Lecture 15: PDE-Based Image Interpolation

Contents

Motivation

What is Interpolation ?

- recovery of missing data from incomplete data within the range given by these data
- has to involve additional assumptions on the data, e.g. smoothness, band limitation
- distinguish from
  - *extrapolation:*
    – uses a model *outside* the given range
    – example: weather forecast
  - *approximation:*
    – model does not reproduce the given data exactly
    – example: regression curve through noisy data

Motivation (2)

M I A
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

Where Do Interpolation Problems Occur ?

- rescaling, zooming: modify the sampling rate of pixels
  (e.g. in so-called digital zooms in digital cameras)
- rotation of digital images by nontrivial angles
- inpainting problems: most data given; fill in missing data in corrupted areas
- demosaicing: Most CCD sensors in digital cameras only give either a red, green or blue value at each pixel.
- reslicing: display a (medical) 3-D data set along planes that do not coincide with planes along the axis directions
- interpolation-based image compression
- many computer vision problems (e.g. warping)
- scattered data interpolation
- computer aided geometric design and geometric modelling

Motivation (3)

M I A
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

What are the Most Common Interpolation Methods ?

- Sometimes very simple approaches such as linear interpolation are sufficient.
- Spline interpolation:
  - represents data by piecewise polynomials
  - cubic splines (using cubic polynomials) are good compromise between simplicity and high quality.
- More recently, PDEs and variational methods are used, in particular for inpainting problems.

Goals of This Lecture

- give an introduction to PDE-based interpolation
- generalises classical spline interpolation and variational image regularisation
- apply it to zooming and image compression

Motivation (4)

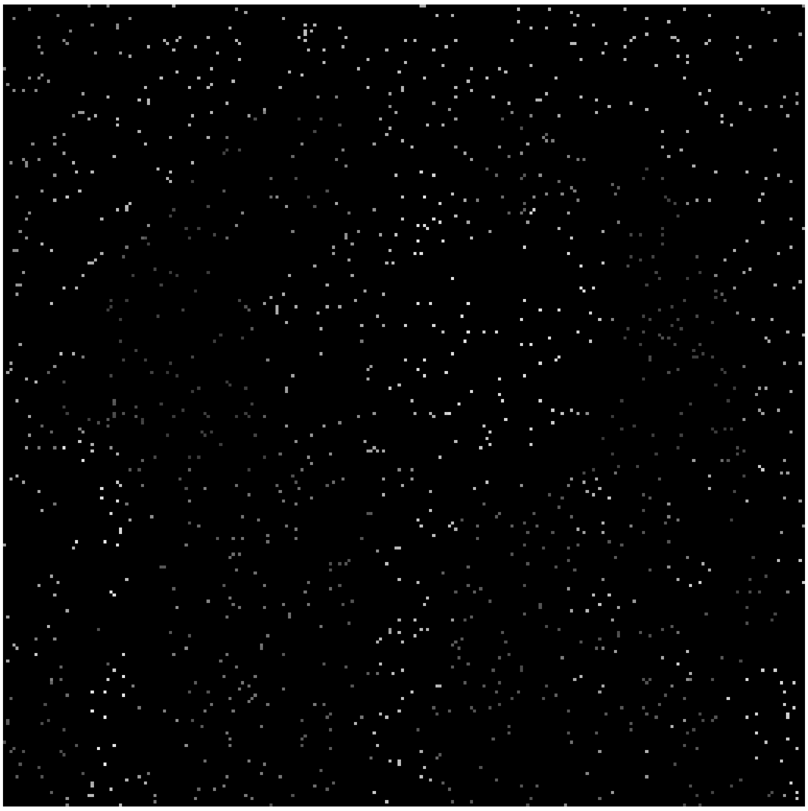
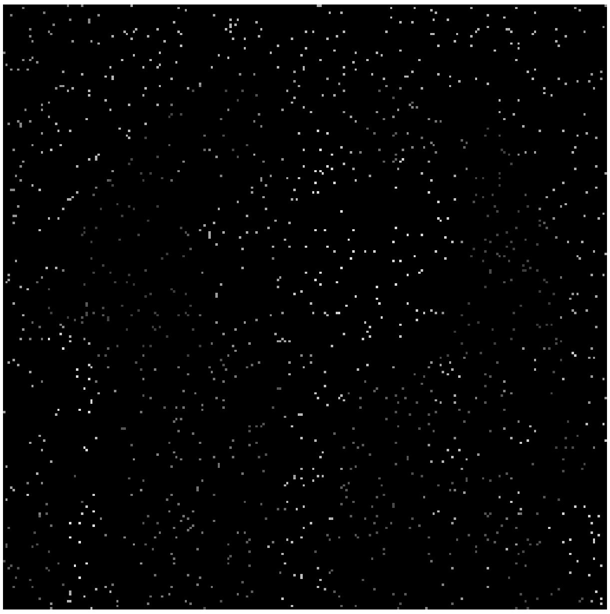


