2018 Data Compression Conference
Snowbird, UT, USA, March 27–30, 2018

Inpainting-Based Compression of Visual Data

Joachim Weickert
Faculty of Mathematics and Computer Science
Saarland University
Saarbrücken, Germany

long term research funded by

Deutsche Forschungsgemeinschaft DFG

Gottfried Wilhelm Leibniz Prize

ERC Advanced Grant INCOVID

Collaborators

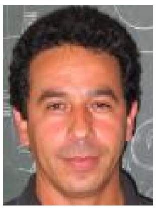
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

joint work with

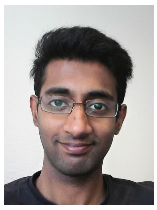

Naoufal Amrani


Sarah Andris


Egil Bae


Zakaria Belhachmi


Matthias Berg


Pinak Bheed


Andrés Bruhn


Dorin Bucur


Irena Galić


Sebastian Hoffmann


Lena Karos

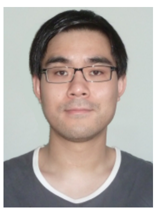

Markus Mainberger


Pascal Peter


Gerlind Plonka


Christian Schmaltz


Joan Serra-Sagrista


Ching Hoo Tang


Martin Welk

Introduction (1)

Introduction

Transform-based Codecs for Lossy Image and Video Compression

- Widely-used lossy codecs such as JPEG, JPEG2000, HEVC are transform-based.
- rely on sparse representation in transform domain (DCT, wavelets)
- sparse approximation in terms of an orthogonal base offers many benefits:
  - clean theory
  - fast algorithms
- quality deteriorates with sparsity

Introduction (2)

Quality Deterioration under JPEG Compression

original

compression rate 10 : 1

compression rate 40 : 1

compression rate 80 : 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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Introduction (3)

Further Improvements through Transform-Based Codecs

- carefully engineered by many experts over many years
- approaches have become highly sophisticated
- increasingly difficult to achieve further fundamental improvements by staying within the universe of transform-based ideas

Things that Transform-Based Codecs Ignore

- more semantical, higher level representation of image structures:
  – need for features (e.g. edges) rather than pixels
  – filling-in effect of our visual system (Werner 1935)
- approximation qualities beyond pixelwise quality measures (MSE, PSNR): respecting e.g. perceptual similarities of stochastic textures

Introduction (4)

An Emerging Alternative: Inpainting-Based Compression

- store only a very small, carefully selected subset of the data
- reconstruct missing data with a suitable inpainting process

What is Inpainting ?

- interpolation strategy for missing / degraded data
- so far mainly used for repairing smaller parts of an image
- two classes relevant for us:
  - PDE-based inpainting (Masnou/Morel 1998, Bertalmio et al. 2000)
  - exemplar-based inpainting (Efros/Leung 1999, Criminisi et al. 2004)

Introduction (5)

PDE-Based Inpainting

original

inpainted

Left: Original image with severe degradations by a text. From Bertalmio et al. (2000). **Right:** Result of inpainting with an anisotropic diffusion process.

- fills in with a partial differential equation (PDE), e.g. a diffusion process
- uncorrupted data serve as fixed boundary conditions
- PDE propagates this information to inpainting domain
- local approach that can be justified from the statistics of natural images

Introduction (6)

Exemplar-Based Inpainting

original

inpainted

Left: Original image. From Criminisi et al. (2004). **Right:** Inpainting result with Criminisi's exemplar-based approach.

- clever copy-and-paste from similar patches within the image
- nonlocal approach that exploits the intrinsic statistical properties of the image
- good for texture-dominated images
- computationally more expensive than PDE-based inpainting

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

How Sparse can the Data be for PDE-based Inpainting ?

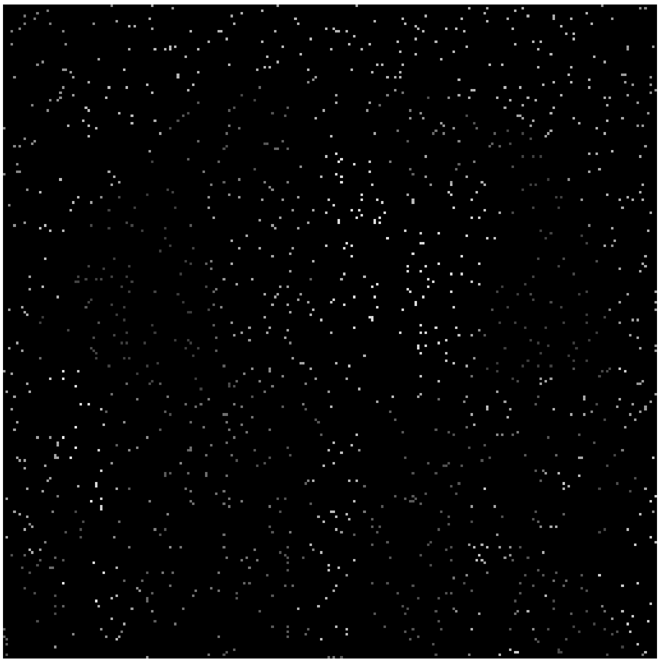


Image where only 2 percent of all pixels are known. Can you recognise what is depicted ?

