Adaptive Wavelet Methods for the Efficient Approximation of Images

Gerlind Plonka

Institute for Numerical and Applied Mathematics
University of Göttingen

in collaboration with Dennis Heinen, Armin Iske
Daniela Roşca, Stefanie Tenorth

Dolomites Research Week on Approximation

September, 2014

Adaptive Wavelet Methods for Image Approximation

Outline

- Introduction: Adaptive wavelet transforms
  - Generalized lifting schemes
  - Geometric approaches with adaptivity costs
- Description of the EPWT algorithm
- Examples and experiments
- A hybrid method using the EPWT
- Numerical experiments
- Denoising of scattered data
- References

Introduction

Idea Design adaptive approximation schemes respecting the local geometric regularity of two-dimensional functions

Basic adaptive wavelet approaches

a) Apply a generalized lifting scheme to the data using (nonlinear) data-dependent prediction and update operators

b) Adaptive approximation schemes using geometric image information, usually with extra adaptivity costs

Basic adaptive wavelet approaches

a) Apply a generalized lifting scheme to the data using (nonlinear) data-dependent prediction and update operators

Literature (incomplete)

- **discrete MRA and generalized wavelets** (Harten '93)
- **second generation wavelets** (Sweldens '97)
- **edge adapted multiscale transform** (Cohen & Matei '01)
- **Nonlinear wavelet transforms** (Claypoole et al. '03)
- **adaptive lifting schemes** (Heijmans et al. '06)
- **adaptive directional lifting based wavelet transf.** (Ding et al. '06)
- **edge-adapted nonlinear MRA (ENO-EA)** (Arandiga et al. '08)
- **meshless multiscale decompositions** (Baraniuk et al. '08)
- **nonlinear locally adaptive filter banks** (Plonka & Tenorth '09)

How does it work?

The general lifting scheme consists of three steps.

1. **Split** Split the given data $\mathbf{a} = (a(i,j))_{i,j=0}^{N-1}$ into two sets $\mathbf{a}^e$ and $\mathbf{a}^o$

2. **Predict** Find a good approximation $\tilde{\mathbf{a}}^o$ of $\mathbf{a}^o$ of the form

$$\tilde{\mathbf{a}}^o = P_1\mathbf{a}^o + P_2\mathbf{a}^e$$

Put

$$\mathbf{d}^o := \tilde{\mathbf{a}}^o - \mathbf{a}^o.$$

Assume that $(\mathbf{a}^e, \mathbf{a}^o) \mapsto (\mathbf{a}^e, \mathbf{d}^o)$ is invertible, i.e., $I - P_1$ is invertible.

3. **Update** Find a "smoothed" approximation of $\mathbf{a}^e$ (a low-pass filtered subsampled version of $\mathbf{a}$)

$$\tilde{\mathbf{a}}^e := U_1(\mathbf{d}^o) + U_2(\mathbf{a}^e)$$

Assume that $(\mathbf{a}^e, \mathbf{d}^o) \mapsto (\tilde{\mathbf{a}}^e, \mathbf{d}^o)$ is invertible, i.e., that U_2 is invertible.

How to choose the prediction and update operators?

Prediction operator local approximation of $\mathbf{a}^o$ by an adaptively weighted average of "neighboring" data

Example 1.

- Fix a stencil at a neighborhood of $a^o(i,j)$ (adaptively)
- Compute a polynomial p by interpolating/approximating the data on the stencil
- Choose $p(i,j)$ to approximate $a^o(i,j)$.

Example 2. Use nonlinear diffusion filters to determine the prediction operator

Update operator usually linear, non-adaptive

Basic adaptive wavelet approaches

b) Adaptive wavelet approximation schemes using geometric image information, usually with extra adaptivity costs

Literature (incomplete)

- **wedgelets** (Donoho '99)
- **bandelets** (Le Pennec & Mallat '05)
- **geometric wavelets** (Dekel & Leviatan '05)
- **geometrical grouplets** (Mallat '09)
- **EPWT** (Plonka et al. 09)
- **tetrolets** (Krommweh '10)
- **generalized tree-based wavelet transform** (Ram, Elad et al. '11)

Basic adaptive wavelet approaches

wedgelets (Donoho '99)

approximation of images using an adaptively chosen domain decomposition

bandelets (Le Pennec & Mallat '05)

wavelet filter bank followed by adaptive geometric orthogonal filters

geometric wavelets (Dekel & Leviatan '05)

binary space partition and polynomial approximations in subdomains

geometrical grouplets (Mallat '09)

association fields that group points, generalized Haar wavelets

EPWT (Plonka et al. 09)

tetrolets (Krommweh '10)

generalized Haar wavelets on adaptively chosen tetrolet partitions

Comparison of basic adaptive wavelet approaches

a) Generalized lifting scheme with nonlinear prediction

Advantages invertible transform, no side information necessary
usually a justifiable computational effort

Drawbacks bad stability of the reconstruction scheme
only slightly better approximation results compared with linear (nonadaptive) transforms

b) Adaptive wavelet approximation using geometric image information

Advantages very good approximation results

Drawbacks adaptivity costs for encoding
usually high computational effort

Description of the EPWT

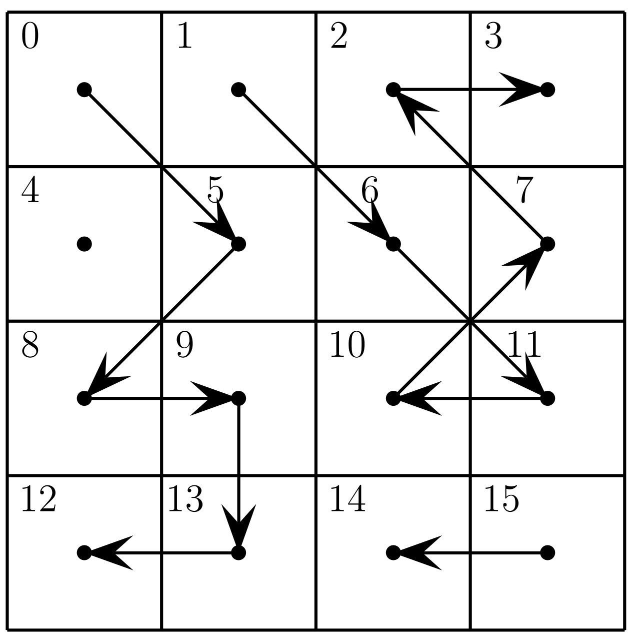
Problem Given a matrix of data points (image values), how to compress the data by a wavelet transform thereby exploiting the local correlations efficiently?

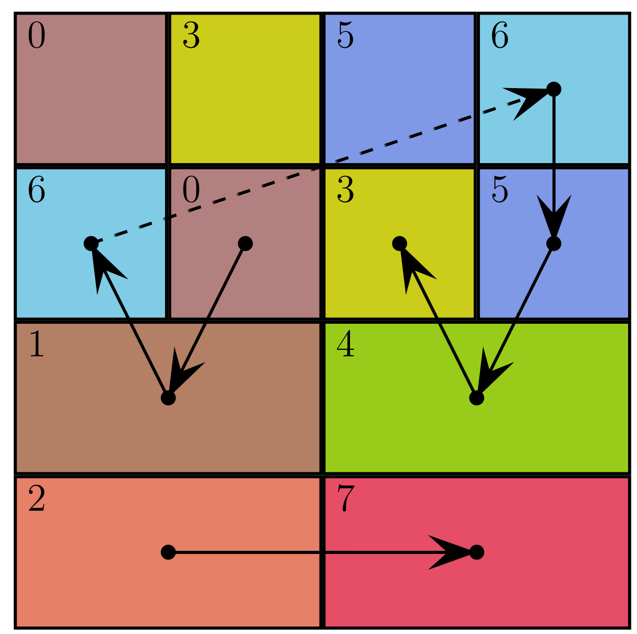
Idea

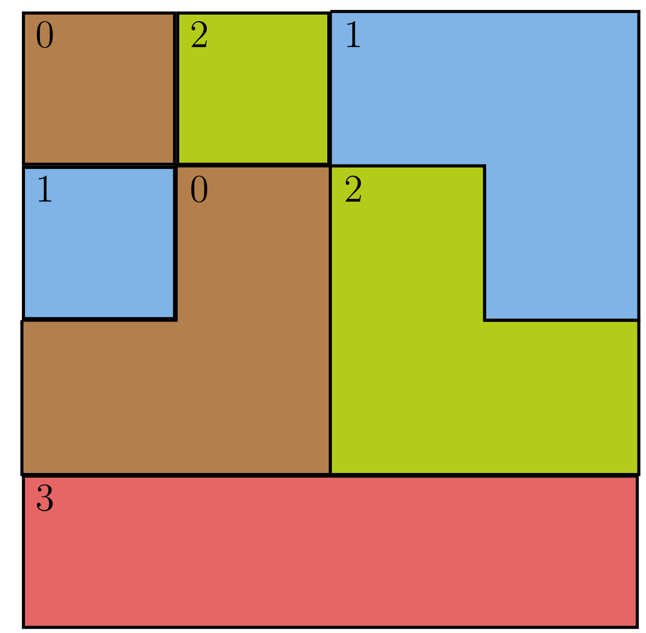
1. Find a (one-dimensional) path through all data points such that there is a strong correlation between neighboring data points.
2. Apply a one-dimensional wavelet transform along the path.
3. Apply the idea repeatedly to the low-pass filtered array of data.

