Image Compression with Partial Differential Equations

Joachim Weickert
Mathematical Image Analysis Group
Saarland University, Saarbrücken, Germany

Thanks to

Irena Galić	(U. Saarbrücken)
Martin Welk	(U. Saarbrücken)
Bernhard Burgeth	(U. Saarbrücken)
Andrés Bruhn	(U. Saarbrücken)
Zakaria Belhachmi	(U. Metz)
Dorin Bucur	(U. Chambéry)
Egil Bae	(U. Bergen)
Alexander Belyaev	(U. Edinburgh)
Hans-Peter Seidel	(MPI Saarbrücken)

Introduction (1)

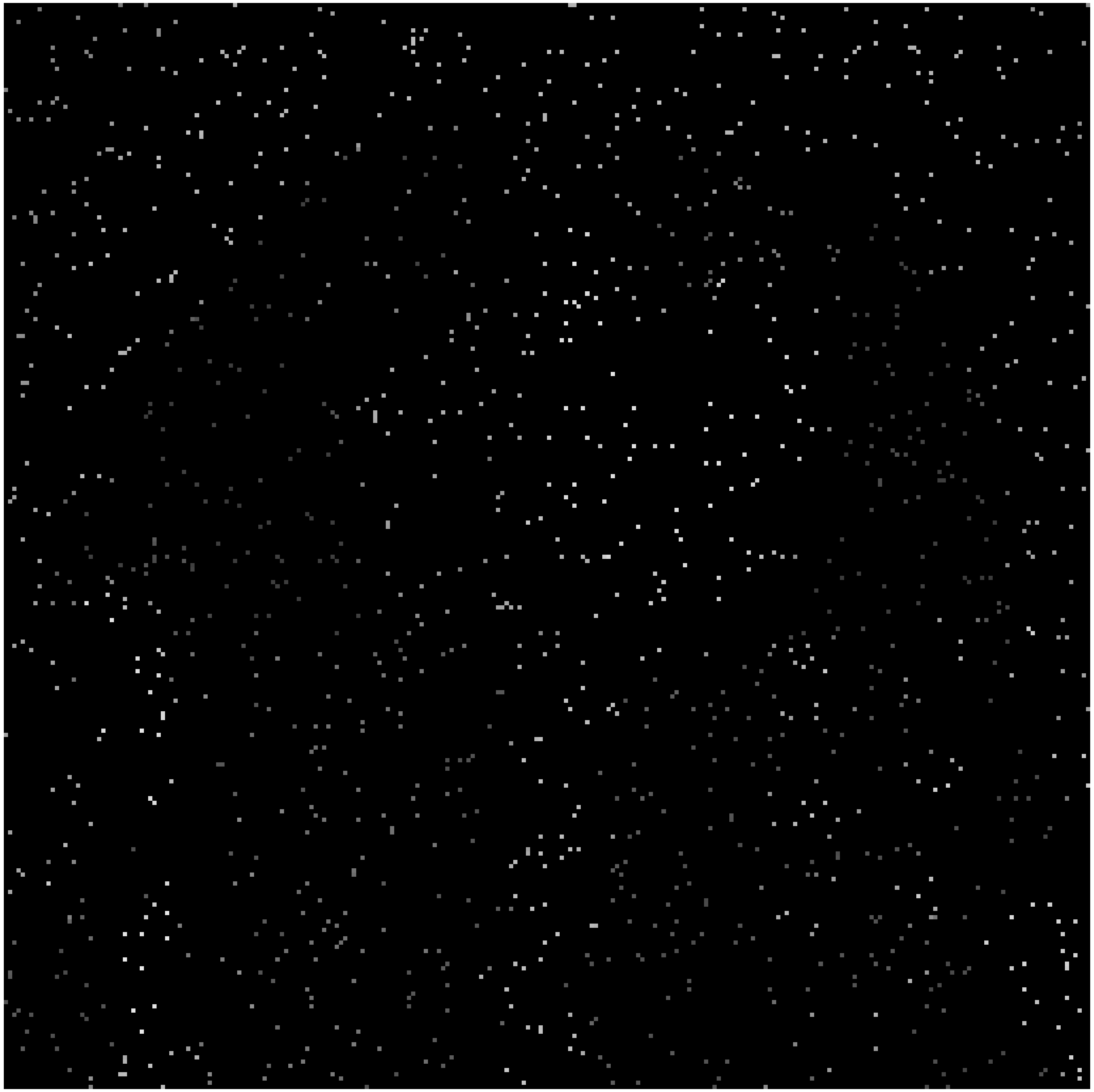
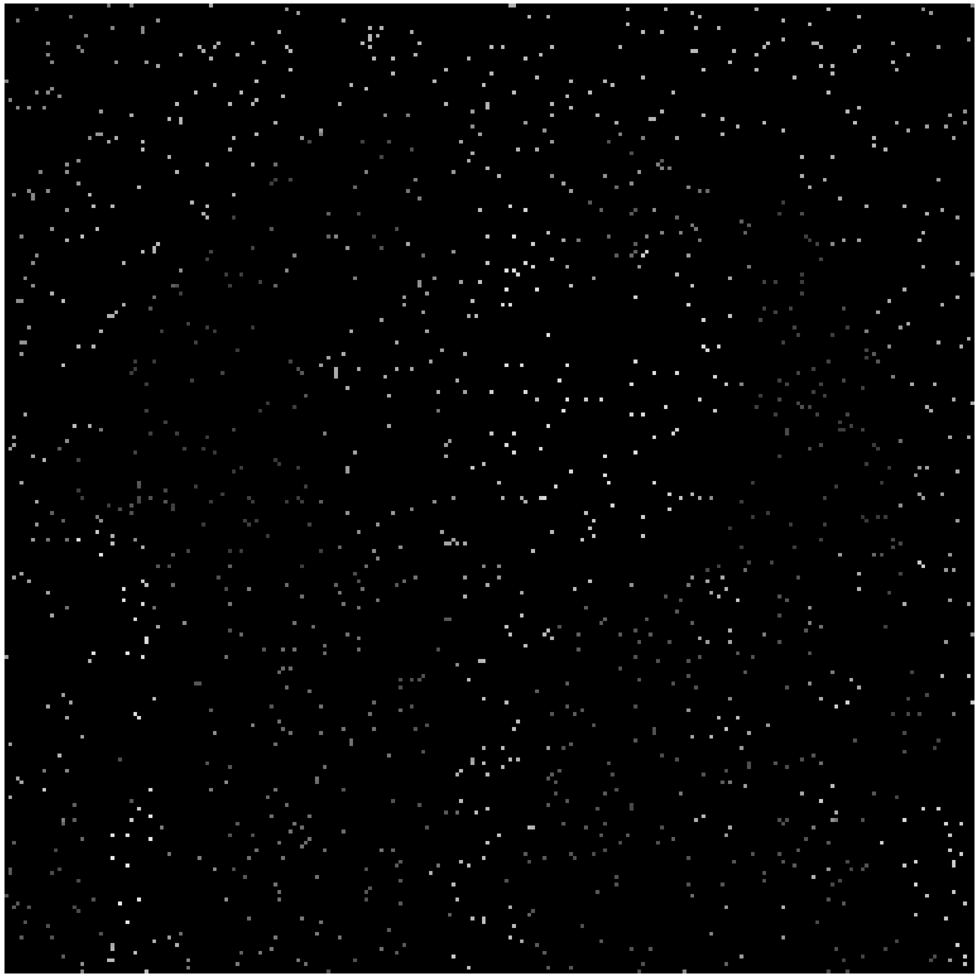