Image where only 2 percent of all pixels are known. Can you recognise what is depicted ?

Motivation (5)



Left: Original image. **Right:** After using a PDE-based interpolation method.

M I A
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

A General Model for Interpolation and Approximation

One-Dimensional Spline Interpolation

- Given: points $0 = x_1 < x_2 < \ldots < x_n = 1$ with function values $f(x_1), \ldots, f(x_n)$.
  Wanted: smooth function $u(x) : [0,1] \to \mathbb{R}$ that minimises

$$E(u) = \int_0^1 \left(\frac{d^m u}{dx^m}\right)^2 dx$$

subject to

$$u(x_i) = f(x_i) \qquad (i = 1,\ldots,n).$$

- $m = 1$: linear interpolation with C^0-smoothness at interpolation points
- $m = 2$: cubic spline interpolation with C^2-smoothness
- $m = 3$: quintic spline interpolation with C^4-smoothness
- arbitrary $m \in \mathbb{N}$: spline interpolation of degree $2m-1$ with C^{2m-2}-smoothness

M I A
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

Euler-Lagrange Equation for Functionals with Higher-Order Derivatives

In order to reinterpret spline interpolation as PDE-based interpolation, we need an extension of the Euler–Lagrange theorem to functionals with higher-order derivatives:

A smooth function $u(x)$, $x \in [a,b]$ that minimises the 1-D energy functional

$$E(u) \;=\; \int_a^b F(x, u, u^{(1)}, \ldots, u^{(m)})\, dx$$

with $u^{(k)} := \frac{d^k u}{dx^k}$ satisfies necessarily the Euler–Lagrange equation

$$F_u + \sum_{k=1}^{m} (-1)^k \frac{d^k}{dx^k} F_{u^{(k)}} \;=\; 0.$$

The case $m = 1$ corresponds to our previously used Euler-Langrange theorem (Lecture 10).

M I A
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

PDE Interpretation of Spline Interpolation

- The interpolating function $u(x)$ must satisfy
  - the interpolation constraints $u(x_i) = f(x_i)$ for $i = 1,...,n$,
  - the Euler-Lagrange equation $(-1)^m \, \partial_{xx}^m u = 0$ outside $x_1,...,x_n$.

- formulation in a single equation:

$$c(x) \cdot \underbrace{(u(x) - f(x))}_{\text{interpolation}} \;-\; (1 - c(x)) \cdot \underbrace{(-1)^{m+1} \partial_{xx}^m u}_{\text{smoothness}} \;=\; 0$$

  with

$$c(x) := \begin{cases} 1 & \text{if } x \in \{x_0, ..., x_n\} \\ 0 & \text{else.} \end{cases}$$

- linear PDE of order $2m$
- very smooth solution for large m, but no maximum–minimum principle for $m > 1$
- case $m = 1$ not very exciting: linear interpolation

M I A
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

Regularisation

- Given some noisy signal f, find a signal u that minimises

$$E(u) = \int_\Omega \Big(c \cdot \underbrace{(u - f)^2}_{\text{similarity}} \;+\; (1 - c) \cdot \underbrace{\Psi(u_x^2)}_{\text{smoothness}} \Big) dx$$

  with some weight $0 < c < 1$ and an increasing penaliser Ψ.