Toy Example

$$\mathbf{f} = \begin{bmatrix} 115 & 108 & 109 & 112 \\ 106 & 116 & 107 & 109 \\ 112 & 110 & 108 & 108 \\ 108 & 109 & 103 & 106 \end{bmatrix} \qquad \text{array of data.}$$







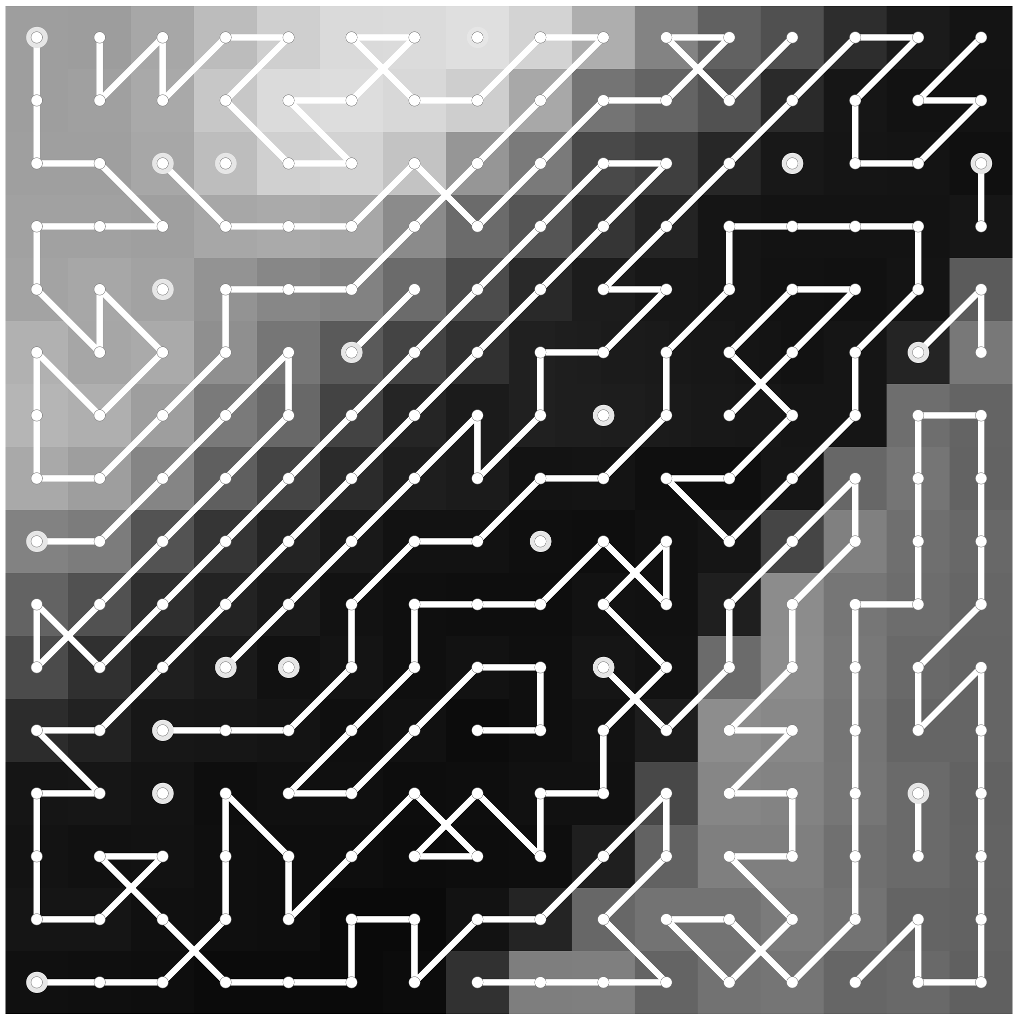
$$\mathbf{p}^4 = ((0,5,8,9,13,12),(1,6,11,10,7,2,3),(4),(15,14)),$$

$$\mathbf{f}^3 = (115.5, 111, 108.5, 107.5, 108, 109, 109, 104.5),$$

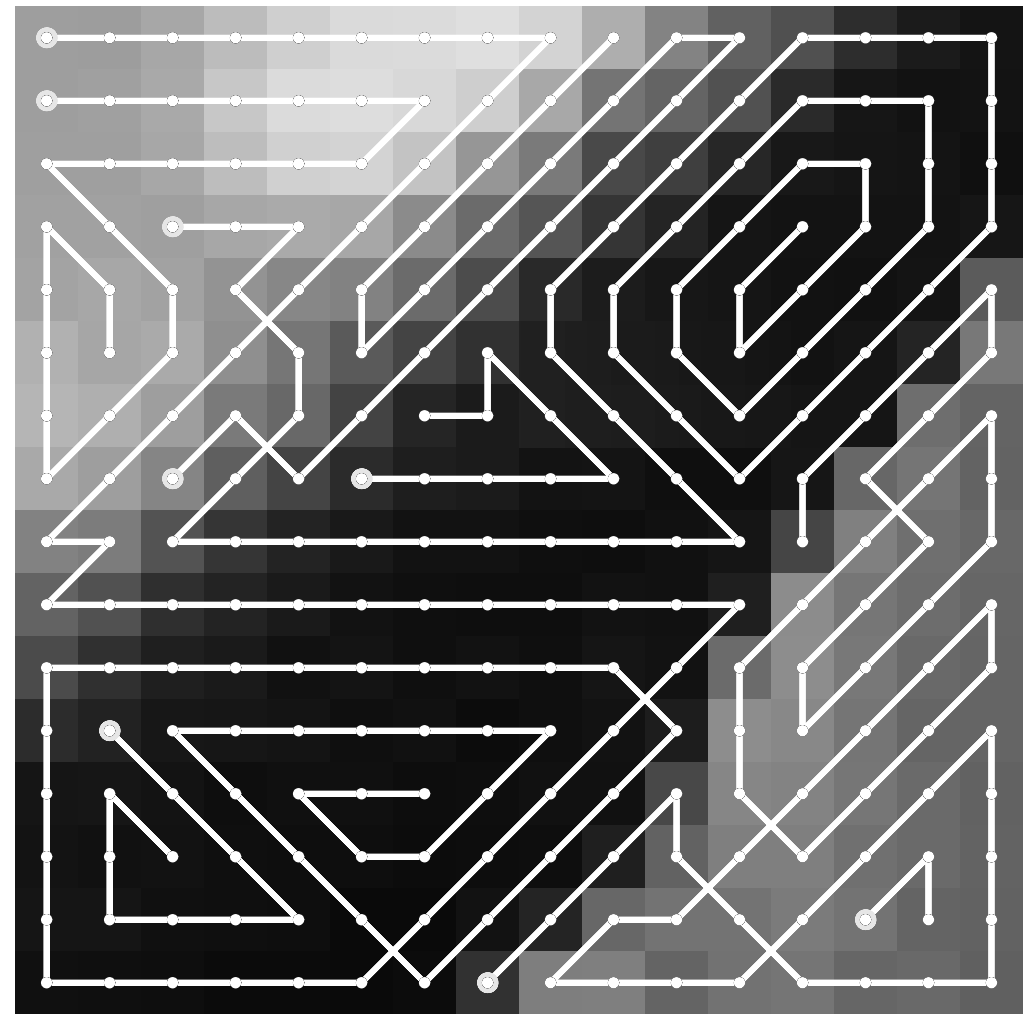
$$\mathbf{p}^3 = ((0,1,6,5,4,3),(2,7)), \qquad \mathbf{p}^2 = (0,1,2,3).$$

The relaxed EPWT

Idea: Change the direction of the path only if the difference of data values is greater than a predetermined value θ.



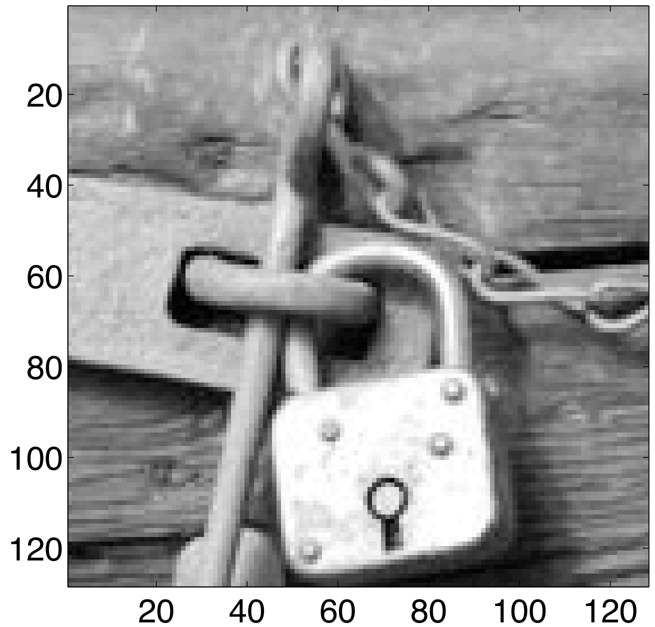
rigorous EPWT ($\theta = 0$)
Entropy 2.08 bit per pixel



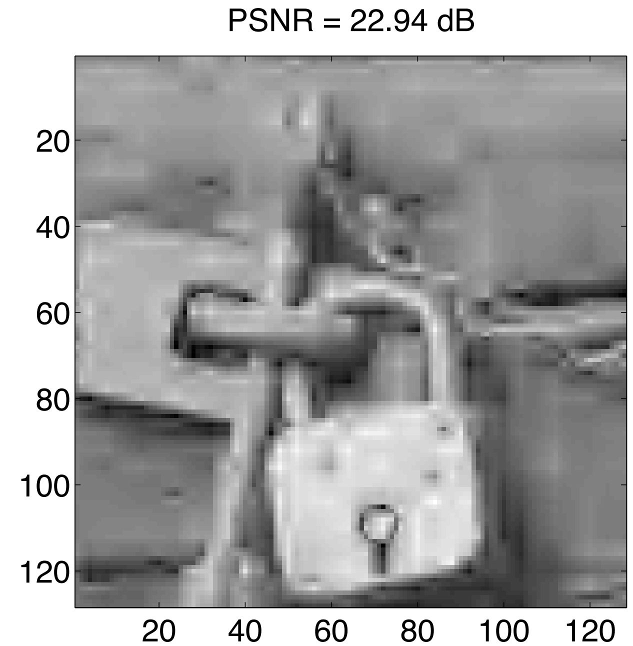
relaxed EPWT ($\theta = 0.14$)
Entropy 0.39 bit per pixel

Test: door lock image (128×128)

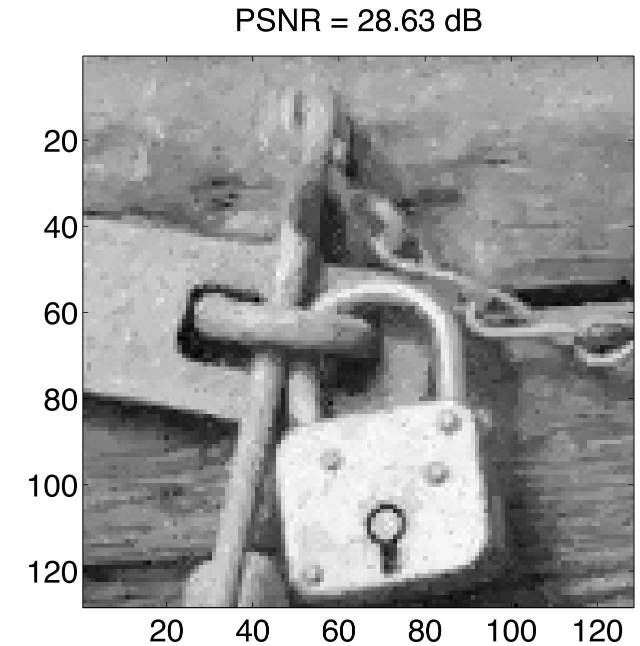
WT	θ_1	levels	nonzero coeff	PSNR	entropy of $\tilde{\mathbf{p}}^{14}$
tensor prod. Haar	-	7	512	22.16	-
tensor prod Daub.	-	6	512	22.94	-
tensor prod 7-9	-	4	512	22.49	-
EPWT Haar	0.00	14	512	28.04	2.22
EPWT Haar	0.05	14	512	28.37	1.11
EPWT Haar	0.10	14	512	27.74	0.55
EPWT Daub.	0.00	12	512	28.63	2.22
EPWT Daub.	0.05	12	512	29.23	1.11
EPWT Daub.	0.10	12	512	28.67	0.55
EPWT Daub.	0.15	12	512	27.65	0.32
EPWT 7-9	0.00	10	512	28.35	2.22
EPWT 7-9	0.05	10	512	28.99	1.11
EPWT 7-9	0.10	10	512	28.38	0.55