Image where only 2 percent of all pixels are known. Can you recognise what is depicted ?



Left: Original image. **Right:** After using an interpolation method where an anisotropic diffusion equation is solved.

Introduction

Image Compression

- The huge size of digital imagery requires lossy compression techniques.
- quasi-standard: JPEG
  - decomposes image into 8×8 pixel blocks
  - frequency decomposition within each block (DCT)
  - higher fequencies are coded with less precision (coarser quantisation)
  - quality deteriorates significantly for high compression rates
- need for methods that perform well for high compression rates (e.g. JPEG2000)
- our goal: analyse the potential of diffusion-like differential equations for image compression

Introduction (4)

(a) Top left: Original image, 256×256 pixels, 8 bits-per-pixel (bpp). **(b) Top right:** JPEG compression to 0.8 bpp (compression rate 10:1) gives rather good quality. **(c) Bottom left:** At 0.4 bpp (compression rate 20:1) visible deteriorations appear. **(d) Bottom right:** Severe block artifacts at 0.2 bpp (compression rate 40:1). Author: I. Galić (2006).

Some Related Work

- **PDE-Based Inpainting**

  Masnou/Morel 1998, Bertalmío et al. 2000, Chan/Shen 2000, ...

- **PDE-Based Interpolation**

  Caselles et al. 1998, Malgouyres/Guichard 2001, Tschumperlé/Deriche 2005, ...

- **PDE-Based Reconstruction from Image Features**

  Johansen et al. 1986, Hummel/Moniot 1986, Carlsson 1988, Desai et al. 1996, Elder 1999, Lillholm et al. 2003, Kanters et al. 2005

- **PDEs for Image Compression**

  mostly PDE-based pre- or postprocessing, but also

  - interpolation of digital elevation maps: Solé et al. 2004
  - PDE-improved wavelet thresholding: Chan/Zhou 2000
  - inpainting within JPEG: Liu et al. 2007, Xiong et al. 2007

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Basic Idea

- **Encoding**:
  - store image $f : \Omega \to \mathbb{R}$ only in some small subset $K \subset \Omega$.

- **Decoding**:
  - In the interpolation data set K, the reconstructed image u is known exactly:

$$u(x) = f(x).$$

  - In the unknown areas $\Omega \setminus K$, it solves the PDE

$$Lu = 0$$

    with some elliptic differential operator L.

Problems

- Which PDEs give good interpolation results?
- Which pixels are relevant for compression?
- How can these pixels be coded efficiently?

These problems cannot be solved in an isolated way:

- optimal point set depends on PDE:
  - good PDEs can cope with bad points
  - good points allow simple (suboptimal) PDEs
- suboptimal point set can pay off if coded efficiently

Good News on Pixel Selection

- discrete problem is finite:
  global optimum w.r.t. the best pixels for a given compression rate exists

Bad News

- combinatorically hopeless:
  selecting the best 10 percent pixels of a 256×256 image offers

  $$\binom{65536}{6554} \approx 4.3 \cdot 10^{9249}$$

  options.

- suboptimal approximation algorithms required

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

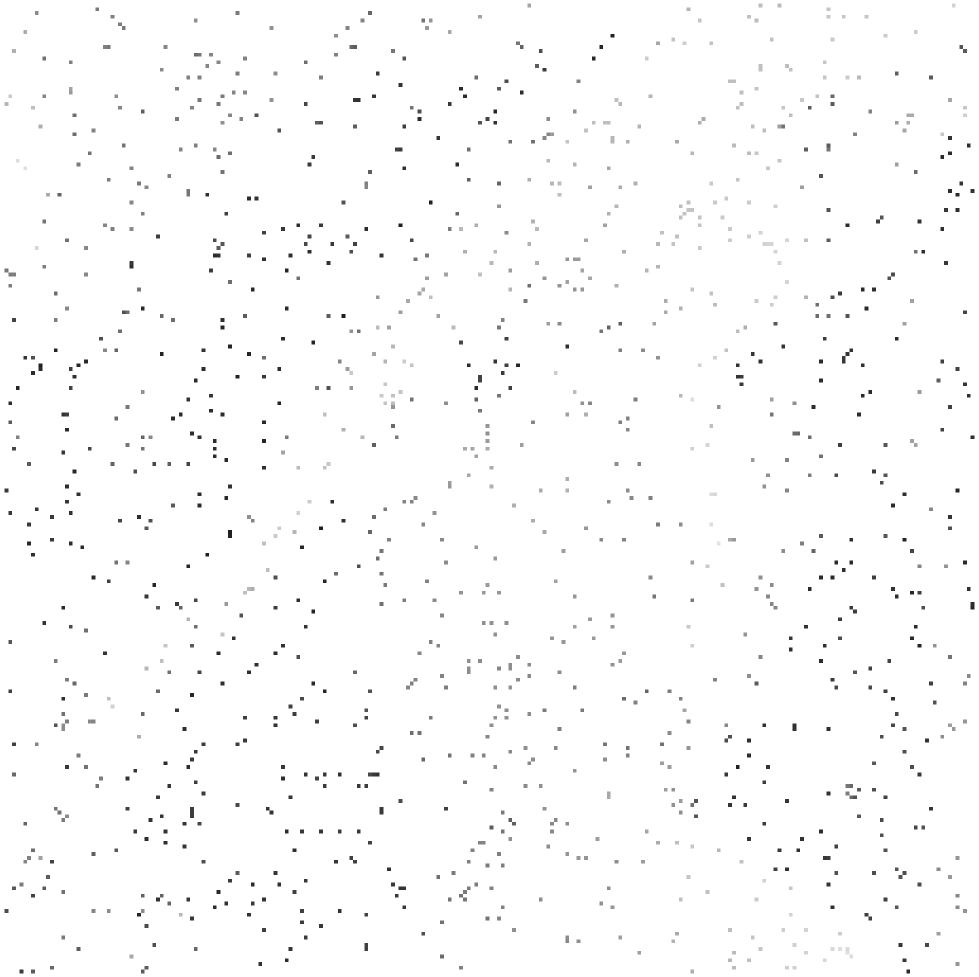
Which PDEs are Good?