- minimiser satisfies Euler–Lagrange equation

$$c \cdot \underbrace{(u - f)}_{\text{similarity}} \;-\; (1 - c) \cdot \underbrace{\partial_x \big(\Psi'(u_x^2)\, u_x\big)}_{\text{smoothness}} \;=\; 0$$

- nonlinear PDE of order 2, satisfies maximum–minimum principle
- nonlinear penaliser $\Psi(u_x^2)$ allows discontinuity preserving smoothing
- Example: TV penaliser $\Psi(u_x^2) = 2|u_x|$ yields diffusivity $g(u_x^2) = \Psi'(u_x^2) = \frac{1}{|u_x|}$.

A General Model for Interpolation and Approximation (5)

Unified Model

- Given: scalar N-dimensional image $f : \Omega \subset \mathbb{R}^N \to \mathbb{R}$
- Find a restoration u that satisfies

$$c(\mathbf{x}) \cdot \underbrace{(u - f)}_{\text{approximation}} \; - \; (1 - c(\mathbf{x})) \cdot \underbrace{Lu}_{\text{smoothness}} \; = \; 0$$

with a confidence function $c(\mathbf{x}) : \Omega \to [0, 1]$
and some elliptic differential operator L

M I A 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

A General Model for Interpolation and Approximation (6)

The Confidence Function $c(\mathbf{x})$

- Fills in missing data at locations $\mathbf{x}$ where $c(\mathbf{x}) = 0$.
- Reproduces data at locations $\mathbf{x}$ with $c(\mathbf{x}) = 1$.
- Interpolation for

$$c(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in \{\mathbf{x}_0, ..., \mathbf{x}_n\}, \\ 0 & \text{else.} \end{cases}$$

- also for scattered data interpolation
- regularisation at locations $\mathbf{x}$ with $0 < c(\mathbf{x}) < 1$
- For classical regularisation, c is fixed.
- However, $c(\mathbf{x})$ expresses the confidence in the data.
  Can be inversely proportional to local noise variance of f.
- Denoising with simultaneous filling-in of data also possible:
  Choose c such that $0 < c(\mathbf{x}) < 1$ in the noisy data points, and $c(\mathbf{x}) := 0$ elsewhere.

M I A 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

The Differential Operator L

smoothing strategy	differential operator Lu	max-min principle
linear	Δu	yes
isotropic nonlinear	$\text{div}\,(g(\|\nabla u\|^2)\,\nabla u)$	yes
anisotropic nonlinear	$\text{div}\,(g(\nabla u_\sigma \nabla^\top u_\sigma)\,\nabla u)$	yes
biharmonic	$-\Delta^2 u$	no
triharmonic	$\Delta^3 u$	no

- g is some decreasing function such as the Charbonnier diffusivity

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

- u_σ is a Gaussian-smoothed version of u.
- The specific anisotopic nonlinear diffusion model is edge-enhancing diffusion (EED).