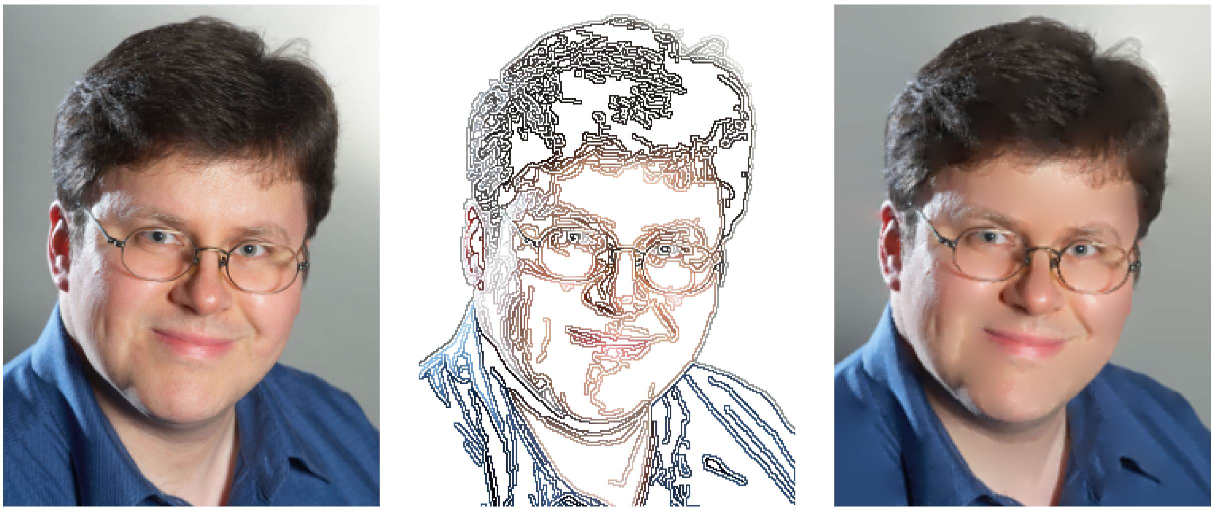
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Is Exemplar-based Inpainting also Useful for Sparse Representations ?

Texture synthesis with the exemplar-based approach of Criminisi et al. (2004), using our fast algorithm with space-filling curves (Dahmen et al. 2017).

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

What Happens if We Select Semantically More Important Data ?



original | data near edges kept | inpainted

Image reconstruction from edges. **Left:** Original image, 237×316 pixels. **Middle:** Edge set. Only the left and right neighbours of each Canny edge are stored. **Right:** Inpainting with the homogeneous diffusion equation.

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Chances for Inpainting-Based Compression

- spatial representation more intuitive than transform-based coding
- respects more properties of human visual system:
  - can use semantically important features such as edges
  - PDE-based inpainting resembles biological filling-in mechanisms
  - exemplar-based inpainting captures statistics of textures, while renouncing pixelwise accuracy
- may go beyond limits of sparse approximations in orthogonal bases
- generic framework for various data types:
  - 1D, 2D, 3D signals
  - still images and videos
  - scalar-, vector-, matrix-valued
  - surface data, graphs
  - dedicated codecs for specific applications

Introduction (11)

Key Challenges for Inpaiting-Based Compression

1. Which data should be kept?
2. What are the most suitable inpainting processes?
3. How can the selected pixels be encoded in an efficient way?
4. Can one design fast algorithms for encoding and decoding?

These Problems are Highly Interrelated:

- Optimality of data depends on inpainting process.
- Suboptimal pixels can pay off if they are cheaper to encode.
- There is a natural tradeoff between high efficiency and high quality.

Introduction (12)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

How Does this Differ from Related Approaches ?

- **Classical Inpainting Methods:**
  - PDE-based inpainting:
    Masnou/Morel 1998, Bertalmio et al. 2000, Chan/Shen 2000, W./Welk 2006, Bornemann/März 2007, Facciolo et al. 2009, Schönlieb 2015
  - exemplar-based inpainting:
    Efros/Leung 1999, Wei/Leroy 2000, Criminisi et al. 2004, Levina/Bickel 2006, Aujol et al. 2010, Arias et al. 2011

  rarely used for very sparse data; do not optimise inpainting data

- **Scattered Data Interpolation Methods, Radial Basis Functions:**

  e.g. Shepard 1968, Duchon 1976, DiBlasi et al. 2010, Achanta et al. 2017

  no data optimisation, usually not applied to compression

- **Image Compression with Subdivision:**

  Dyn et al. 1990, Sullivan et al. 1994, Distasi et al. 1997, Demaret et al. 2006, Kohout 2007, Peyré et al. 2009, Cohen et al. 2012

  usually not inpainting-based, uses e.g. linear splines; more localised

Introduction (13)

- **Inpainting-Based Reconstruction from Image Features:**
  - toppoints in scale-space:
    Johansen et al. 1986, Kanters et al. 2005
  - edges:
    Carlsson 1988, Hummel/Moniot 1989, Desai et al. 1996, Elder 1999, Mainberger et al. 2011, Gautier et al. 2012
  - level lines:
    Solé et al. 2004, Facciolo et al. 2006, Xie et al. 2007
  - derivatives:
    Lillholm et al. 2003, Brinkmann et al. 2015, Peter et al. 2016
  - SIFT points or local binary descriptors:
    Weinzaepfel et al. 2011, DAngelo et al. 2014

often not for compression purposes;
suboptimal for general imagery;
can be useful for specific image classes, e.g. digital elevation maps, depth maps

Introduction (14)

- **PDEs and Inpainting Ideas within Classical Codecs**
  - many papers on PDE-based pre- or postprocessing
  - PDE-improved wavelet thresholding:
    Chan/Zhou 2000
  - inpainting within JPEG, H.264/AVC etc.:
    Liu et al. 2007, Xiong et al. 2007, Ballé et al. 2011, Ndjiki-Nya et al. 2012, Racape et al. 2013

Does this exploit the full potential of inpainting?