name	differential operator Lu	max-min principle
linear diffusion	Δu	yes
biharmonic smoothing	$-\Delta^2 u$	no
triharmonic smoothing	$\Delta^3 u$	no
AMLE	$\partial_{\eta\eta} u$	yes
isotropic nonlinear diffusion	$\operatorname{div}\left(g(\lvert\nabla u\rvert^2)\,\nabla u\right)$	yes
anisotropic nonlinear diffusion	$\operatorname{div}\left(D(\nabla u_\sigma)\,\nabla u\right)$	yes

- η: normalised gradient direction ∇u
- g is some decreasing function such as the Charbonnier diffusivity

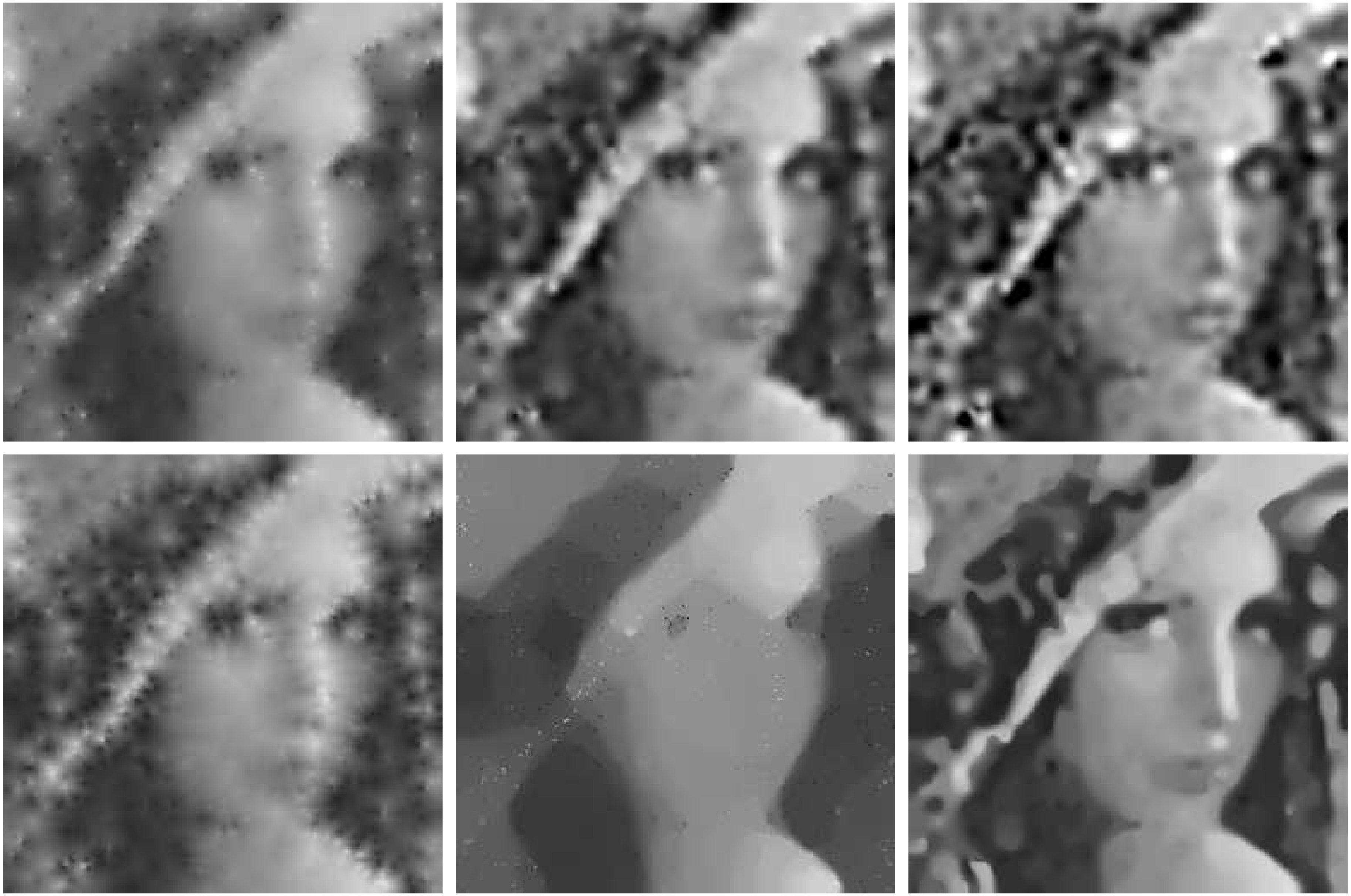
$$g(|\nabla u|^2) = \frac{1}{\sqrt{1+|\nabla u|^2/\lambda^2}}$$

- u_σ is a Gaussian-smoothed version of u.
- The diffusion tensor $D(\nabla u_\sigma)$ smoothes along the edge direction ∇u_σ with eigenvalue 1, and across the edge with eigenvalue $g(|\nabla u_\sigma|^2)$.
- This specific anisotopic diffusion model is called *edge-enhancing diffusion (EED)*.



(a) **Left:** Zoom into the test image *lena*, 256×256 pixels. (b) **Right:** Grey values of the scattered interpolation points (2 percent of all pixels, chosen randomly)

Which PDEs are Good? (3)



Impact of the PDE on the interpolation result. (a) **Top left:** Interpolation by linear diffusion. (b) **Top middle:** Biharmonic smoothing. (c) **Top right:** Triharmonic smoothing. (d) **Bottom left:** AMLE. (e) **Bottom middle:** Isotropic nonlinear diffusion. (f) **Bottom right:** EED.