M I A

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

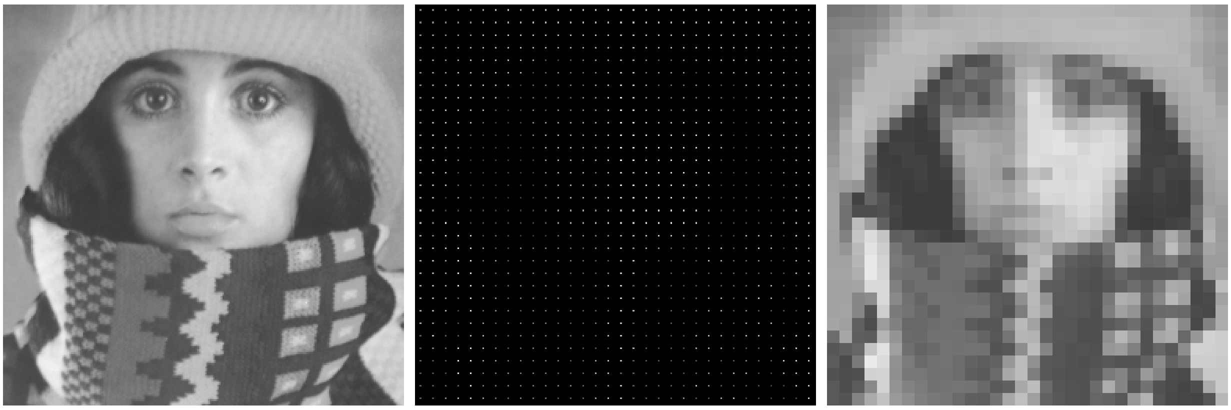
Computational Aspects

- extract $u(\mathbf{x})$ as steady state ($t \to \infty$) of

$$\partial_t u = (1-c(\mathbf{x})) \cdot Lu - c(\mathbf{x}) \cdot (u-f)$$

- finite differences with semi-implicit time discretisation
- leads to a nonsymmetric linear system of equations
- SOR with automatic adaptation of relaxation parameter
- default parameter settings work well:
  $\lambda = 0.1$, $\sigma = 1$, 200–1000 iterations with time step size 1000

Application to Image Zooming



(a) **Left:** Original image. (b) **Middle:** Data points for interpolation. Only 1 out of 64 points is used. This simulates a zooming with a factor 8 in both directions. (c) **Right:** Corresponding downsampled image that is to be interpolated by a factor 8 in both directions.

M I A
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

(a) **Top Left:** Data points for interpolation. (b) **Top Middle:** Interpolation with linear diffusion. (c) **Top Right:** Isotropic nonlinear diffusion. (d) **Bottom Left:** Anisotropic nonlinear diffusion. (e) **Bottom Middle:** Biharmonic smoothing. (f) **Bottom Right:** Triharmonic smoothing.

M I A
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

Quantitative Evaluation

- If v denotes the ground truth and u the interpolation, compute the *average Euclidean error*

$$\mathrm{AEE}\,(u,v) \;:=\; \left(\frac{1}{|\Omega|}\int_{\Omega}\big(u(\mathbf{x})-v(\mathbf{x})\big)^2\,d\mathbf{x}\right)^{1/2}$$

  for all interpolation operators.

- Anisotropic nonlinear interpolation performs best and satisfies a max-min principle:

smoothing operator	AEE	max–min principle
linear diffusion	19.80	yes
isotropic nonlinear diffusion	18.42	yes
anisotropic nonlinear diffusion	15.16	yes
biharmonic smoothing	15.76	no
triharmonic smoothing	16.36	no

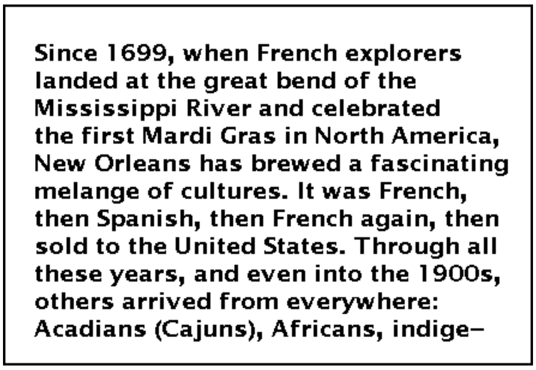
Application to Image Inpainting

- In image inpainting most of the pixel values are correct. Only a few of them are degraded, e.g. by scratches or characters.

- One marks the degraded pixels. These are the regions where the image is interpolated and $c(\mathbf{x}) := 0$. Otherwise $c(\mathbf{x}) := 1$.

- Interpolation with a good PDE-based interpolant such as EED may give convincing results, if texture is not dominating.

- This can also be generalised naturally to vector- and tensor-valued images (cf. Lecture 14).



Top left: Colour image (438×297 pixels) with severe degradations by a text. From M. Bertalmio et al. (2000). **Top right:** Interpolation mask depicting the function $c(\mathbf{x})$. **Bottom:** Result of EED-based interpolation ($\sigma = 0.8$, $\lambda = 0.1$).