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Goals of This Talk

- survey achievements and challenges in inpainting-based compression
- philosophy:
  - focus mainly on PDE-based inpainting, use exemplar-based inpainting when necessary
  - aim at designing dedicated inpainting-based codecs from scratch, rather than embedding inpainting into existing frameworks
  - show that they can reproduce important features of transform-based codecs

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

Optimising the Data

Our Baseline Codec

- **Encoding**:
  - store (greyscale) image $f : \Omega \to \mathbb{R}$ only in some small subset $K \subset \Omega$.
- **Decoding**:
  - In this subset K *(inpainting mask)*, the reconstruction u is known:

$$u(\boldsymbol{x}) \;=\; f(\boldsymbol{x}).$$

  - In the *inpainting domain* $\Omega \setminus K$ where the data are unknown:
    Compute the steady state $(t \to \infty)$ of the homogeneous diffusion equation

$$\partial_t u \;=\; \Delta u$$

    with the known data as fixed boundary conditions
  - In short: Inpaint with the Laplace equation $\Delta u \;=\; 0$.

Optimising the Data (2)

Continous Spatial Data Optimisation

- Belhachmi et al. (2009): analytic theory based on shape optimisation
- result: choose density of the data points as an increasing function of $|\Delta f|$
- real-time encoding: just compute Laplacian and apply dithering

Felix Klein

Laplacian magnitude

10% mask by dithering

inpainting

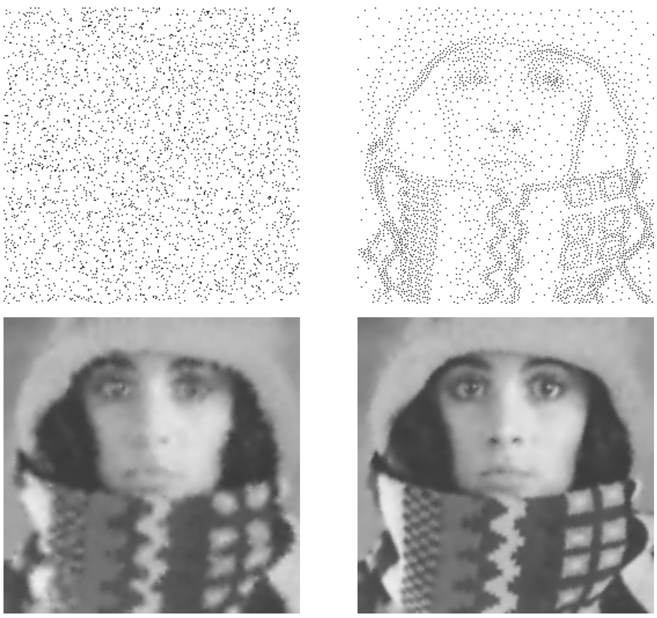
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Optimising the Data (3)

Analytic Approach Clearly Outperforms Random Selection

	randomly selected (5% pixels)	analytic approach [1] (5% pixels)

mask / reconstruction



MSE: 189.90 | MSE: 49.47

original

[1] with electrostatic dithering, Schmaltz et al. (2010)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Discrete Spatial Data Optimisation

Why Discrete?

- Althought the analytic approach is optimal in theory, dithering may introduce substantial errors.
- Does a genuinely discrete data optimisation perform better?

What Happens if we Discretise our Inpainting Problem ?

(Mainberger et al. 2011)

- finite differences lead to sparse linear system of equations: one unknown per unknown pixel
- has a unique solution, if at least one pixel is specified

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Good and Bad News about Discrete Data Selection

Good News: Problem is finite, thus a global optimum exists.

Bad News: Selecting the best 5% pixels of a 256×256 image offers $\binom{65536}{3277} \approx 1.72 \cdot 10^{5648}$ possibilities.