Quantitative Evaluation

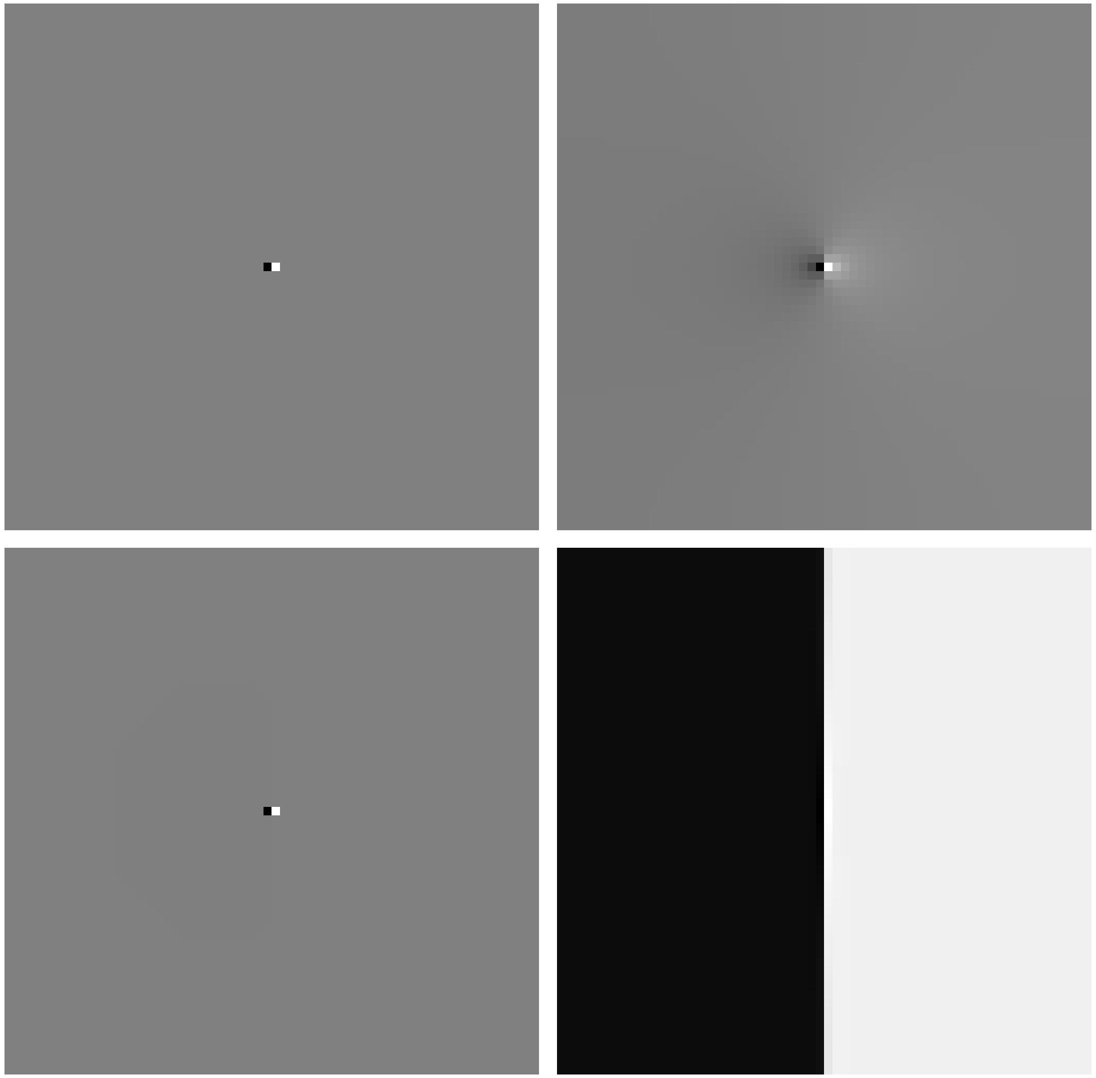
Average absolute error between interpolated image u and exact image f:

$$\mathrm{AAE}\,(u,f) \;:=\; \frac{1}{N}\sum_{i,j} |u_{i,j} - f_{i,j}|$$

PDE method	AAE
linear diffusion	16.98
biharmonic smoothing	15.79
triharmonic smoothing	18.69
AMLE	17.33
isotropic nonlinear diffusion	21.80
EED	14.58

EED performs best.

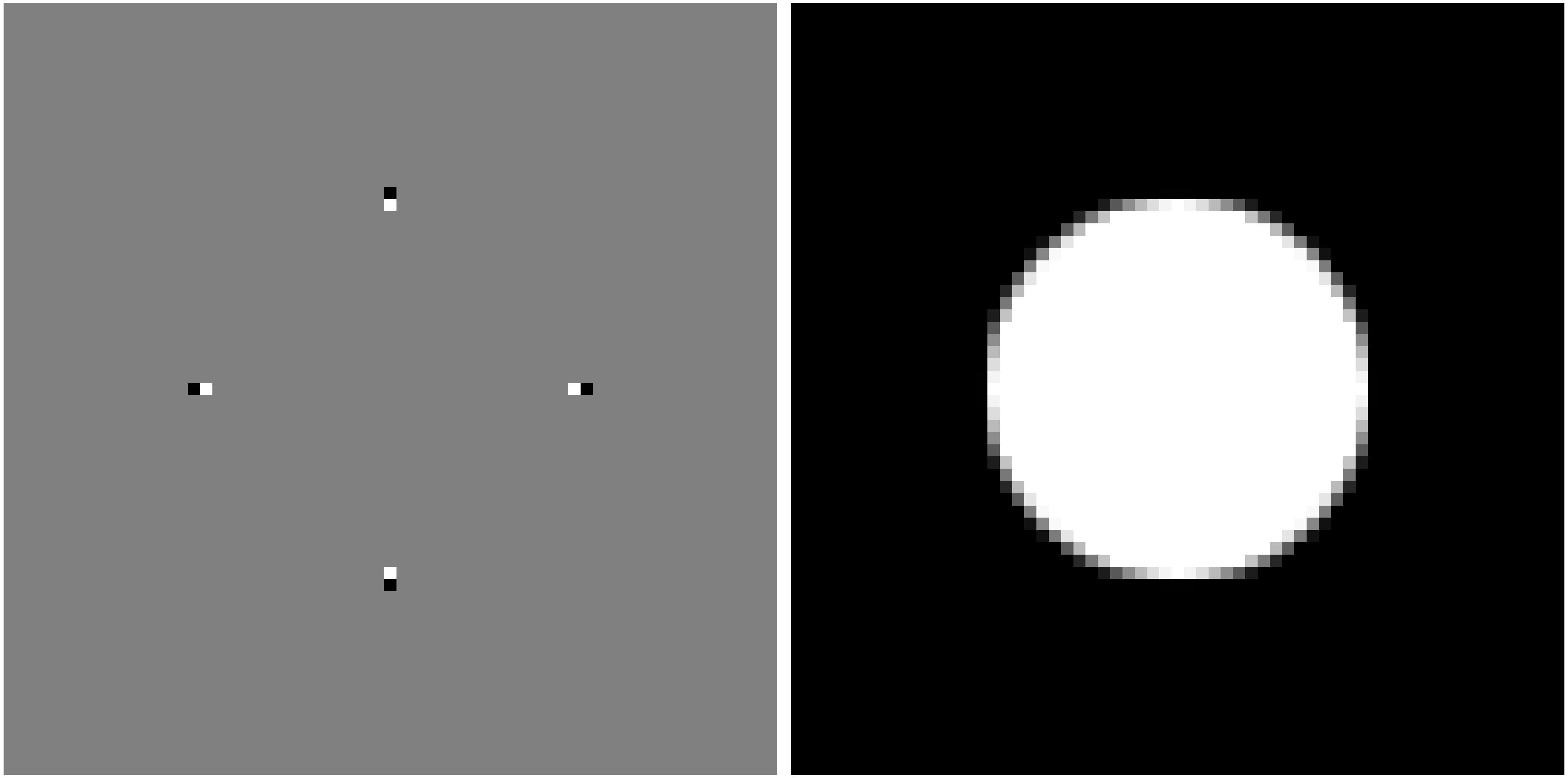
Why Does Anisotropic Nonlinear Diffusion Work so Well ?