M I A
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

Application to Image Compression

Basic Idea

- use good interpolation qualities of EED for (lossy) image compression
- coding: sparsify the image data to a few relevant pixels
- decoding: interpolate the unknown data

Problems

- Which pixels are relevant for compression?
- How can these pixels be coded efficiently?

Application to Image Compression (2)

Coding Step:
The B-Tree Triangular Coding (BTTC) Algorithm by Distasi et al. (1997)

- based on an adaptive triangulation that can be coded in a binary tree
- use image of size $(2^N + 1) \times (2^N + 1)$
- split area along one diagonal into two triangles
- if plane on a each triangle approximates image not well enough:
  subdivide the triangle
- requires 1 bit for coding the location of a relevant node,
  plus 8 bits for its grey value
- further compression by Huffman coding (shorter codes for more frequently used grey values) and quantisation

Application to Image Compression (3)

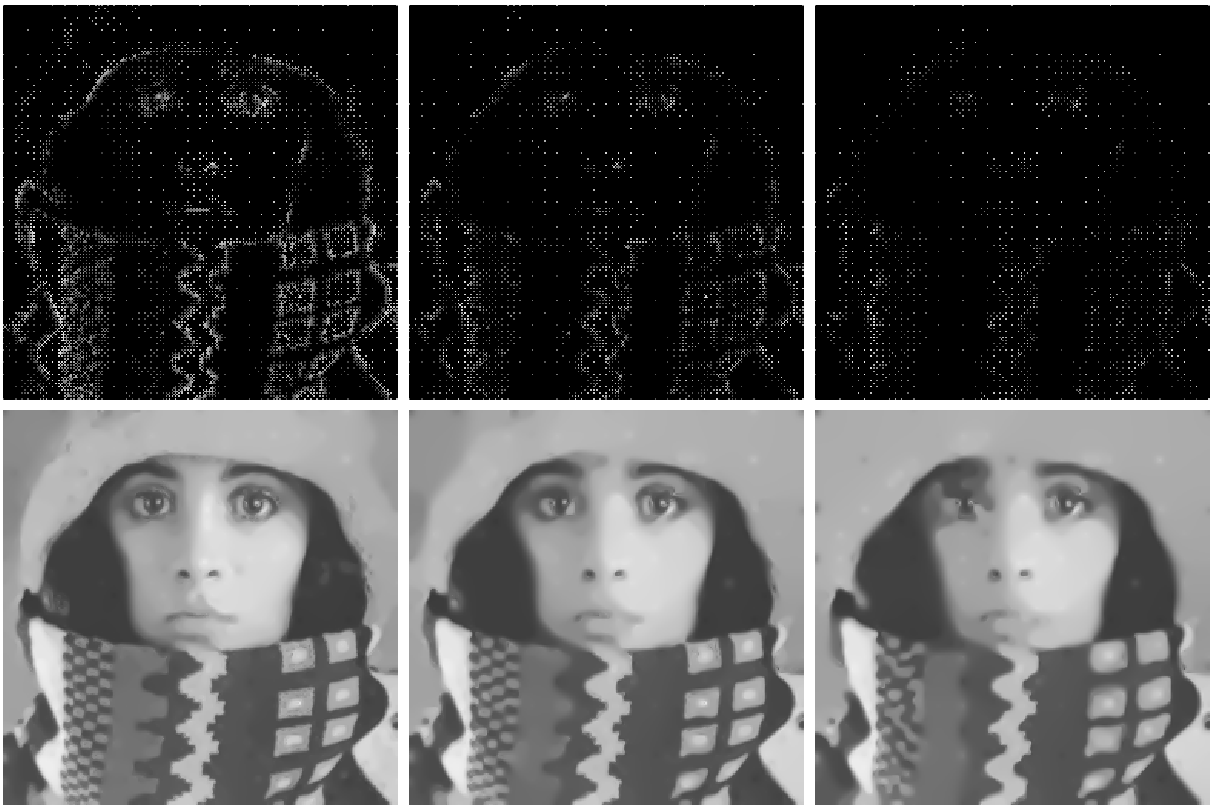
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Decoding Step

- reconstruct the nodes of the adaptive triangulation
- regard the nodes of the triangulation as scattered interpolation points
- use EED for interpolation

Application to Image Compression (4)

M I A
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



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

Application to Image Compression (5)

M I A
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

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

Application to Image Compression (6)

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.