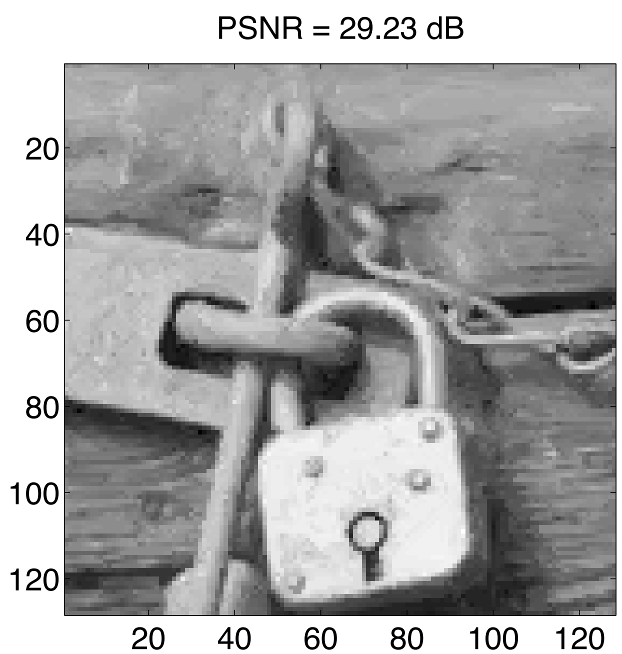
original image



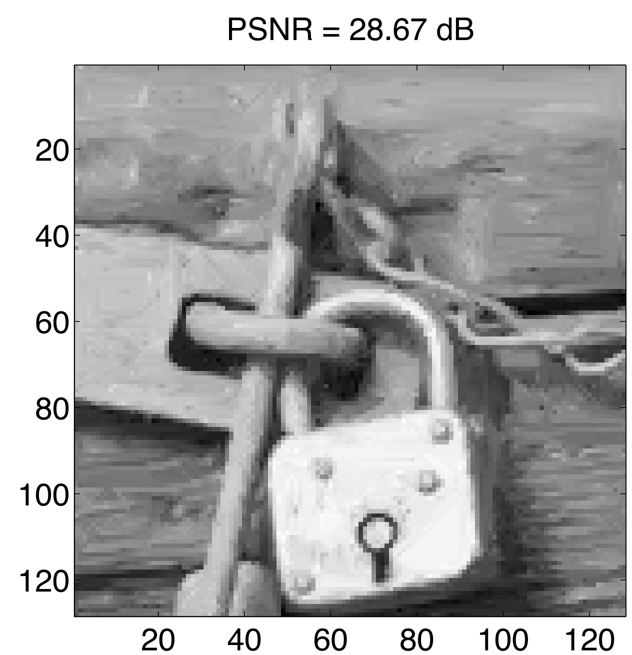
D4, 512 coeff.
PSNR= 22.94



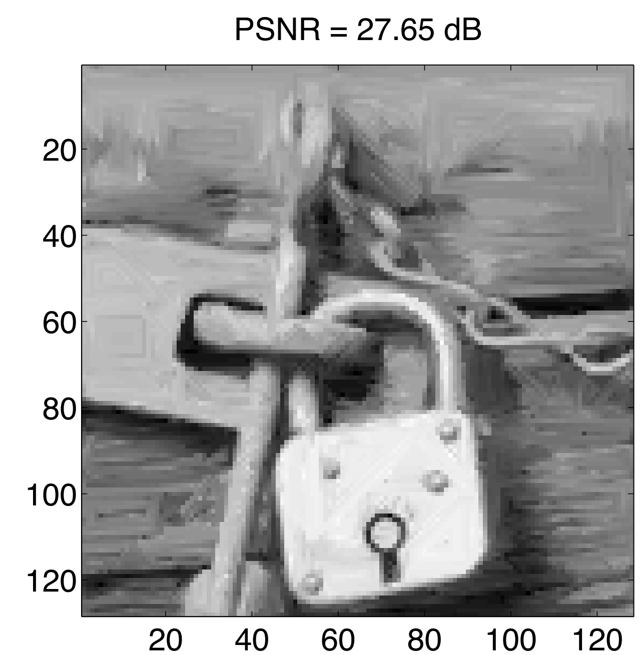
EPWT $\theta_1 = 0$
PSNR=28.63



EPWT, $\theta_1 = 0.05$
PSNR=29.23



EPWT, $\theta_1 = 0.1$
PSNR=28.67



EPWT, $\theta_1 = 0.15$
PSNR = 27.65

Results for N-term approximation

Theorem 1 (Plonka, Tenorth, Iske (2011))

The EPWT (with the Haar wavelet transform) leads for suitable path vectors to an N-term approximation of the form

$$\|f - f_N\|_2^2 \le C\, N^{-\alpha}$$

for piecewise Hölder continuous functions of order α (with $0 < \alpha \le 1$) possessing discontinuities along curves of finite length.

Theorem 2 (Plonka, Iske, Tenorth (2013))

The application of the EPWT leads for suitably chosen path vectors to an N-term approximation of the form

$$\|f - f_N\|_2^2 \le C\, N^{-\alpha}$$

for piecewise Hölder smooth functions of order $\alpha > 0$ possessing discontinuities along curves of finite length.

The hybrid method using the EPWT

Idea

1. Apply an image separation into a smooth image part and a remainder part containing edges and texture

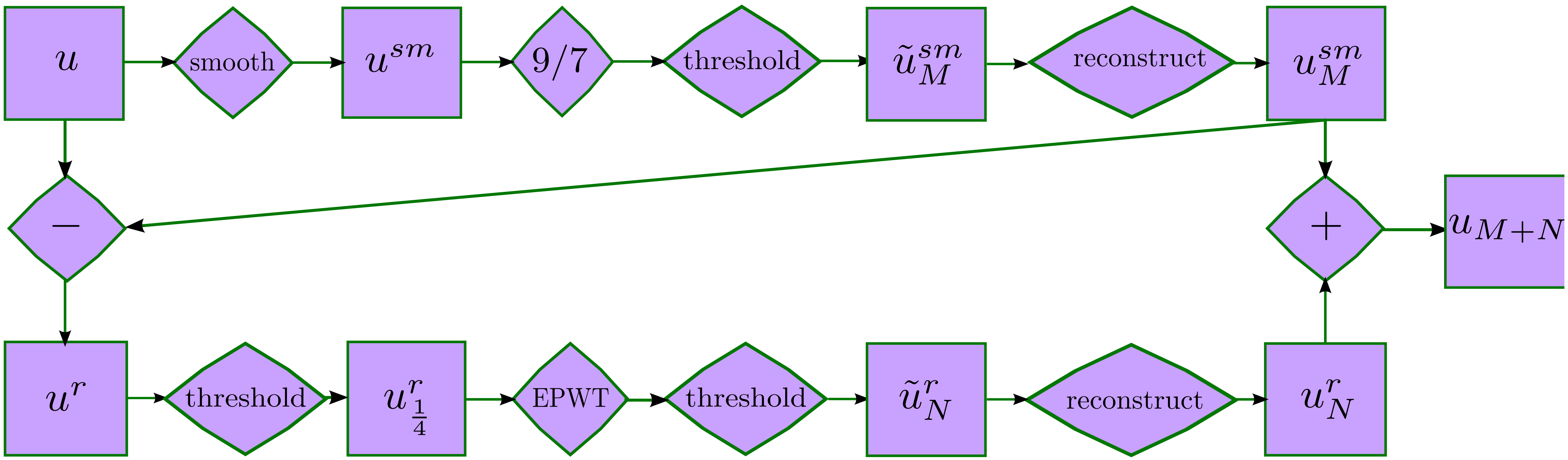
$$u = u^{sm} + u^r$$

   using e.g. a suitable smoothing filter.

2. Apply a tensor product wavelet transform to the smooth image part u^{sm} to get an N-term approximation u_N^{sm}.
3. Apply the EPWT to the (shrinked) remainder u^r to get am M-term approximation u_M^r.
4. Add u_N^{sm} and u_M^r to find a good approximation of u.

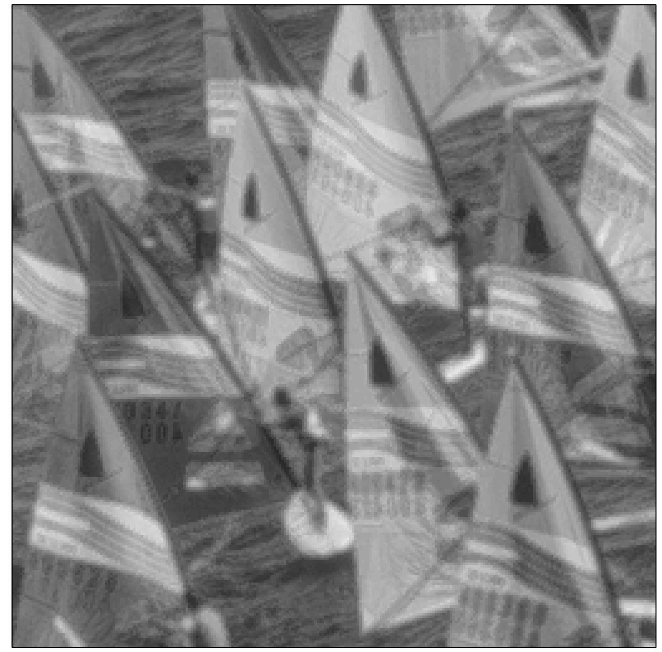
A sketch of the hybrid method

We use the tensor-product wavelet transform for the smoothed image and the EPWT for the (shrunken) difference image.



