30.73 dB)


(e) *Cameraman* (PSNR 29.45 dB)


(c) *Man* (PSNR 29.62 dB)


(d) *Boats* (PSNR 29.91 dB)


(f) *Couple* (PSNR 29.72 dB)

BM3D denoising algorithm: fragments of noisy (σ=25) and denoised test images.

Publications, software, examples:

http://www.cs.tut.fi/~lasip (LPA-ICI)

http://www.cs.tut.fi/~foi/SA-DCT

http://www.cs.tut.fi/~foi/GCF-BM3D

References:

Bergmann, Ø., O. Christiansen, J. Lie, and A. Lundervold, "Shape Adaptive DCT for Denoising of Tensor Valued Images", submitted, (preprint available online at https://bora.uib.no/bitstream/1956/2159/1/Paper+E_Johan+Lie.pdf), Jan. 2007.

Buades, A., B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one", Multisc. Model. Simulat., vol. 4, no. 2, pp. 490-530, 2005.

Coifman, R.R., and D. Donoho, "Translation invariant de-noising", Technical Report 475, Dept. of Statistics, Stanford University, May 1995.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Inverse halftoning by pointwise shape-adaptive DCT regularized deconvolution", Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006, Florence, Sep. 2006.

Dabov, K., A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," European Signal Processing Conference (EUSIPCO-2007), September 2007.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," IEEE Int. Conf. Image Process., ICIP 2007, January 2007.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," IEEE Transactions on Image Processing, vol. 16, no. 8, August 2007.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Joint Image Sharpening and Denoising by 3D Transform-Domain Collaborative Filtering," Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2007, Moscow, Sep. 2007.

Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in Electronic Imaging'06, Proc. SPIE 6064, no. 6064A-30, San Jose, California USA, January 2006.

Fan, J., and I. Gijbels, Local polynomial modelling and its application, Chapman and Hall, London, 1996.

Farnebäck, G., "A Unified Framework for Bases, Frames, Subspace Bases, and Subspace Frames" Proc. of the 11th Scandinavian Conf. on Image Analysis, Kangerlussuaq, Greenland, 1999.

Foi, A., Anisotropic nonparametric image processing: theory, algorithms and applications, Ph.D. Thesis, Department of Mathematics, Politecnico di Milano, id. ERLTDD-D01290, April 2005.

Foi, A., S. Alenius, V. Katkovnik, and K. Egiazarian, "Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets", to appear in IEEE Sensors Journal, 2007 (in press).

Foi, A., S. Alenius, M. Trimeche, and V. Katkovnik, "Adaptive-size block transforms for Poissonian image deblurring", Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006, Florence, Sep. 2006.

Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", Proc. IEEE 2005 Int. Conf. Image Processing, ICIP 2005, September 2005.

Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Anisotropic local approximations for pointwise adaptive signal-dependent noise removal", Proc. XIII European Signal Proc. Conf., EUSIPCO 2005, September 2005.

Foi, A., R. Bilcu, V. Katkovnik, and K. Egiazarian, "Adaptive-Size Block Transforms for Signal-Dependent Noise Removal", Proc. 7th Nordic Signal Processing Symposium, NORSIG 2006, Reykjavik, Iceland, June 2006.

Foi, A., K. Dabov, V. Katkovnik, and K. Egiazarian, "Shape-adaptive DCT for denoising and image reconstruction", Proc. SPIE El. Imaging 2006, Image Process.: Algorithms and Systems V, 6064A-18, Jan. 2006.

Foi, A., and V. Katkovnik, "From local polynomial approximation to pointwise shape-adaptive transforms: an evolutionary nonparametric regression perspective", Proc. 2006 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006, Florence, Sep. 2006.

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

Foi, A., V. Katkovnik, K. Egiazarian, and J. Astola, "Inverse halftoning based on the anisotropic LPA-ICI deconvolution", Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Proc., SMMSP 2004, Vienna, pp. 49-56, September 2004.

Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT as an overcomplete denoising tool", Proc. 2005 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2005, pp. 164-170, Riga, June 2005.

Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images", IEEE Trans. Image Process., vol. 16, no. 5, pp. 1395-1411, May 2007.

Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT denoising with structure preservation in luminance-chrominance space", Proc. 2nd Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006, Scottsdale, AZ, Jan. 2006.

Foi, A., V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images", Proc. 14th Eur. Signal Process. Conf., EUSIPCO 2006, Florence, Sep. 2006.

Foi, A., V. Katkovnik, and K. Egiazarian, "Signal-dependent noise removal in Pointwise Shape-Adaptive DCT domain with locally adaptive variance", Proc. 15th Eur. Signal Process. Conf., EUSIPCO 2007, Poznan, Sep. 2007.

Gilge, M., T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform", Signal Process.: Image Comm., vol. 1, no. 2, pp. 153-180, Oct. 1989.

Gilge, M., "Region oriented transform coding (ROTC) of images", Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-90, Albuquerque, USA, vol. 4, pp. 2245-2248, Apr. 1990.

Goldenshluger, A., "On pointwise adaptive nonparametric deconvolution", Bernoulli, vol. 5, pp. 907-925, 1999.

Goldenshluger, A., and A. Nemirovski, "On spatial adaptive estimation of nonparametric regression", Math. Meth. Statistics, vol. 6, pp. 135-170, 1997.