(a) **Top left:** Original image with dipole data (64×63 pixels). (b) **Top right:** Interpolation by homogeneous diffusion. (c) **Bottom left:** Isotropic nonlinear diffusion ($\lambda = 0.01$). (d) **Bottom right:** Anisotropic nonlinear diffusion ($\sigma = 1$, $\lambda = 0.01$).

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42

Which PDEs are Good? (6)



(a) **Left:** Original image with four dipoles (64×64 pixels). (b) **Right:** Interpolation by anisotropic nonlinear diffusion ($\sigma = 1$, $\lambda = 0.01$).

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Method 1: Semantic Approach

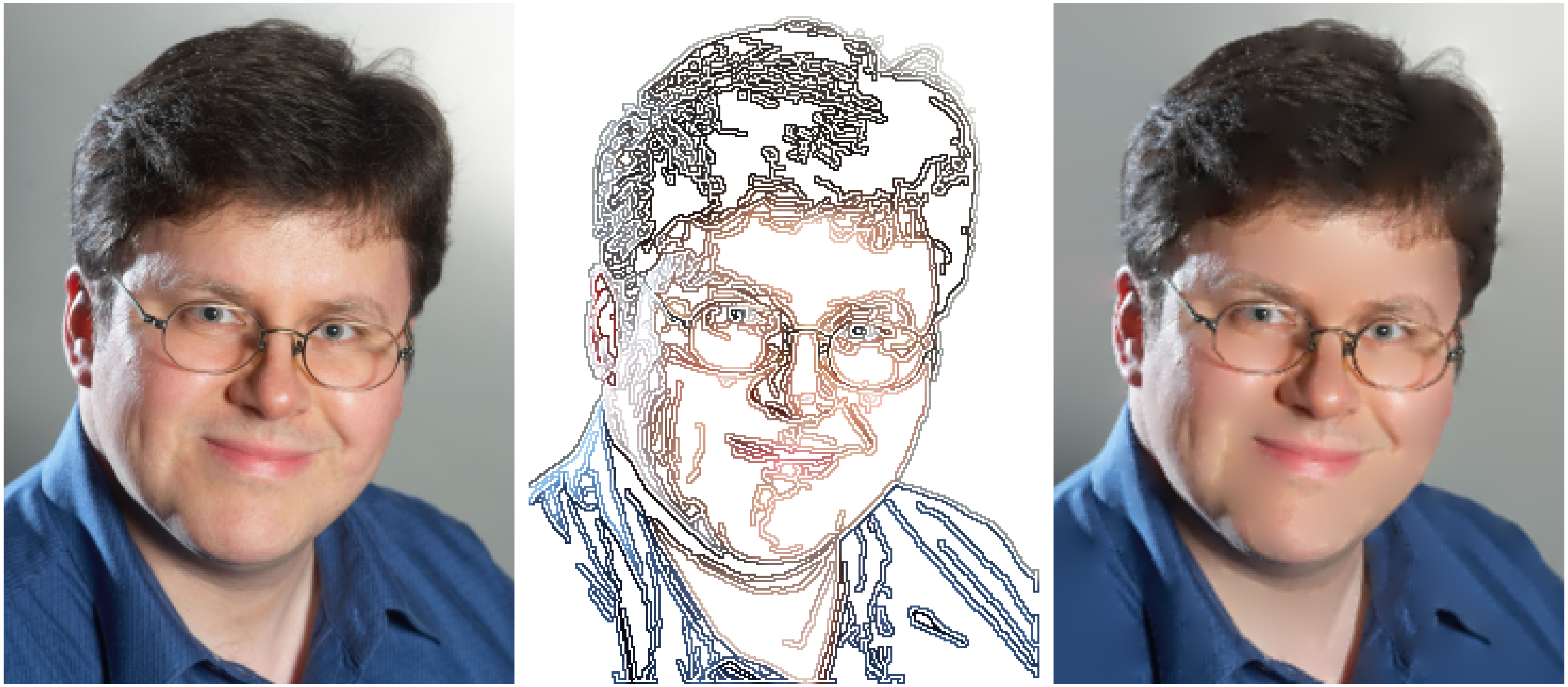
- Edges are relevant for understanding the image content.

- choose gradient information at edges as interpolation data:
  - apply a standard edge detector (e.g. Canny detector)
  - use pixels left and right to the edge as interpolation points

- properties:

  $\oplus$ fast coding step: edges are inexpensive to compute
  $\oplus$ nice results even for linear diffusion (works better for lines than for points)
  $\oplus$ fast decoding due to linear diffusion
  $\ominus$ bad compression rates (contours require much storage)

Method 1: Semantic Approach (2)



Interpolation for image reconstruction from the edge information. **Left:** Original image, 240×320 pixels. **Middle:** Canny-like edge set. Only the left and right neighbours of each edge are stored. **Right:** Interpolation by linear diffusion.