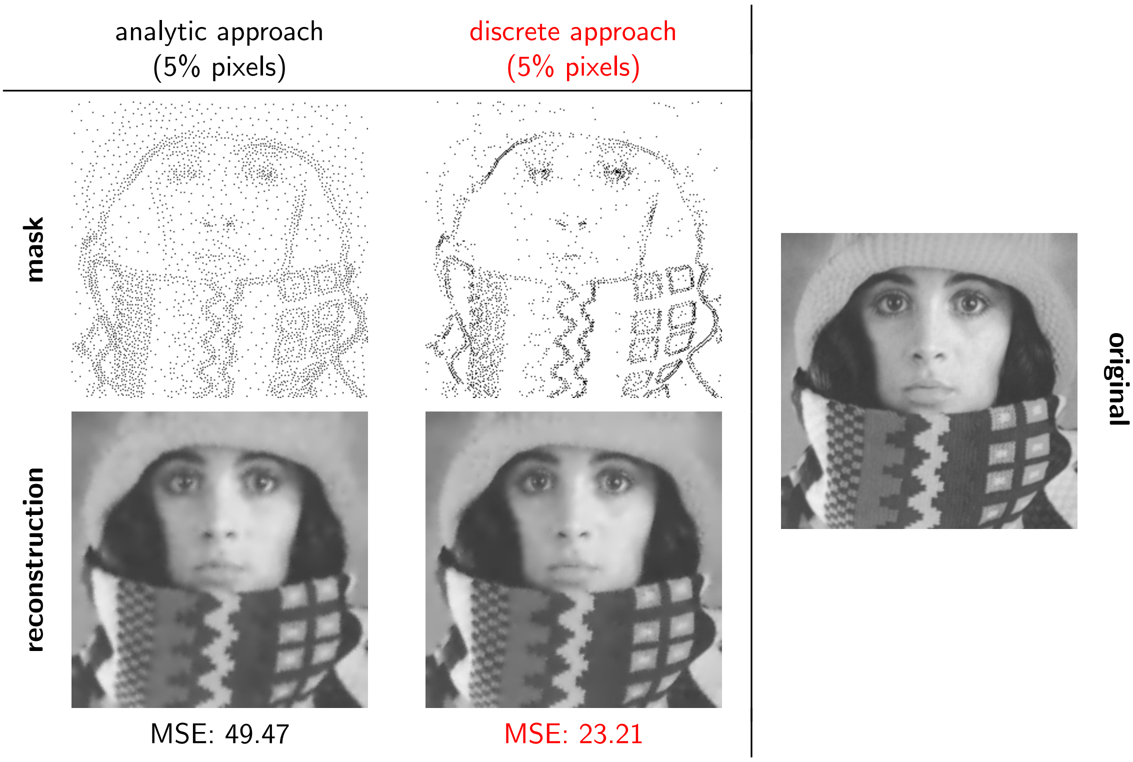
Well-Performing Minimisation Algorithm

(Mainberger et al. 2012)

- **Probabilistic Sparsification:**
  - start with full mask
  - gradually discard pixels that give only small errors when they are removed
- **Postprocessing by Nonlocal Pixel Exchange:**
  - to avoid getting trapped in local minima
  - exchange randomly selected mask pixels with non-mask pixels with large errors, if this improves the total error
- **Alternative Optimisation Strategies:**
  - e.g. bilevel ideas (Hoeltgen et al. 2013, Ochs et al. 2014, Bonnetini et al. 2017)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Substantial Improvements over Analytic Approach:



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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Discrete Tonal Data Optimisation

(Mainberger et al. 2012)

Motivation

- So far we have optimised the *location* of the pixels:
- Can we achieve further improvements by optimising their *grey values* ?
- We call this *tonal optimisation*.
- Such a mechanism would not help for transform-based codecs:
  Orthogonal basis approximations cannot be improved by tuning the coefficients.

How Does This Work ?

- Let some image $\boldsymbol{f}$ and some optimised fixed pixel mask be given.
- Perturb data $\boldsymbol{f}$ within the pixel mask by some vector $\boldsymbol{\alpha}$.
- Minimising the MSE w.r.t. $\boldsymbol{\alpha}$ is a least squares problem with unique solution.
- resulting linear system is symmetric and dense (as many unknowns as mask pixels)
- can be solved with standard algorithms such as Cholesky, QR, or CG

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Tonal Optimisation Leads to Significant Quality Gains:

randomly selected (5 % pixels)	analytic approach (5 % pixels)	discrete approach (5 % pixels)
MSE: 189.90	MSE: 49.47	MSE: 23.21

before / after

MSE: 106.17	MSE: 31.62	MSE: 17.17

original

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Optimising the Inpainting Operator

- so far: inpainting with the homogeneous diffusion operator Δu;
  requires many data to represent edges

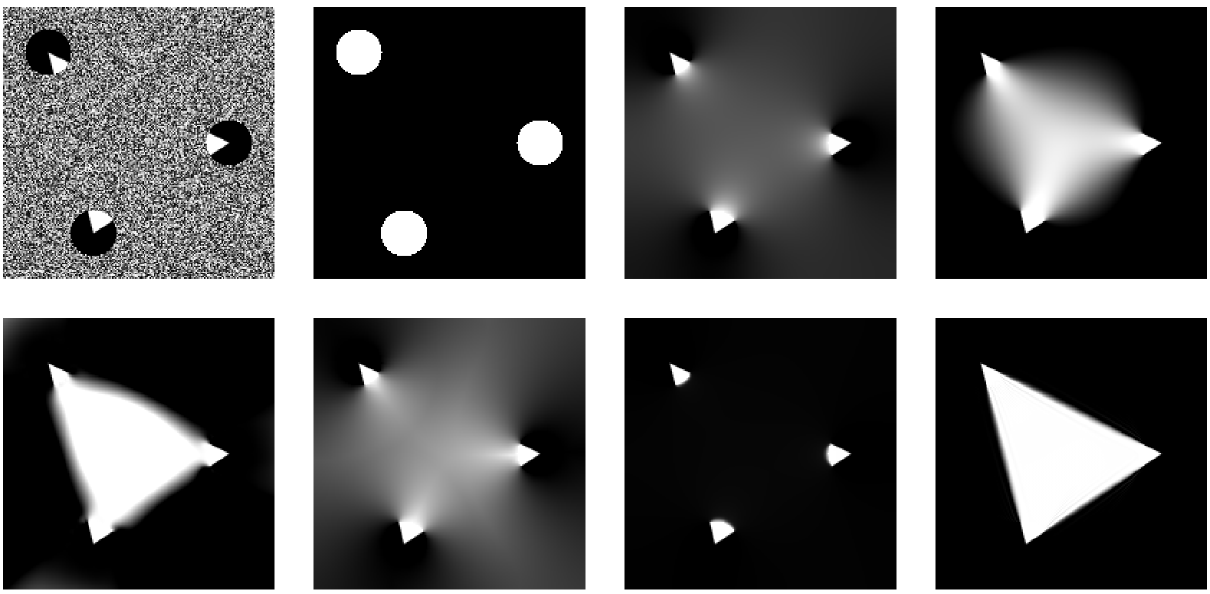
- Galić et al. 2005, Schmaltz et al. 2009, Chen et al. 2014:
  - experimental evaluations of various differential operators
  - motivated from spline theory or inpainting literature

interpolation strategy	authors	differential oper.	max–min
homogeneous diffusion	Iijima 1959	Δu	yes
biharmonic interpol.	Duchon 1976	$-\Delta^2 u$	no
triharmonic interpol.		$\Delta^3 u$	no
AMLE	Caselles et al. 1998	$u_{\eta\eta} \quad (\eta \parallel \boldsymbol{\nabla} u)$	yes
TV inpainting	Rudin et al. 1992	$\operatorname{div}\left(\frac{\boldsymbol{\nabla} u}{\lvert\boldsymbol{\nabla} u\rvert}\right)$	yes
anisotropic diff. (EED)	W. 1996	$\operatorname{div}\left(\boldsymbol{D}(\boldsymbol{\nabla} u_\sigma)\, \boldsymbol{\nabla} u\right)$	yes

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Comparison of Different Inpainting Operators

input image | inpainting mask | homogeneous diff. | biharmonic



triharmonic | AMLE | TV | anisotr. diff. (EED)

- TV performs always bad for sparse data (prefers small edge lengths).
- Edge-enhancing anisotropic diffusion (EED) is consistently good.