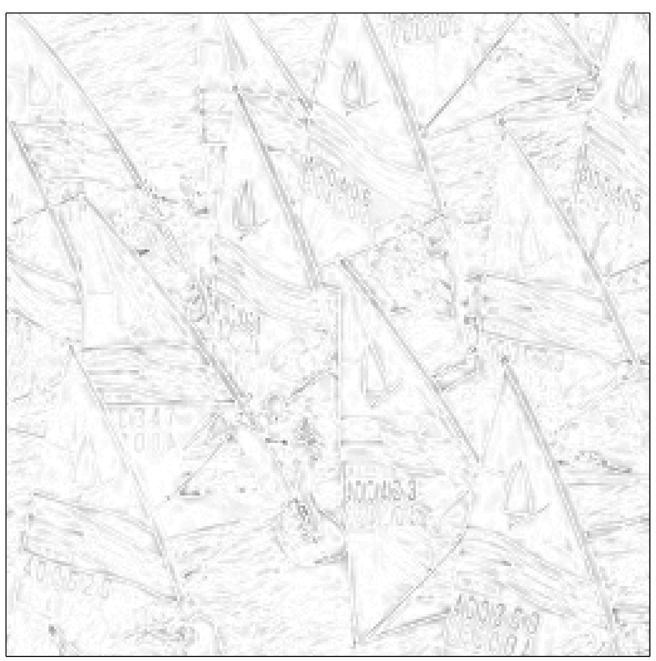
Example

Original image

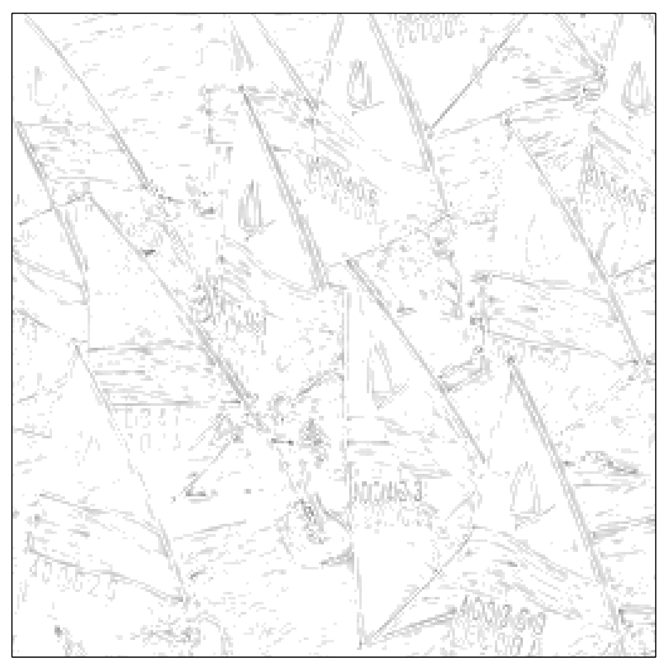


smoothed image u^{sm}

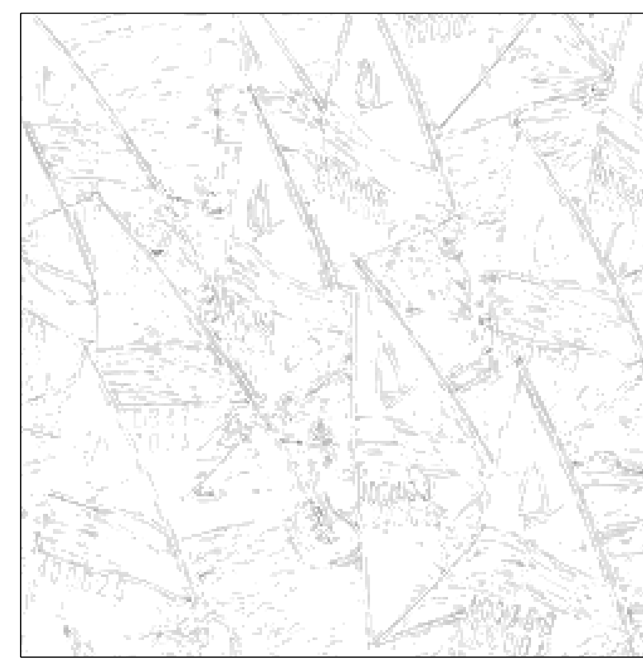
wavelet approximation u^{sm}_{1200}



difference image u^r



shrunken difference $u^r_{1/4}$



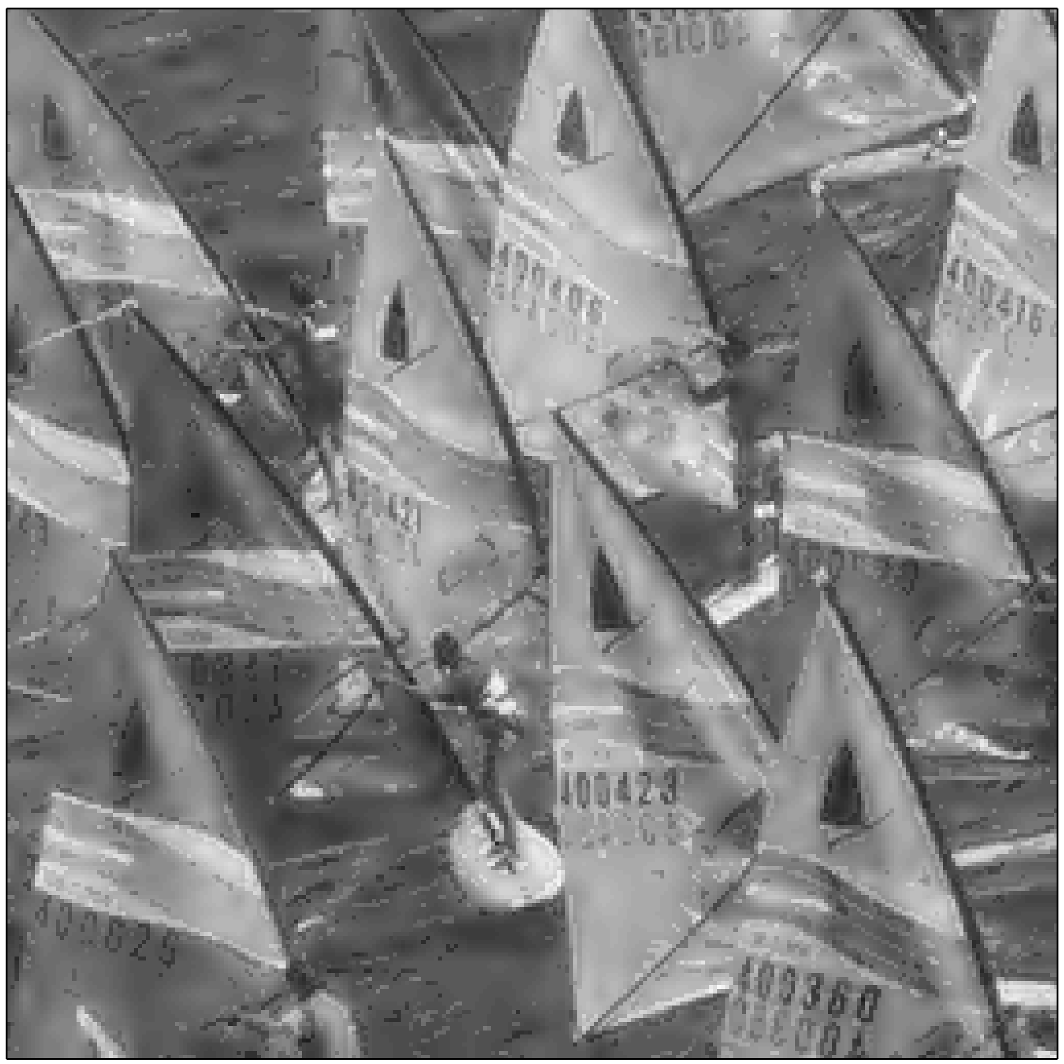
EPWT approximation u^r_{800}

Example continued

N-term approximation with $N = 2000$.

(a) $u_{1200+800}$ using the new hybrid method

(b) u_{2000} using the 9/7 wavelet transform with 2000 non-zero elements



(a)

(b)

Numerical results for the hybrid method

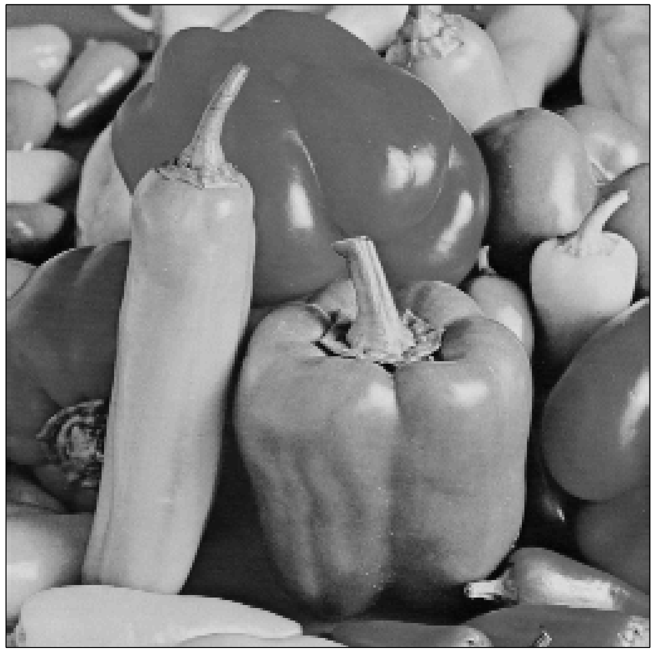
image	nzc	9/7 PSNR	Hybrid PSNR	entropy
barbara	500	23.33	27.28	1.0070
cameraman	500	22.54	27.49	0.9893
clock	500	24.61	30.87	0.8742
goldhill	500	24.18	28.19	0.8408
lena	500	23.21	27.91	0.9022
pepper	500	23.41	28.03	0.8795
sails	500	21.32	25.42	0.9190