Outline

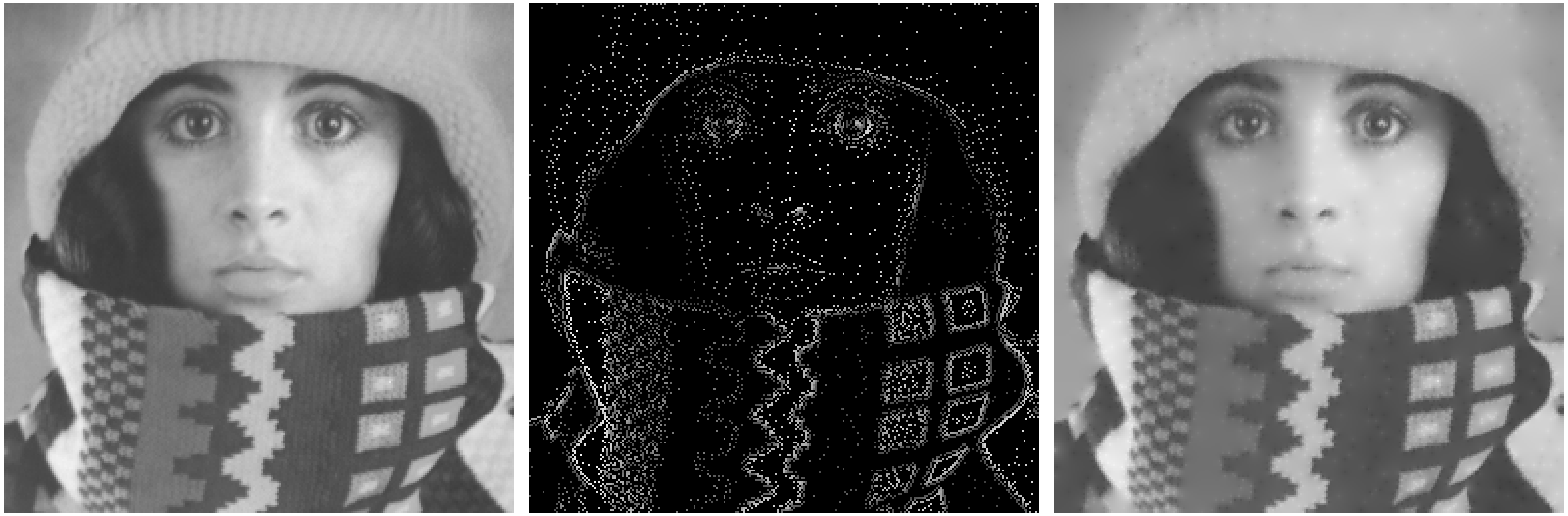
- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Method 2: Probabilistic Optimisation Approach

- tries to find a good local optimum by sparsifying the points
- start with full image
- repeat
  - remove p percent of all pixels (e.g. 5 percent)
  - interpolate with PDE-based method
  - return the percentage $(1-p)p$ where the interpolation error is largest

  until the desired sparsification is reached
- properties:
  - $\ominus$ slow coding step: expensive to compute
  - $\oplus$ nice results even for linear diffusion
  - $\oplus$ fast decoding due to linear diffusion
  - $\odot$ compression rates still not competitive

Method 2: Probabilistic Optimisation Approach (2)



(a) **Left:** Original image. (b) **Middle:** Selection of 10 % well-suited interpolation points by a probabilistic optimisation algorithm. (c) **Right:** Result using linear diffusion interpolation.

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Method 3: Analytic Approach

- use theory of shape optimisation to find optimal interpolation data in a continuous model
- properties:
  $\oplus$ true optimality results
  $\ominus$ mathematically involved and only for linear diffusion
  $\oplus$ fast coding step
  $\oplus$ fast decoding due to linear diffusion
  $\oplus$ fairly nice results
  $\odot$ compression rates still not competitive
- different shape optimisation formulations investigated
- important feature in all models: pixels must have positive area
- suggest to choose the density of the data points as an increasing function of $|\Delta f|$
- close in spirit to semantic, edge-based ideas

Method 3: Analytic Approach (2)

Top left: Original image f, 257×257 pixels. **Top right:** $|\Delta f_\sigma|$ with $\sigma = 1$. **Bottom left:** Floyd-Steinberg dithering of $|\Delta f_\sigma|$ such that 10 % of all pixels are selected. **Bottom right:** Linear diffusion interpolation with the "dithered" data set.

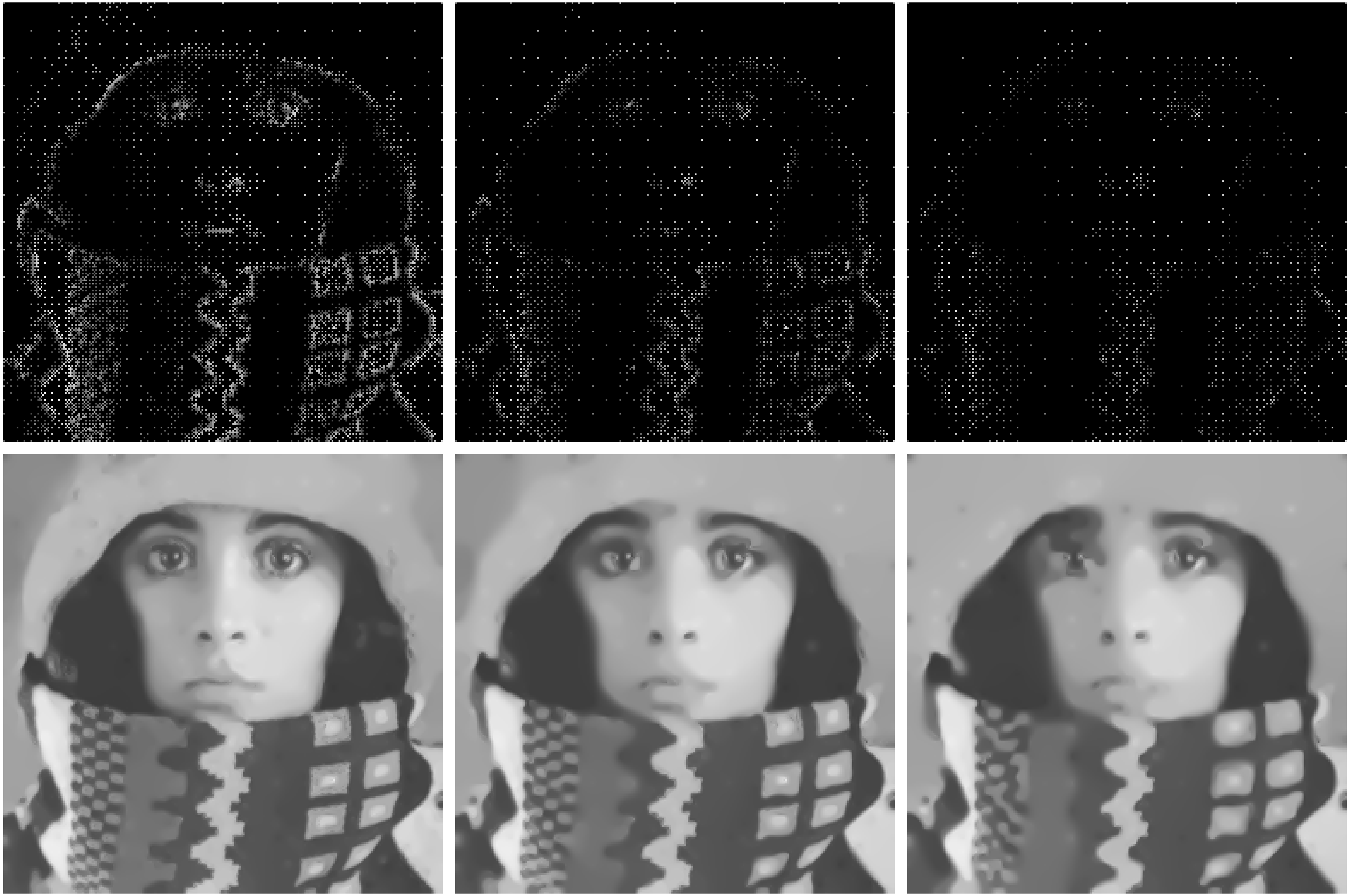
Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