Optimising the Inpainting Operator (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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Best performing inpainting operator in this and other experiments:

Edge-Enhancing Anisotropic Diffusion (EED)

- uses nonlinear inpainting operator

$$\mathrm{div}\,(\boldsymbol{D}(\boldsymbol{\nabla} u_\sigma)\,\boldsymbol{\nabla} u)$$

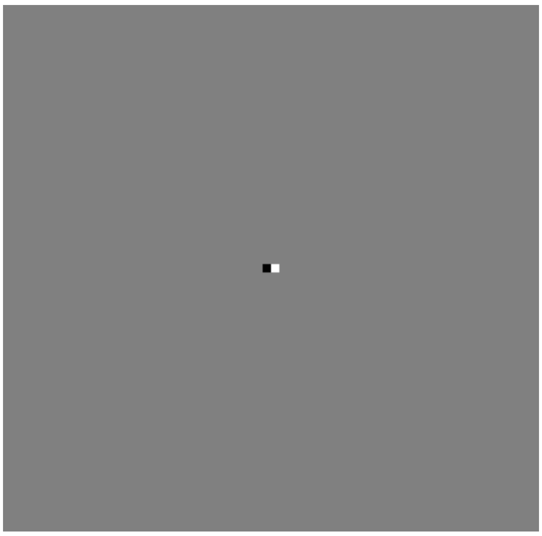
- originally for denoising (W. 1996), later for inpainting (W./Welk 2006)
- u_σ is a Gaussian-smoothed version of u
- diffusion tensor $\boldsymbol{D}(\boldsymbol{\nabla} u_\sigma)$:
  - symmetric 2×2 matrix
  - adapts itself to (semi-) local image structure
  - prefers inpainting along edges over inpainting across them
- requires to solve sparse nonlinear systems of equations

Optimising the Inpainting Operator (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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Why Does Anisotropic Diffusion (EED) Work so Well ?

Two Simple but Instructive Experiments

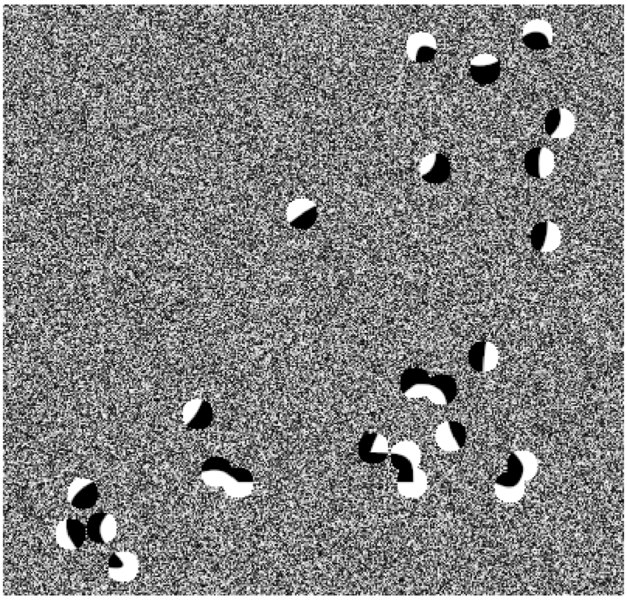


dipole

occluded disk

grey initialisation of inpainting domain

EED has Interesting Shape Reconstruction Qualities



Binary image, where data are specified within 24 small disks. The unknown pixels are initialised with uniform noise. Reconstruction is performed with anisotropic diffusion (EED). Author: W. (2012).

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

What is so Special about EED ?

- **anisotropy due to the diffusion tensor:**
  - makes it well-suited for reconstructing edges
  - no need for high mask pixel density at edges
- **nonlocality due to Gaussian convolution:**
  - allows to propagate edge structures
  - creates a curvature-reducing effect (cf. Merriman et al. 1994)
- **very natural:**
  - EED-like operators reflect statistics of natural images (Peter et al. 2015)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

Optimising the Encoding

Lessons Learnt

- Homogeneous diffusion requires careful data optimisation.
- Anisotropic diffusion (EED) can be a better alternative:
  - It needs fewer data to reconstruct edges (and texture).
  - Thus, the data distribution can be more evenly.
  - This means that the data location is a bit less critical.