Hybrid: Search for suitable path vectors in each level

Original image

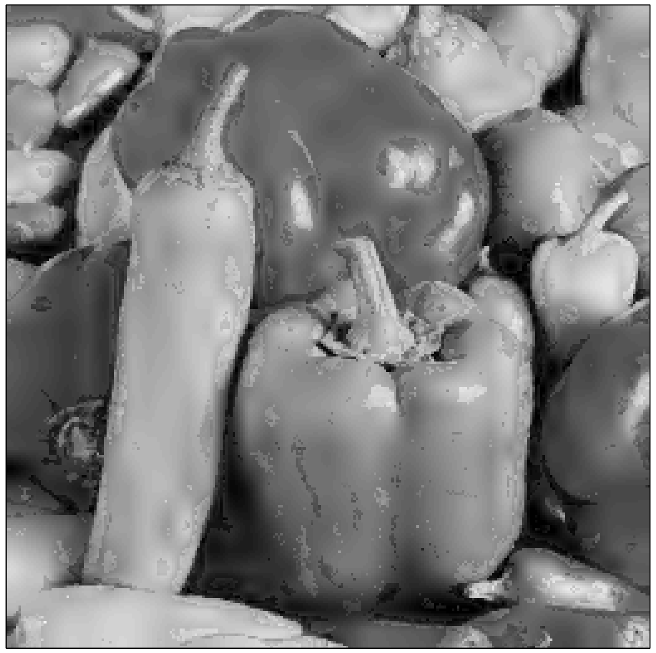
7/9, 500 coeff.
PSNR= 23.21

Hybrid, 500 coeff.
PSNR=27.91



Original image

7/9, 500 coeff.
PSNR=23.41



Hybrid, 500 coeff.
PSNR = 28.03

Denoising of scattered data using the EPWT approach

Given

a set of d-dimensional points $\Gamma = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^d$

noisy function values $\tilde{f}(x_j) = f(x_j) + z_j, \;\; j = 1, \ldots, N$

where

$f : \mathbb{R}^d \to \mathbb{R}$ piecewise smooth

z_j independent and $\mathcal{N}(0, \sigma_j^2)$ distributed (Gaussian noise)

Wanted denoised function values $f(x_j)$

Classical wavelet shrinkage

wavelet decomposition

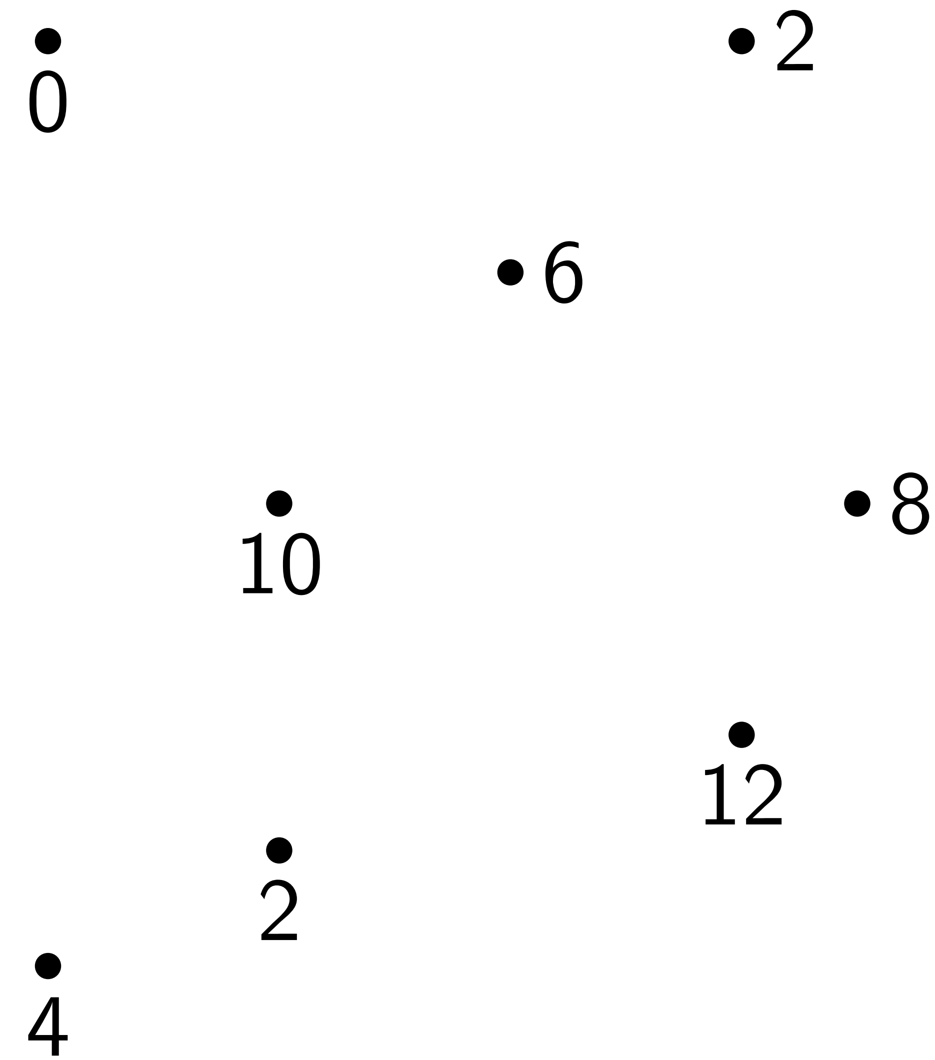
shrinkage: set small high-pass coefficients to zero

wavelet reconstruction

Analogon of cycle shift:
average [shift $\to$ wavelet shrinkage $\to$ un-shift]

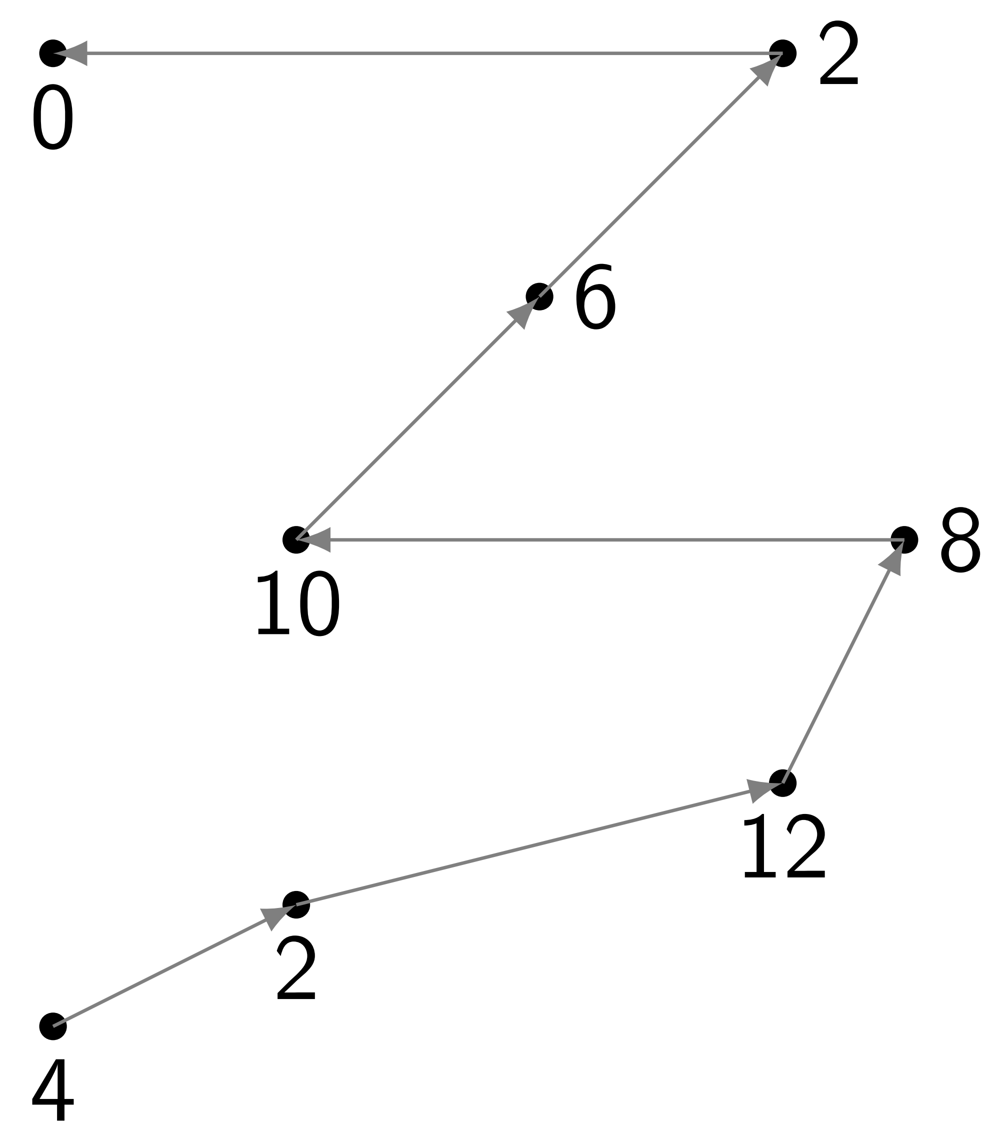
Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points



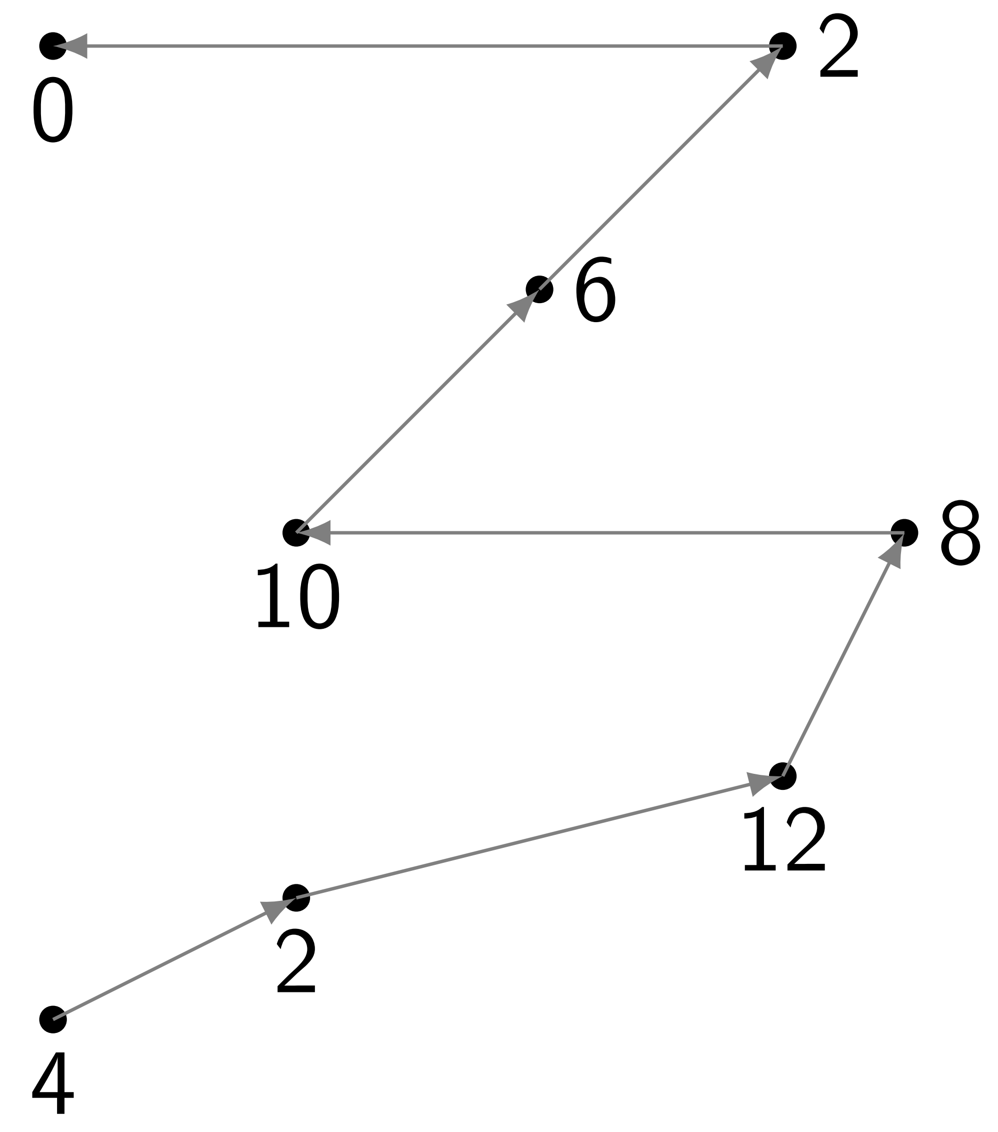
Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points