Method 4: Constrained Point Set Approach

- previous approaches aimed at optimality but neglected costs for storing points (can be high if chosen arbitrarily)
- now: allow suboptimality and find method where storing points is cheap
- coding:
  - adaptive triangulation that can be coded in a binary tree (Distasi et al. 1997)
  - split area along one diagonal into two triangles
  - if plane on a each triangle approximates image not well enough: subdivide the triangle
- decoding by anisotropic diffusion (EED)
- properties:
  ⊕ fast and compact coding (but rather suboptimal points)
  ⊖ expensive decoding: suboptimal points require sophisticated interpolant (EED)
  ⊕ good compression rates, comparable to JPEG

Method 4: Constrained Point Set Approach (2)



Top row, left to right: Adaptive sparsification of *trui* with compression to 0.8 bpp, 0.4 bpp, 0.2 bpp.
Bottom row, left to right: Corresponding EED-based interpolation.

Method 4: Constrained Point Set Approach (3)

Comparison at high compression rate (40:1). (a) **Left:** Original image. (b) **Middle:** JPEG. (c) **Right:** EED-based compression.

Average Absolute Error

JPEG	EED
11.25	8.45

Method 4: Constrained Point Set Approach (4)

Further Improvements

- adaptive error thresholds for the different levels
- EED also in the coding step
- optimise also grey values
- requantisation from 256 to 32 grey levels

results clearly outperform JPEG at high compression rates

Method 4: Constrained Point Set Approach (5)

Comparison of BTTC-based compression methods at a compression rate of 40:1. **Top left:** Original image. **Top middle:** EED-based decoding. **Top right:** with adaptive threshold. **Bottom left:** with EED coding. **Bottom middle:** with grey value biasing. **Bottom right:** with post-quantisation.

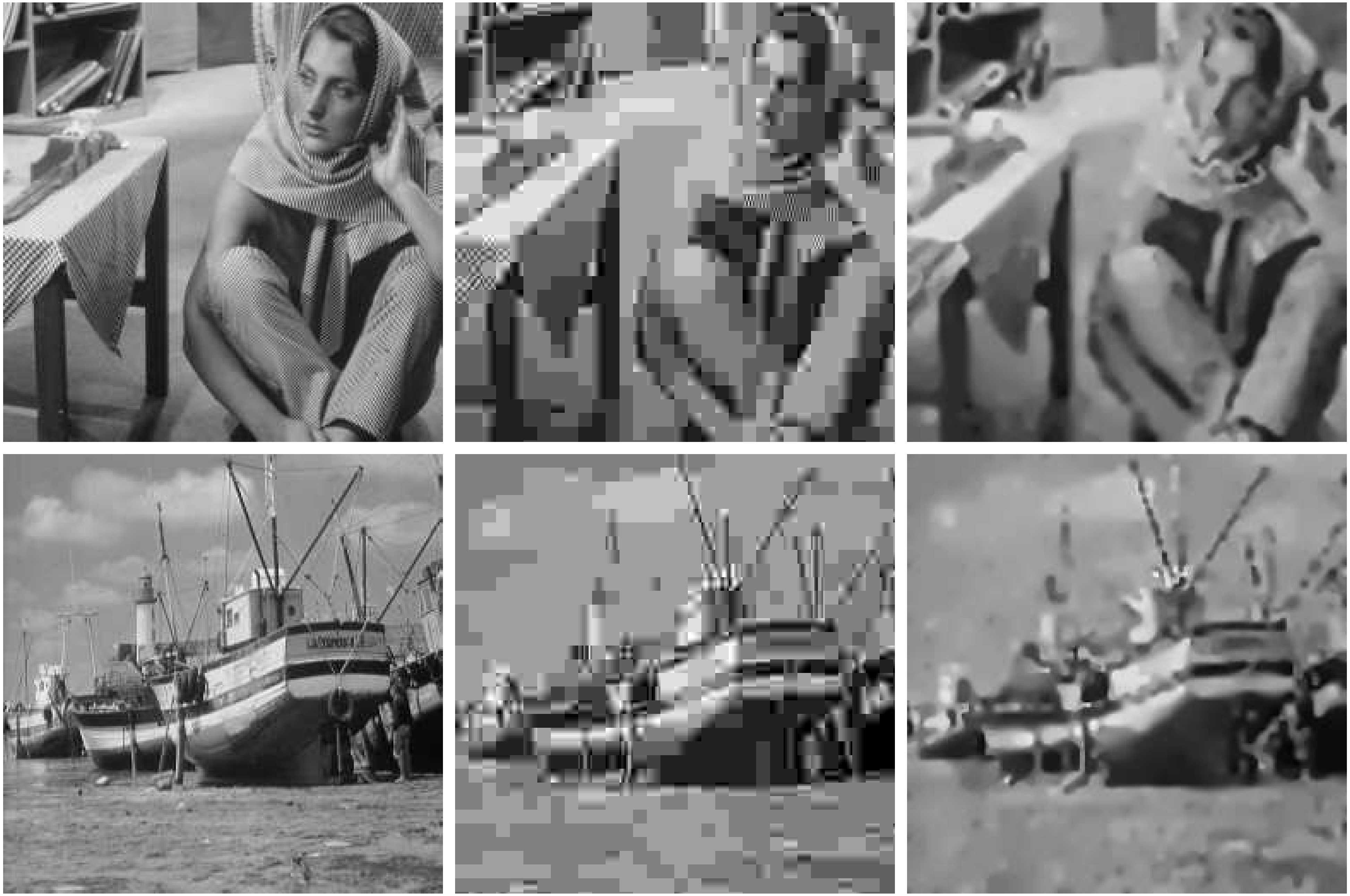
Method 4: Constrained Point Set Approach (6)

Comparison at high compression rates (40:1) for the test images *trui* and *lena*. **Left column:** Original images. **Middle column:** JPEG. **Right column:** EEDC.