Key Aspect Ignored so Far

- encoding costs: storing a freely optimised mask can be expensive

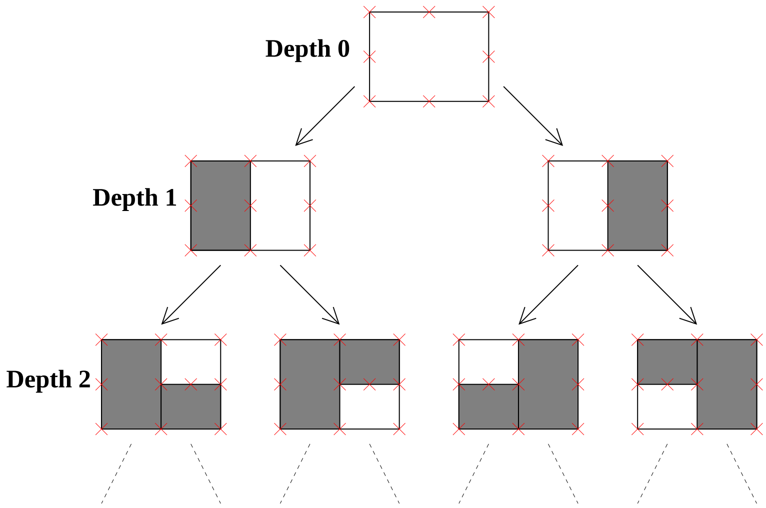
Balancing all Requirements

- use EED and find a slightly suboptimal mask that is much cheaper to encode

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Finding a Useful Mask that is Inexpensive to Encode

- rectangular subdivision where approximation quality of EED is insufficient (cf. Dyn et al. 1990, Sullivan et al. 1994, Distasi et al. 1997)
- encoding in a binary tree costs about 1 bit per pixel location



Rectangular subdivision. Each subdivision step splits the white area and inserts one point.

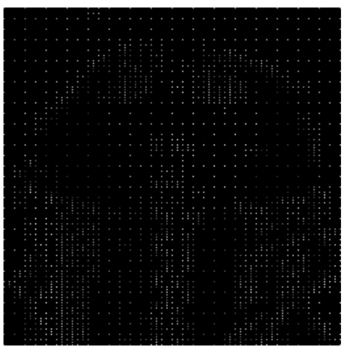
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Designing a Full Codec

- benefit from established concepts on the encoding side:
  – coarser quantisation of grey values
  – remove redundancy from entire bitstring by entropy coding (e.g. PAQ)
- decoding: inpaint with EED

Example at a Compression Rate of 57 : 1

original image



data from subdivision

EED-based decoding

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Comparison to JPEG and JPEG 2000 at 57 : 1

original

JPEG, MSE $= 111.01$

JPEG 2000, MSE $= 70.79$

EED, MSE $= 42.77$

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

What Happens for Extremely High Compression Rates ?

JPEG 2000, 162 : 1

JPEG 2000, 202 : 1

JPEG 2000, 261 : 1

EED, 163 : 1

EED, 211 : 1

EED, 264 : 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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Fast Algorithms and Real-Time Aspects

- distinguish between real-time encoding and real-time decoding:
  - e.g. video decoding must be real-time, video encoding not necessarily
- encoding requires fast data optimisation
  - simple strategies real-time capable, e.g. analytic approach with dithering
- decoding with homogeneneous diffusion:
  - requires to solve sparse linear systems of equations
  - adapted full multigrid schemes
  - real-time performance on CPUs (Mainberger et al. 2011)
- decoding with edge-enhancing anisotropic diffusion (EED):
  - requires to solve sparse nonlinear systems of equations
  - developed novel, so-called fast explicit diffusion schemes (W. et al. 2016)
  - real-time performance on GPUs (Peter et al. 2015)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

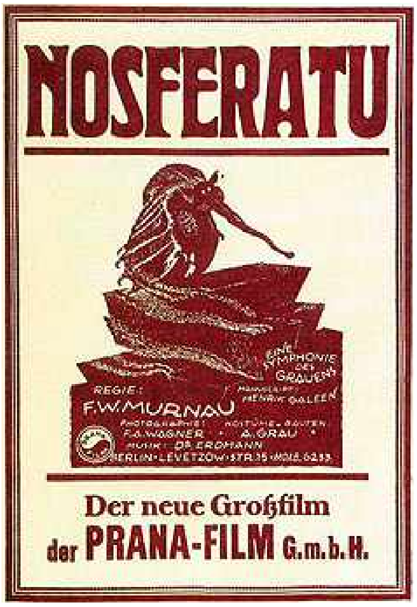
Real-Time Demo: Analytic Approach with Web Cam (640 × 480 Pixels)

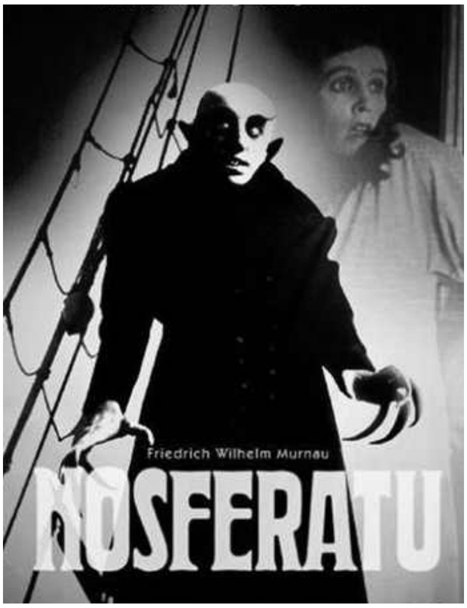
- data selection: Floyd-Steinberg dithering of Laplacian magnitude
- inpainting: homogeneous diffusion with full multigrid algorithm
- computations done on the Laptop CPU (Intel i5–6300HQ, 2.30 GHz, 4 cores)
- solves three systems of 307,200 linear equations with ca. 13 fps

Start (320×240) Start (640×480)

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Real-Time Decoding with EED-Based Inpainting (Peter et al. 2015)





Nosferatu – Eine Symphonie des Grauens is a silent movie directed by Friedrich Wilhelm Murnau in 1922. We can play it in full length (94 minutes) with our EED-based decoder. Inpainting 25 frames per second with 640×480 pixels on a PC requires highly optimised parallel algorithms for GPUs. We use our fast explicit diffusion schemes (W. et al. 2016).