Katkovnik, V., "A new method for varying adaptive bandwidth selection", IEEE Trans. Signal Process., vol. 47, no. 9, pp. 2567-2571, 1999.

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

Katkovnik, V., A. Foi, K. Egiazarian, and J. Astola, "Anisotropic local likelihood approximations", Proc. of Electronic Imaging 2005, 5672-19, January 2005.

Katkovnik, V., K. Egiazarian, and J. Astola, Adaptive varying scale methods in image processing. Tampere International Center for Signal Processing, TICSP Series, no. 19, Tampere, TTY, Monistamo, 2003.

Katkovnik, V., K. Egiazarian, and J. Astola, Local Approximation Techniques in Signal and Image Processing, SPIE Press, Monograph Vol. PM157, September 2006.

Katkovnik, V., K. Egiazarian, and J. Astola, "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule", J. Math. Imaging and Vision, vol. 16, no. 3, pp. 223-235, 2002.
Katkovnik, V., and I. Shmulevich, "Kernel density estimation with adaptive varying window size", Pattern Recognition Letters, no. 23, pp. 1641-1648, 2002.
Katkovnik, V., K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring", IEEE Trans. on Image Process., vol. 14, no. 10, pp. 1469-1478, October 2005.
Kauff, P., and K. Schuur, "An extension of shape-adaptive DCT (SA-DCT) towards DC separation and ΔDC correction", Proc. of 1997 Picture Coding Symposium, pp. 647-652, September 1997.
Kauff, P., and K. Schuur, "Shape-adaptive DCT with block-based DC separation and ΔDC correction", IEEE Trans. Circuits Syst. Video Technol., vol. 8, no. 3, pp. 237-242, 1998.
Kaup, A., and S. Panis, "On the performance of the shape adaptive DCT in object-based coding of motion compensated difference images", Proc. of 1997 Picture Coding Symposium, pp. 653-657, 1997.
Kervrann, C., and J. Boulanger, "Optimal spatial adaptation for patch based image denoising", IEEE Trans. Image Process., vol. 15, no. 10, pp. 2866-2878, October 2006.
Koenen, R., "Overview of the MPEG-4 Standard", ISO/IEC JTC1/SC29/WG11 Doc. N3536, July 2000.
Lie, J., Mathematical Imaging with Applications to MRI and Diffusion Tensor MRI, Ph.D. Thesis, Department of Mathematics, The University of Bergen, (available online at https://bora.uib.no/bitstream/1956/2159/6/Main+Thesis_Johan+Lie.pdf), March 2007.
MPEG-4: Information technology - Coding of audio-visual objects - Part 2: Visual, ISO/IEC 14496-2:2001, Dec. 2001.
Ostermann, J., E.S. Jang, J. Shin, and T. Chen, "Coding of arbitrarily shaped video objects in MPEG-4", Proc. Int. Conf. Image Process., ICIP 1997, pp. 496-499, 1997.
Philips, W., "A Fast Algorithm for Orthogonalizing Polynomials on an Arbitrarily Shaped Region (Revised Version)", Multidimensional Systems and Signal Processing, vol. 8, no. 4, pp. 409-421, October 1997.
Philips, W., "Orthogonal base functions on a discrete two-dimensional region", ELIS Technical Report DG 91-20, Dept. of Electronics and Inform. Syst., Universiteit Gent, Belgium, November 1993.
Rao, K.R., and P. Yip, Discrete Cosine Transform, Algorithms, Advantages and Applications, Academic Press, 1990.
Sikora, T., "Low complexity shape-adaptive DCT for coding of arbitrarily shaped image segments", Signal Processing: Image Communication, vol. 7, pp. 381-395, 1995.
Sikora, T., S. Bauer, and B. Makai, "Efficiency of shape-adaptive 2-D transforms for coding of arbitrarily shaped image segments," IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 3, pp. 254-258, June 1995.
Sikora, T., and B. Makai, "Shape-adaptive DCT for generic coding of video", IEEE Trans. Circuits Syst. Video Technol., vol. 5, no. 1, pp. 59-62, 1995.
Sikora, T., and B. Makai, "Shape-adaptive DCT for generic coding of video", IEEE Trans. on Circuits and Systems for Video Technology, vol. 5, no. 1, pp. 59-62, 1995.
Simoncelli, E.P., and H. Farid, "Steerable wedge filters for local orientation analysis", IEEE Trans. on Image Proc., vol. 5, no. 9, pp. 1377-1382, 1996.
Van der Weken, D., E. Kerre, E. Vansteenkiste, and W. Philips, "Evaluation of fuzzy image quality measures using a multidimensional scaling framework", Proc. 2nd Int. Workshop Video Process. Quality Metrics Consum. Electron., VPQM2006, Scottsdale, AZ, Jan. 2006.
Vansteenkiste, E., D. Van der Weken, W. Philips, and E. Kerre, "Perceived image Quality Measurement of state-of-the-art Noise Reduction Schemes", Lecture Notes in Computer Science 4179 - ACIVS 2006, pp. 114-124, Springer, Sep. 2006.
Yaroslavsky, L., K. Egiazarian, and J. Astola, Transform Domain Image Restoration Methods, Review, Comparison and Interpretation, TICSP Series no. 9, Tampere, 2000.