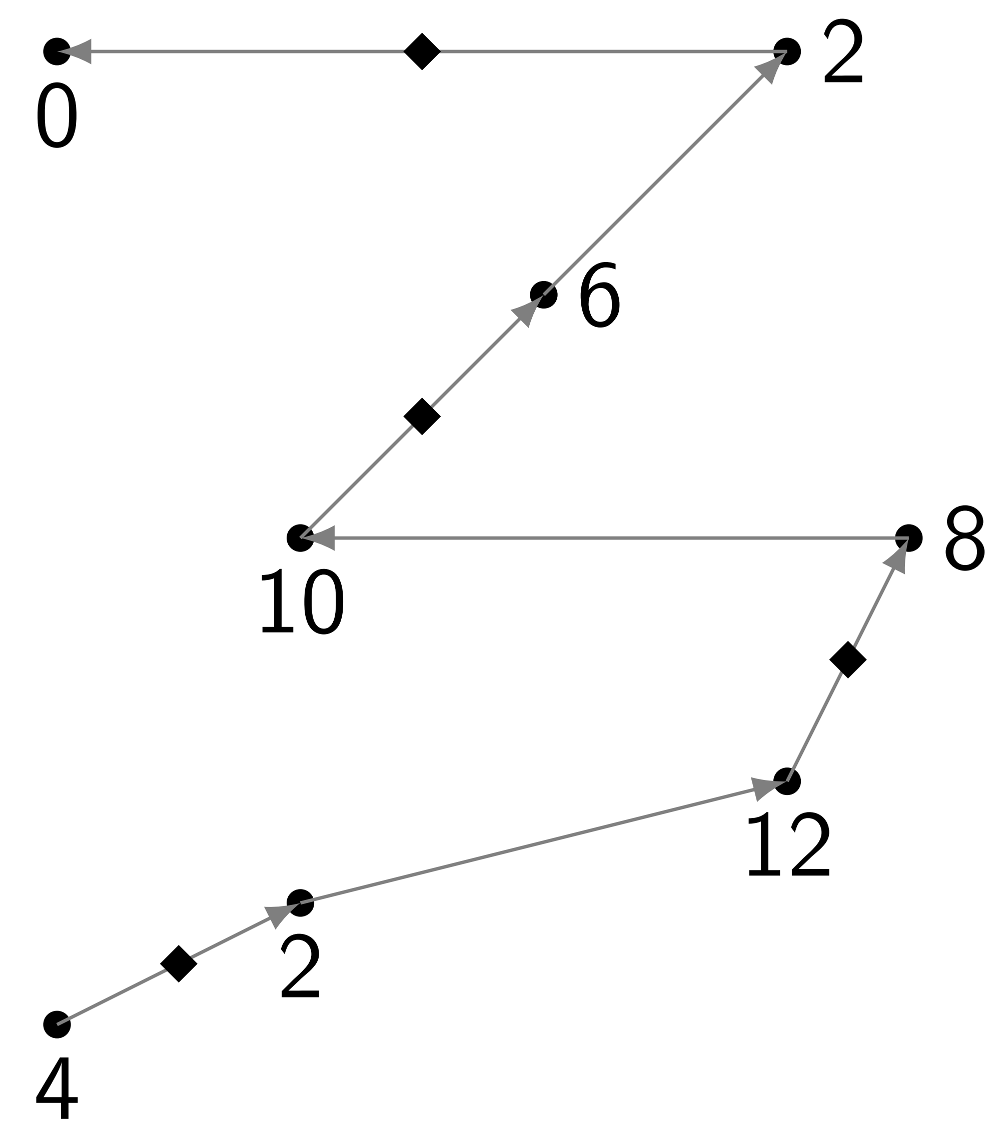
Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points
- apply 1D wavelet transform along the path
  low pass coefficients $(3, 10, 8, 1)$
  high pass coefficients $(1, 2, 2, 1)$



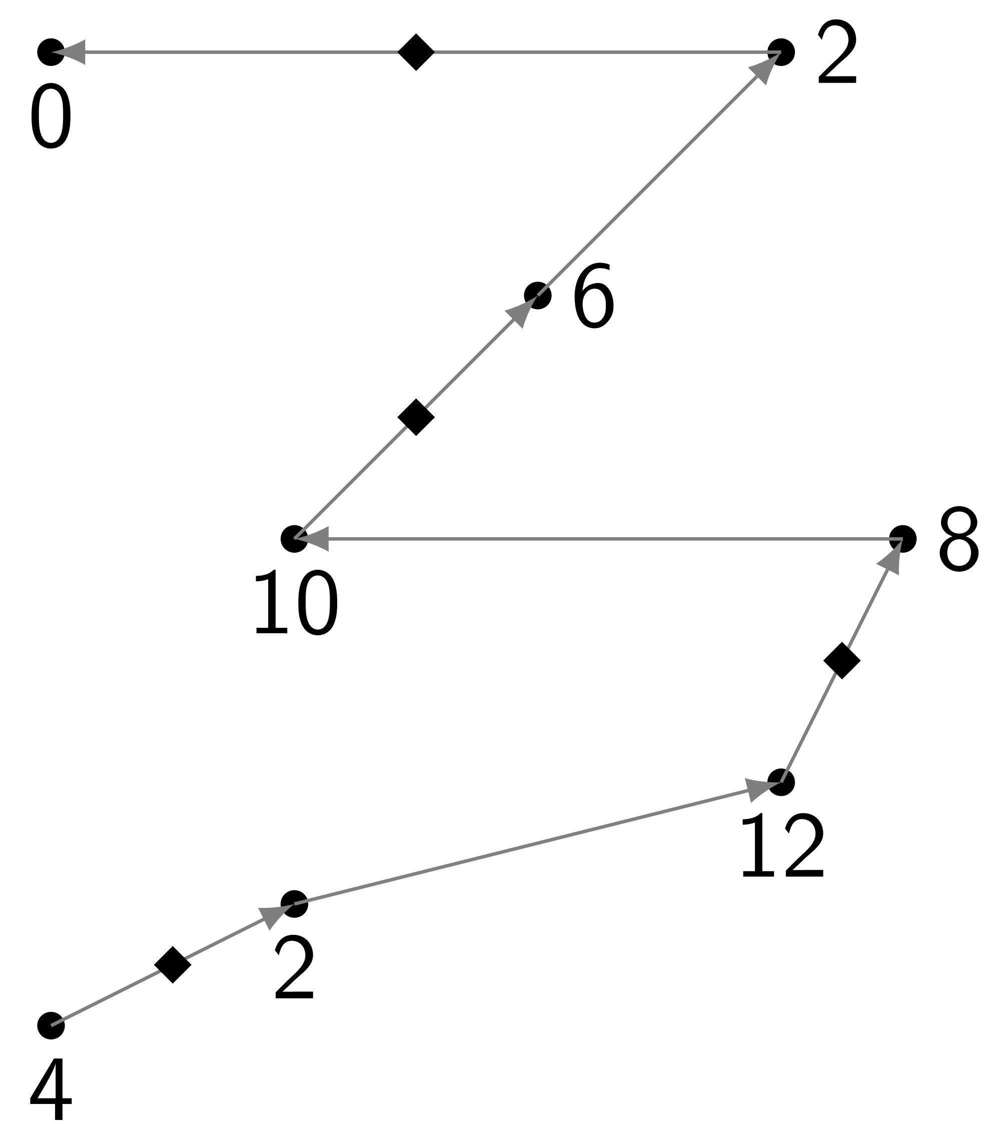
Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points
- apply 1D wavelet transform along the path low pass coefficients $(3, 10, 8, 1)$ high pass coefficients $(1, 2, 2, 1)$
- update point set
- apply shrinkage to wavelet coefficients



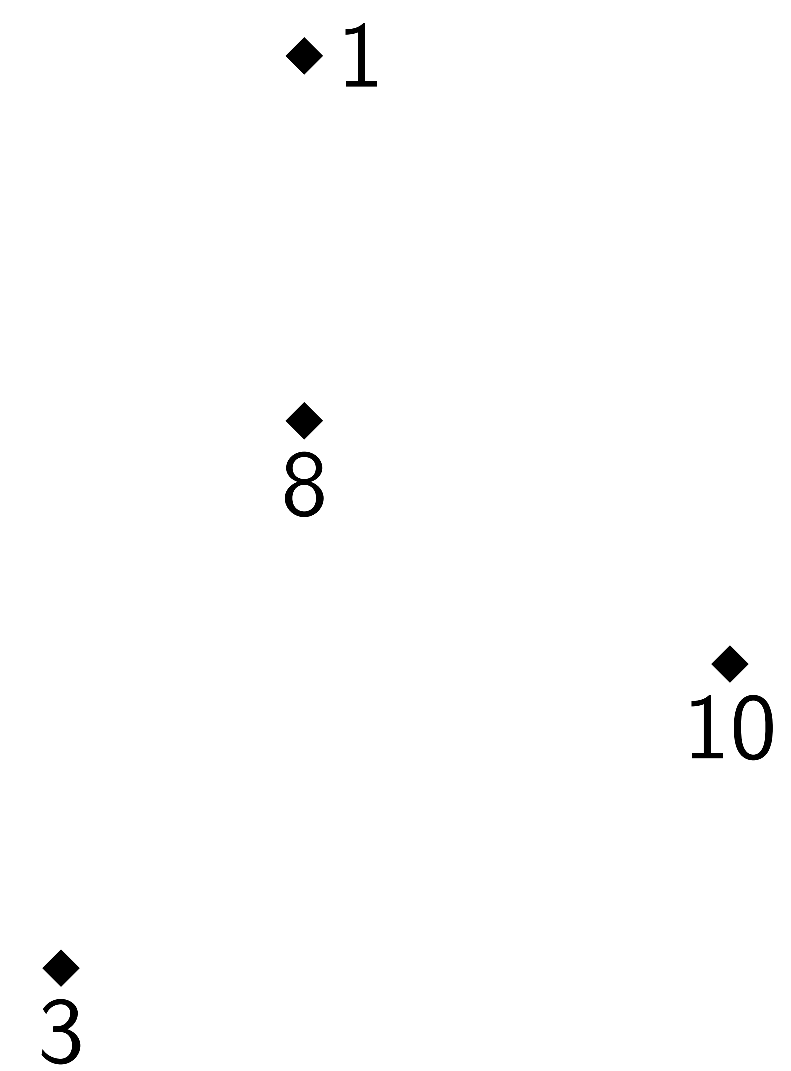
Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points
- apply 1D wavelet transform along the path
  low pass coefficients $(3, 10, 8, 1)$
  high pass coefficients $(1, 2, 2, 1)$
- update point set
- apply shrinkage to wavelet coefficients
- relate low pass coefficients to the updated point set