Outline

- Optimising the Data
- Optimising the Inpainting Operator
- Optimising the Encoding
- Fast Algorithms and Real-Time Aspects
- Connections, Extensions, and Applications
- Summary and Outlook

Connections, Extensions, and Applications

Connections to Sparsity

(Hoffmann et al. 2015)

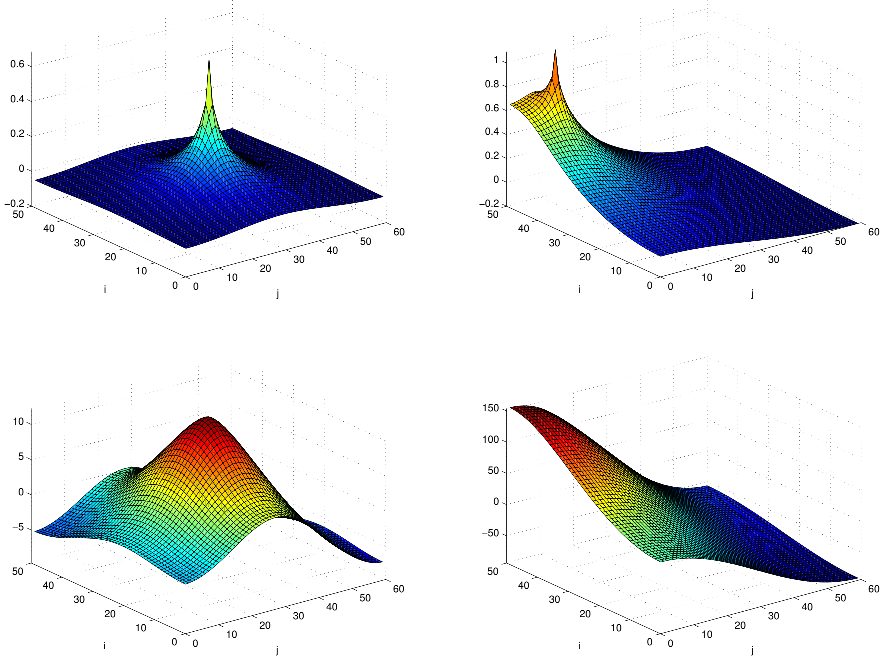
- consider Green's functions of a discrete linear differential operator $\boldsymbol{D}$
- discrete Green's function $\boldsymbol{g}_{k,\ell}$ in a pixel (k,ℓ) solves

$$(\boldsymbol{D}\,\boldsymbol{g}_{k,\ell})_{i,j} \;=\; \delta_{(k,\ell),(i,j)} \qquad \text{for all } (i,j)$$

  where $\delta_{(k,\ell),(i,j)}$ is the Kronecker function.
- inpainting solution via superposition of Green's functions in mask points
- Green's functions are atoms in a dictionary
- allows also fast algorithms for sparse inpainting problems

Discrete Green's Functions for Operators with Reflecting Boundary Conditions

Laplace Operator in $(25, 30)$ | Laplace Operator in $(45, 10)$



Biharmonic Operator in $(25, 30)$ | Biharmonic Operator in $(45, 10)$

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Adaptation to Progressive Mode Coding

(Peter et al. 2015)

JPEG 2000, 33 % | JPEG 2000, 67 %, | JPEG 2000, 100 %

EED, 33 % | EED, 67 % | EED, 100 %

Illustration of progressive mode capacities at a compression rate of $80 : 1$. The percentage refers to the size of the compressed image.

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Edge-based Codecs for Piecewise Almost Constant Images

(Mainberger et al. 2011)

original

data kept

reconstruction

Left: Original image. **Middle:** Stored data with tonal optimisation. The data are subsampled along the contour, requantised and entropy coded. **Right:** Reconstruction with homogeneous diffusion in each colour channel. Compression rate: 200:1.

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Comparison to JPEG and JPEG 2000 at 200 : 1

original image

JPEG, MSE=441.20

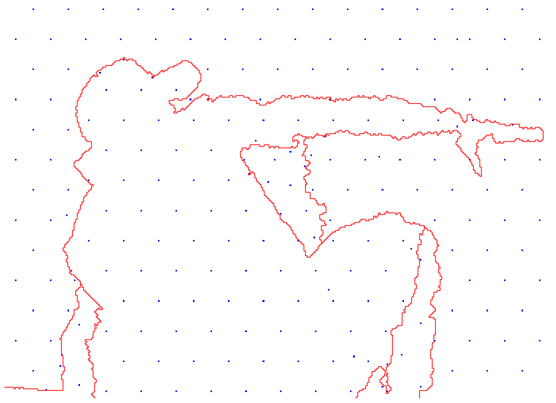
JPEG 2000, MSE=164.53

diffusion, MSE=25.97

Segmentation-based Codecs for Piecewise Smooth Images

(Hoffmann et al. 2013)

depth map

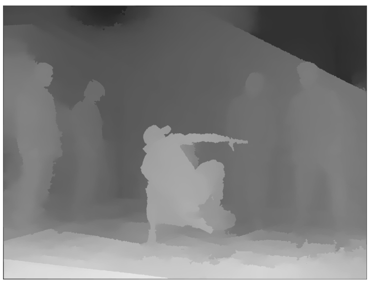


stored data

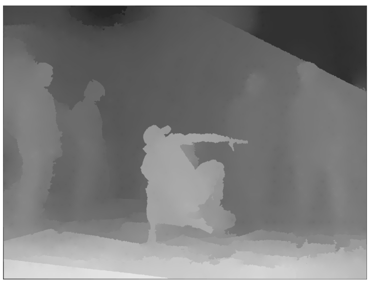
Left: Zoom into a depth map image. It is a good example of a piecewise smooth image. **Right:** Segment boundaries and data used for homogeneous diffusion inpainting. We store the segmentation boundaries, the grey values on a hexagonal grid, and the locations and grey values of a few extra pixels.