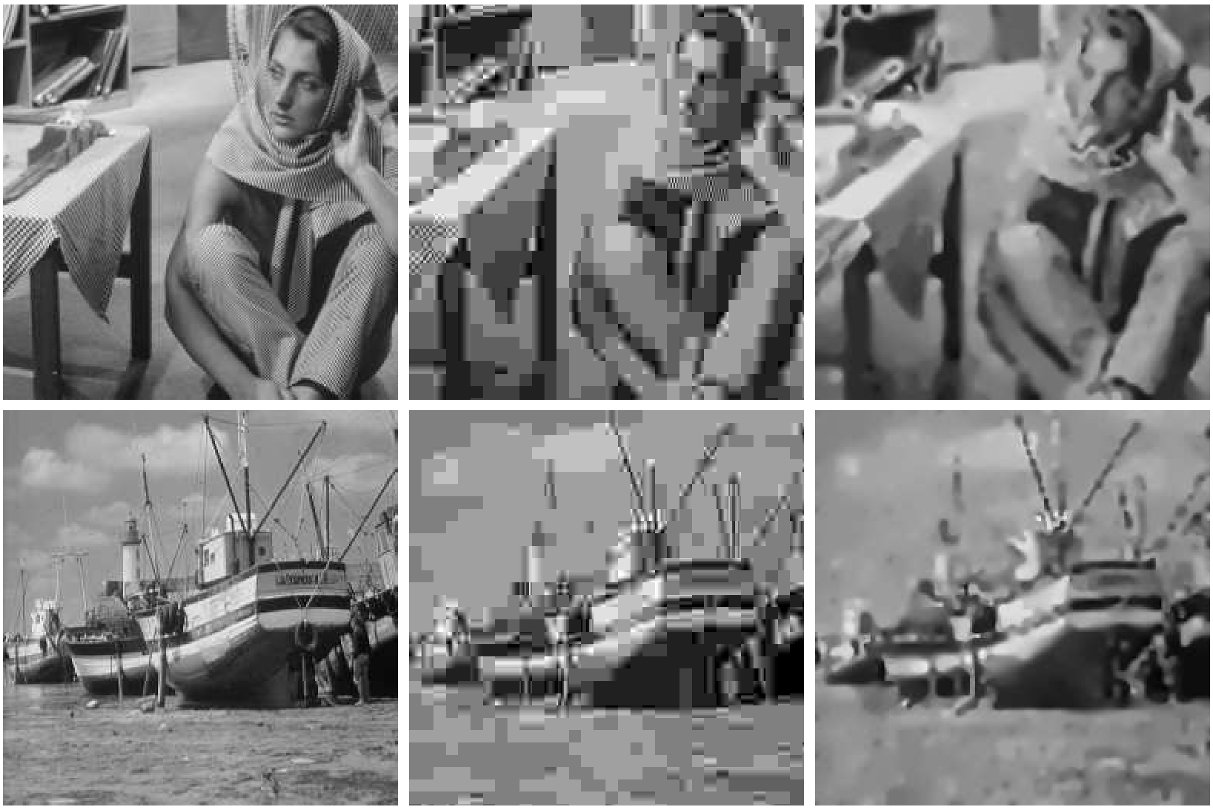
Application to Image Compression (7)

Comparison at high compression rates (40:1) for the test images *trui* and *lena*. **Left column:** Original images. **Middle column:** JPEG. **Right column:** EED-based coding with all amendments (EEDC).

Application to Image Compression (8)

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

Application to Image Compression (9)



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

M I A
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

Quantitative Evaluation

- If v denotes the ground truth and u the interpolation, compute the *average absolute error*

$$\mathrm{AAE}\,(u,v) \;:=\; \frac{1}{|\Omega|}\int_{\Omega} |u(\mathbf{x}) - v(\mathbf{x})|\, d\mathbf{x}$$

  for the different coding methods.