Denoising scheme (wavelet decomposition and shrinkage)

- find path through all points
- apply 1D wavelet transform along the path
  low pass coefficients $(3, 10, 8, 1)$
  high pass coefficients $(1, 2, 2, 1)$
- update point set
- apply shrinkage to wavelet coefficients
- relate low pass coefficients to the updated point set
- continue at the next level



Adaptive path reconstruction

- Choose first path index $p(1)$ randomly from $\Gamma := \{1, \ldots, N\}$.
- For $k = 1, \ldots, N-1$ choose $p(k+1)$ such that

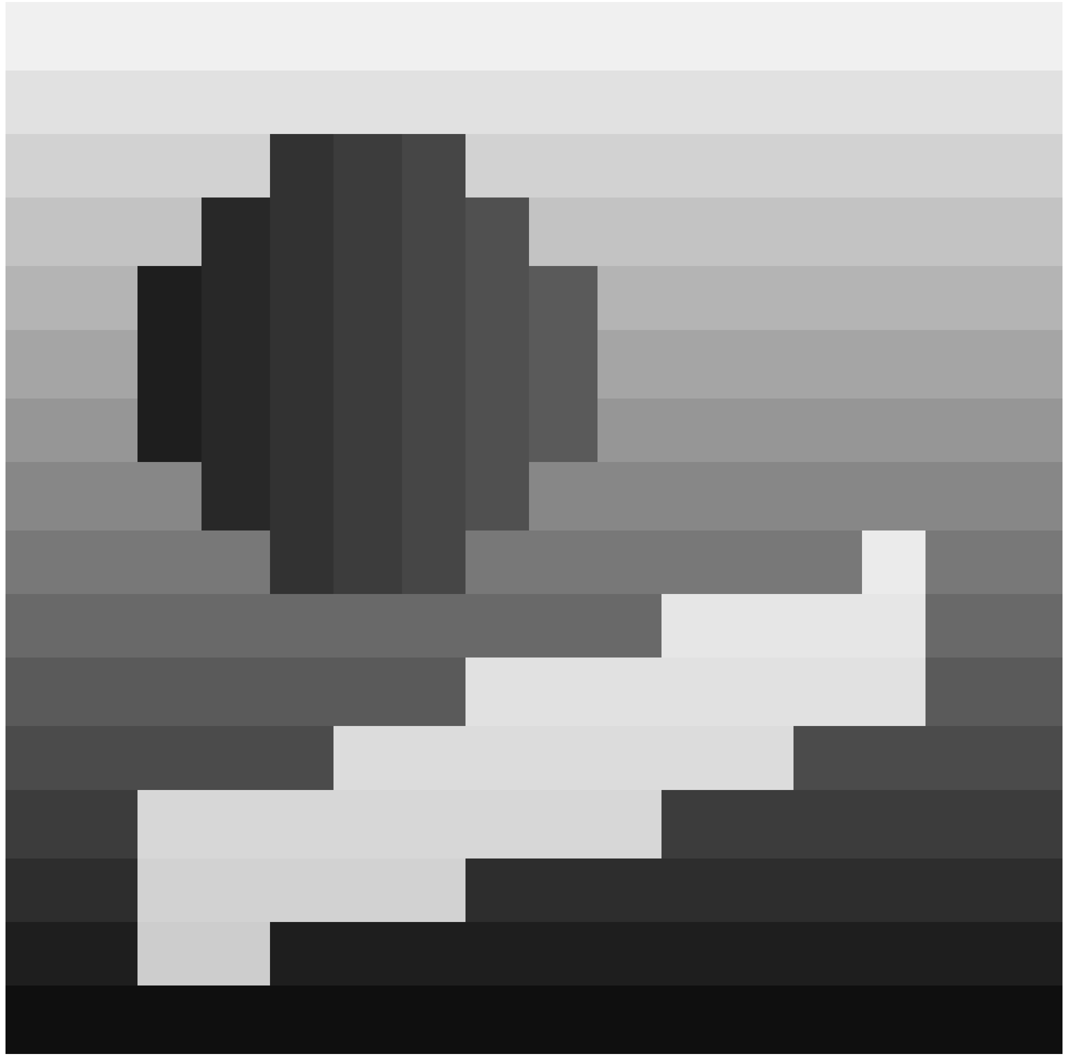
$$x_{p(k+1)} = \operatorname*{argmax}_{x \in N_{C,\theta}(x_{p(k)})} \frac{\langle x_{p(k)} - x_{p(k-1)}, x - x_{p(k)} \rangle}{\|x_{p(k)} - x_{p(k-1)}\| \cdot \|x - x_{p(k)}\|}$$

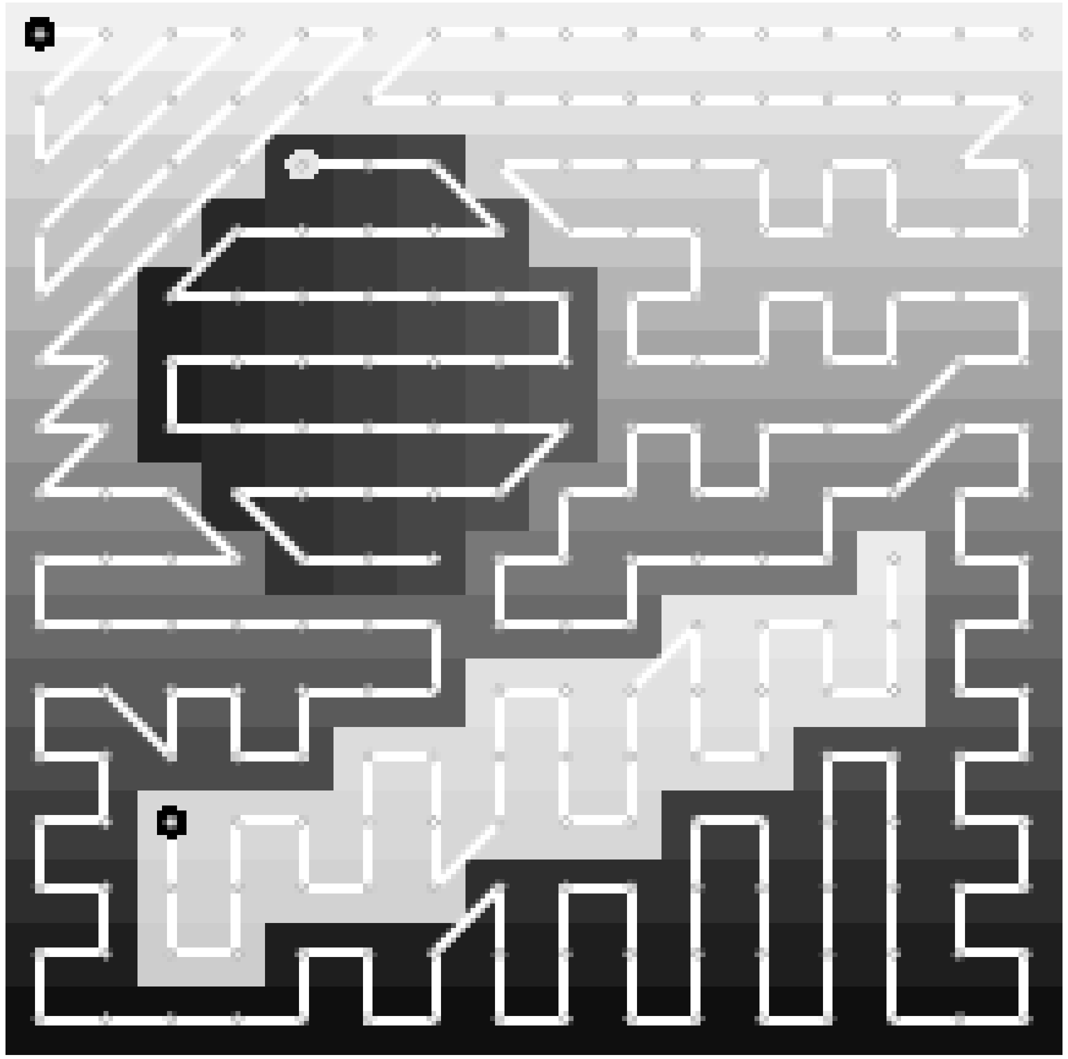
where $N_{C,\theta}(x_{p(k)})$ contains all points $x_r \in \Gamma$ fulfilling:

1. $r \notin \{p(1), \ldots, p(k)\}$
2. $\|x_r - x_{p(k)}\|_2 \leq C$
3. $|f(x_r) - f(x_{p(k)})| \leq \theta$.

If $N_{C,\theta}(x_{p(k)}) = \emptyset$, randomly choose $p(k+1)$ among the indices fulfilling $1 \,\&\, 2$ or only 1.