Method 4: Constrained Point Set Approach (7)

Comparison at high compression rates (40:1) for the test images *cameraman* and *peppers*. **Left column:** Original images. **Middle column:** JPEG. **Right column:** EEDC.

Method 4: Constrained Point Set Approach (8)



Comparison at high compression rates (40:1) for the test images *barbara* and *boats*. **Left column:** Original images. **Middle column:** JPEG. **Right column:** EEDC.

Evaluation of the Average Absolute Error

Image	JPEG	EEDC
trui	11.25	4.99
lena	13.61	8.72
cameraman	13.75	9.38
peppers	12.19	7.53
barbara	15.47	12.22
boats	14.68	11.82

Outline

- Basic Idea
- Which PDEs are Good ?
- Method 1: Semantic Approach
- Method 2: Probabilistic Optimisation Approach
- Method 3: Analytic Approach
- Method 4: Constrained Point Set Approach
- Summary and Outlook

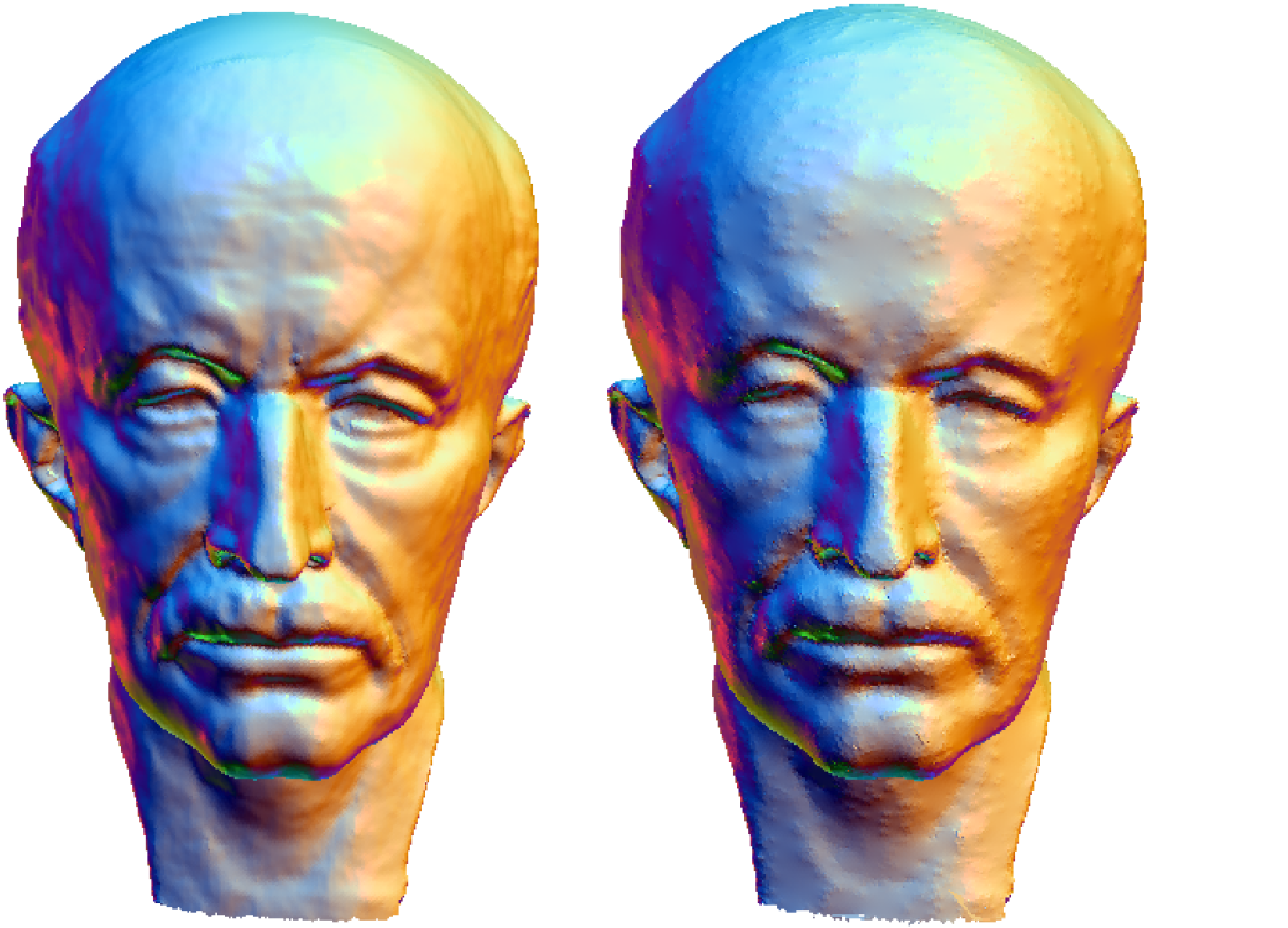
Summary

- considered PDE-based image compression that drives inpainting to the extreme
- anisotropic nonlinear diffusion (EED) performs favourably
- considered four strategies for selecting good interpolation points:
  - use visually relevant features such as edges
  - heuristic sparsification
  - based on the mathematics of shape optimisation
  - with points constrained to a B-tree
- advanced methods clearly outperform JPEG at high compression rates

Ongoing and Future Work

- well-posedness and regularity results for EED in the continuous setting ?
- better performing alternatives to EED ?
- fast numerical solvers
- extensions to image sequences and surface data

Outlook (2)



(a) **Left:** Original data set. (b) **Right:** Reconstruction with 10 % of the points and geometric linear diffusion.

References

- Z. Belhachmi, D. Bucur, B. Burgeth, J. Weickert: How to choose interpolation data in images. Technical Report No. 205, Dept. of Mathematics, Saarland University, 2008. Submitted to *SIAM Journal on Applied Mathematics*, 2008. *(shape optimisation framework for linear diffusion)*

- I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, H.-P. Seidel: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, Vol. 31, 255–269, 2008. *(adaptive triangulations with EED-based interpolation)*

Download: `www.mia.uni-saarland.de`

Thank you for your attention!