- Results for 0.2 bpp (compression rate 40:1):

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

Summary

- unified interpolation / approximation framework:
  - uses a confidence function
  - generalises variational regularisation and interpolation with splines
- egde-enhancing anisotropic diffusion (EED):
  - performs favourably for interpolation tasks
  - satisfies extremum principle (unlike higher order PDEs)
  - suffers less from singularities than isotropic 2nd order PDEs
  - rotationally invariant (unlike tensor product splines)
- EED is useful for zooming, image inpainting, and image compression.
- example for EED-based image compression:
  - coding via BTTC sparsification
  - decoding as scattered data interpolation
  - can outperform JPEG at high compression rates

References

- J. Weickert, M. Welk: Tensor field interpolation with PDEs. In J. Weickert, H. Hagen (Eds.): *Visualization and Processing of Tensor Fields*. Springer, Berlin, 315–325, 2006. (http://www.mia.uni-saarland.de/publications.shtml)
(introduced the unified PDE model for interpolation and approximation)
- I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, H.-P. Seidel: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, 2008, accepted for publication. (http://www.mia.uni-saarland.de/publications.shtml)
(used this model for image compression)
- R. Distasi, M. Nappi, S. Vitulano: Image compression by B-tree triangular coding. *IEEE Transactions on Communications*, Vol. 45, No. 5, 1095–1100, Sept. 1997.
(introduced the underlying B-tree coding model)
- S. Masnou, J.-M. Morel: Level lines based disocclusion. *Proc. 1998 IEEE International Conference on Image Processing*, Chicago, IL, Oct. 1998, Vol. 3, pp. 259–263.
(early paper on PDE-based inpainting)
- M. Bertalmío, G. Sapiro, V. Caselles, C. Ballester: Image inpainting. *Proc. SIGGRAPH 2000*, New Orleans, LI, July 2000, pp. 417–424.
(shaped the word inpainting for this specific application)
- T. F. Chan, J. Shen: *Image Processing and Analysis – Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, 2005.
(covers also PDEs for inpainting)

Assignment C4 – Classroom Work

Problem 1 (Stability of Diffusion–Reaction Discretisations)

Assume that the diffusion–reaction equation

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(g(|\nabla u|^2)\,\nabla u\right) - \frac{u-f}{\alpha}$$

is discretised with the *modified explicit scheme*

$$\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} = A^k\,\mathbf{u}^k - \frac{1}{\alpha}(\mathbf{u}^{k+1}-\mathbf{f}).$$

Show that its solution $\mathbf{u}^{k+1}$ can be computed as

$$\mathbf{u}^{k+1} = \frac{\alpha\mathbf{v}^{k+1}+\tau\mathbf{f}}{\alpha+\tau}$$

where $\mathbf{v}^{k+1}$ denotes the solution of the explicit diffusion scheme without reaction term:

$$\frac{\mathbf{v}^{k+1}-\mathbf{u}^k}{\tau} = A^k\,\mathbf{u}^k.$$

Show that this implies stability in terms of the discrete maximum-minimum principle

$$\min_j f_j \leq u_i^{k+1} \leq \max_j f_j \qquad \forall\, i, \forall\, k \geq 0$$

if we use the initialisation $\mathbf{u}^0 := \mathbf{f}$ and if the explicit scheme without reaction term satisfies

$$\min_j u_j^k \leq v_i^{k+1} \leq \max_j u_j^k \qquad \forall\, i, \forall\, k \geq 0.$$

Thus, what is the stability condition for τ if $h_1 = h_2 = 1$, $|g(s^2)| \leq 1$ and $\alpha = 30$?

Problem 2 (Stability of Diffusion–Reaction Discretisations)

For the same diffusion–reaction equation as in Problem 1, consider the *fully explicit scheme*

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\tau} = A^k\, \mathbf{u}^k - \frac{1}{\alpha}\,(\mathbf{u}^{k} - \mathbf{f}).$$

What is the stability condition for τ if $h_1 = h_2 = 1$, $|g(s^2)| \leq 1$ and $\alpha = 30$?