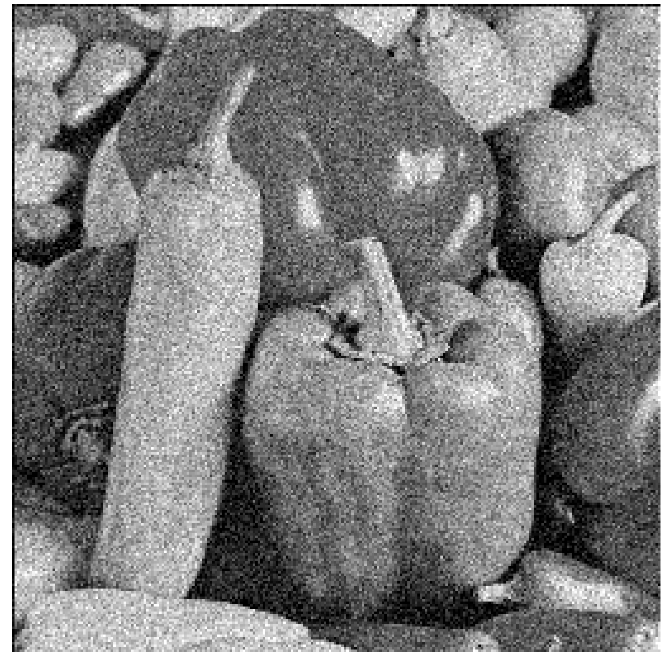
Example: Adaptive path reconstruction





$\sigma = 0.1$

Original image



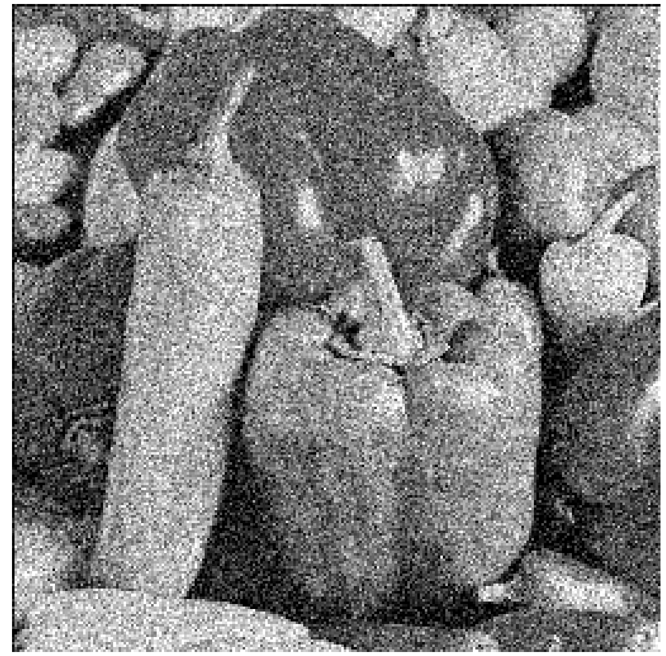
noisy image
PSNR= 19.97

adaptive path constr.
PSNR 29.01

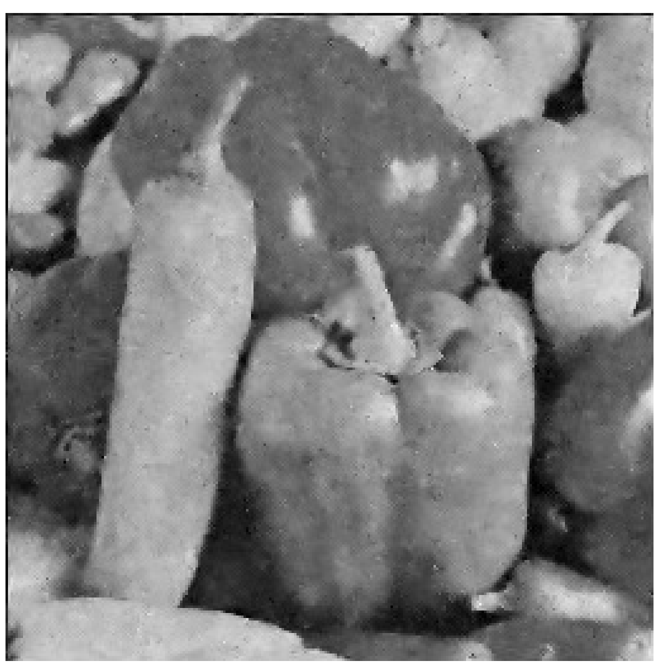
random path constr.
PSNR = 27.96

$\sigma = 0.15$

Original image



noisy image
PSNR= 16.45



adaptive path constr.
PSNR 26.44

random path constr.
PSNR = 25.69

Comparison of denoising results

	peppers	peppers	cameraman	cameraman
noisy image	19.97	16.45	19.97	16.45
tensor product wavelet shrinkage	24.91	23.20	24.74	22.86
with cycle spinning	28.11	25.86	27.19	25.14
4-pixel scheme	28.26	26.13	27.64	25.73
curvelet shrinkage	26.36	23.95	25.48	23.73
shearlet shrinkage	26.82	25.04	26.07	24.23
deterministic path	**29.01**	**26.44**	**28.28**	**26.15**
random path	27.96	25.69	27.44	24.85

Denoising of non-rectangular domains

Original image

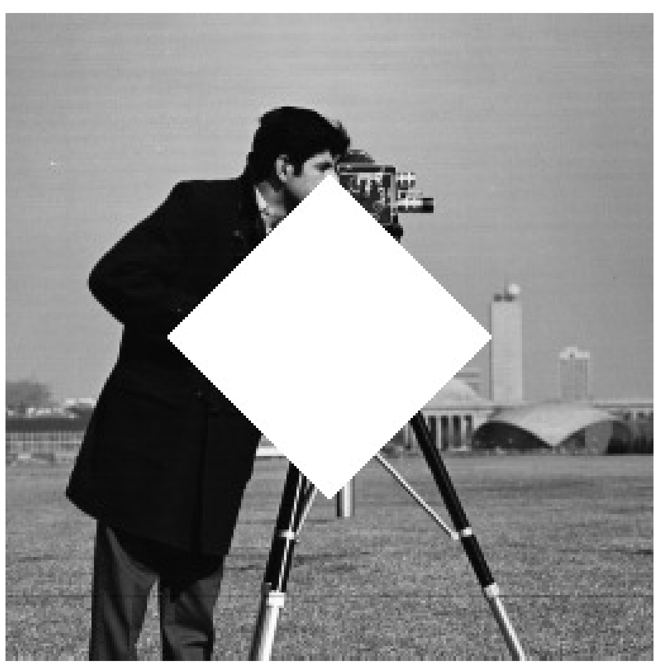
noisy image
PSNR= 19.97

denoised image
PSNR=27.77

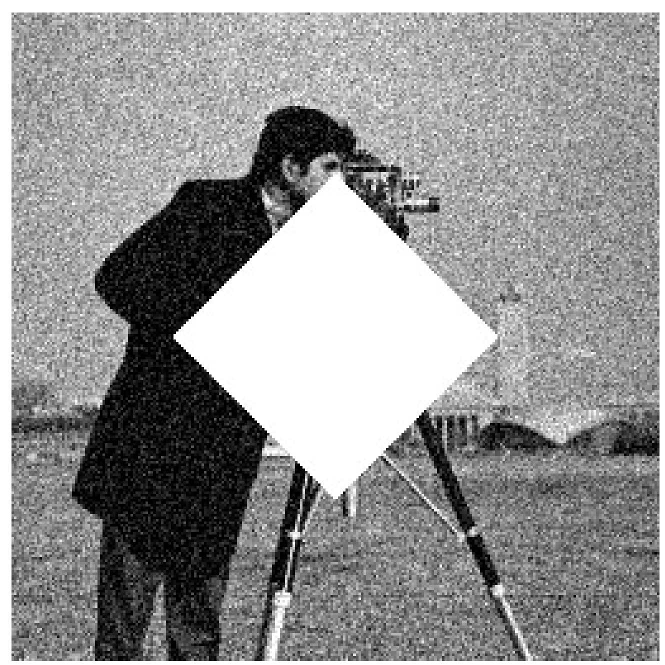
Original image

noisy image
PSNR= 19.98

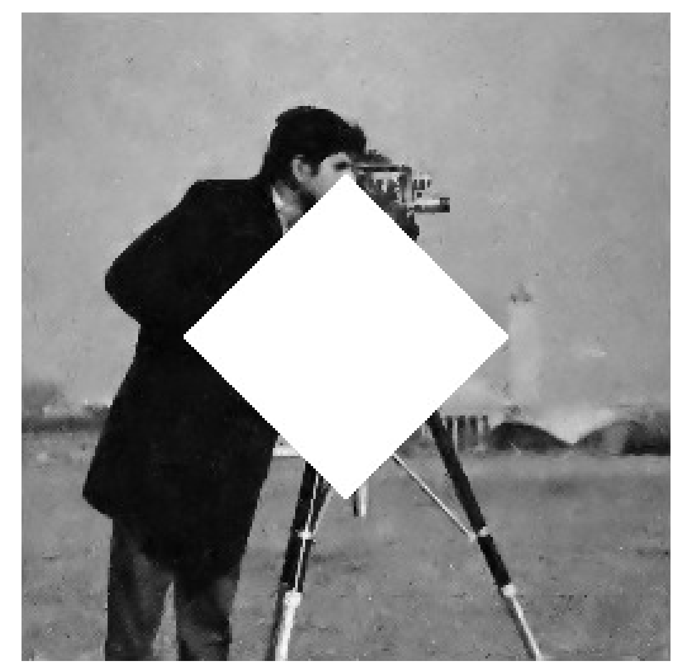
denoised image
PSNR=26.31



Original image



noisy image
PSNR=19.96



denoised image
PSNR = 28.71

Our publications

- Gerlind Plonka.
  The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data.
  SIAM Multiscale Modeling and Simulation 7(3) (2009), 1474-1496.
- Gerlind Plonka, Daniela Roşca.
  Easy Path Wavelet Transform on triangulations of the sphere.
  Mathematical Geosciences 42(7) (2010), 839-855.
- Jianwei Ma, Gerlind Plonka, Hervé Chauris.
  A new sparse representation of seismic data using adaptive easy-path wavelet transform.
  IEEE Geoscience and Remote Sensing Letters 7(3) (2010), 540-544.
- Gerlind Plonka, Stefanie Tenorth, Daniela Roşca.
  A hybrid method for image approximation using the easy path wavelet transform.
  IEEE Trans. Image Process. 20(2) (2011), 372-381.

- Gerlind Plonka, Stefanie Tenorth, Armin Iske.
  Optimally sparse image representation by the easy path wavelet transform.
  International Journal of Wavelets, Multiresolution and Information Processing 10(1) (2012), 1250007 (20 pages).
- Dennis Heinen, Gerlind Plonka.
  Wavelet shrinkage on paths for denoising of scattered data.
  Results in Mathematics 62(3) (2012), 337-354.
- Gerlind Plonka, Armin Iske, Stefanie Tenorth.
  Optimal representation of piecewise Hölder smooth bivariate functions by the easy path wavelet transform.
  Journal of Approximation Theory 176 (2013), 42-67.

\thankyou