Comparison at 177 : 1 (0.045 bpp)

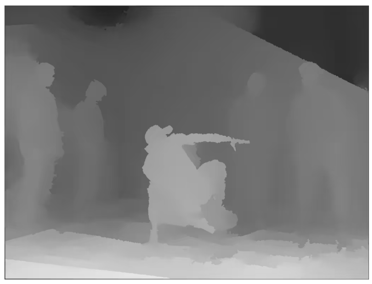
original depth map



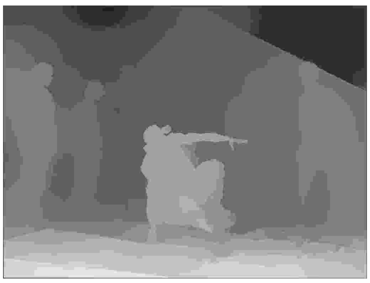
segmentation-based hom. diff.



HEVC



JPEG 2000



JPEG

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Exemplar-based Inpainting for Highly Textured Images

Hybrid Codec with EED- and Exemplar-Based Inpainting (Peter/W. 2015)

original EED hybrid

MSE=118.70 MSE=77.35

We use EED-optimised data and decide locally if EED-based inpainting or Facciolo's exemplar-based inpainting for sparse data is better. **Top:** Images in full size. **Bottom:** Zoom. Compression rate: 18:1.

Data Optimisation for Exemplar-Based Inpainting

(Karos et al. 2018)

original image optimised data, 5 % inpainted

For highly textured images, it can be beneficial to use nonlocal, patch-based inpainting methods for sparse data (Facciolo et al. 2009). Currently we are developing novel data optimisation strategies that improve the quality substantially.

Extension to Colour Images

(Peter et al. 2014)

original

JPEG 2000, MSE=151.31

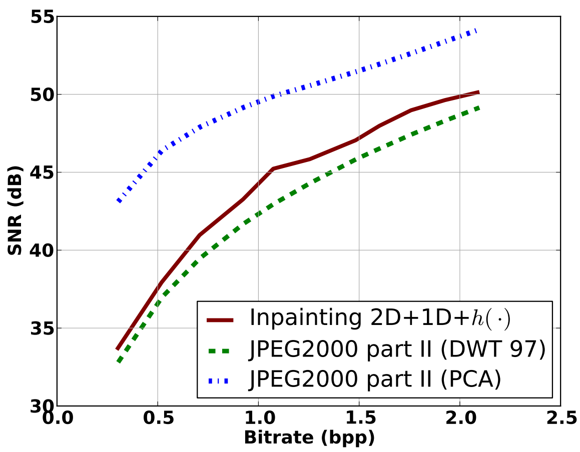
EED, MSE=77.53

Compression of a 256×256 subimage of *peppers* with compression rate 110:1. **Left:** Original image. **Middle:** JPEG 2000. **Right:** EED in Luma Preference Mode. This mode works in the YCbCr space. It compresses the luma channel Y with higher quality. The resulting diffusion tensor steers the inpainting in the chroma channels Cb and Cr.

Extension to Hyperspectral Data

(Amrani et al. 2017)

- homogeneous diffusion inpainting in space
- biharmonic inpainting in spectral dimension
- additional linear prediction model
- qualitatively between JPEG 2000 Part II with DWT 9/7 and JPEG 2000 with PCA



Extension to Three-Dimensional Data Sets

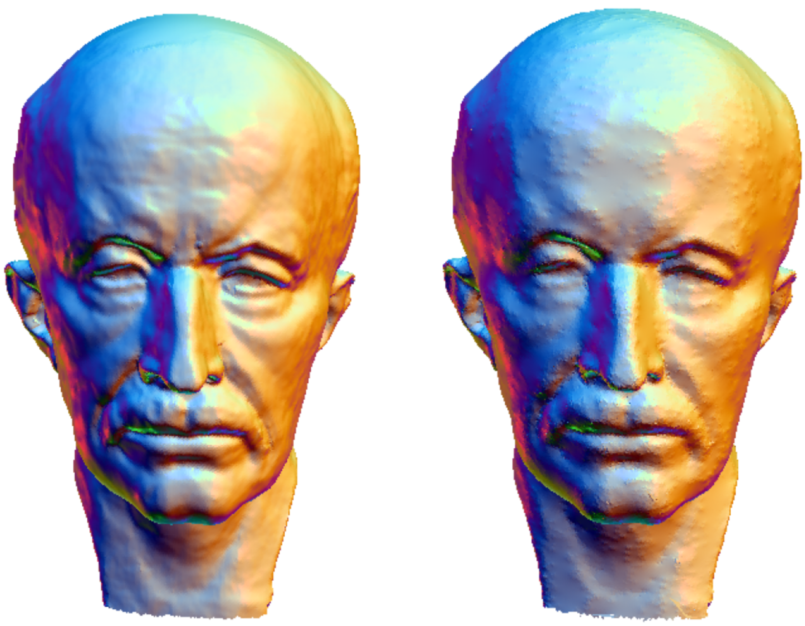
(Peter 2013)

original | DICOM, MSE=65.02 | EED, MSE=38.66

Compression of a 3-D data set with a compression rate of 207:1. **Left:** Original slice from a 3-D CT data set of a trabecular bone. **Middle:** Result after compression in the DICOM standard, based on JPEG 2000. **Right:** Compression with 3-D EED with cuboidal subdivision.

Extension to Surface Data

(Bae/W. 2010)

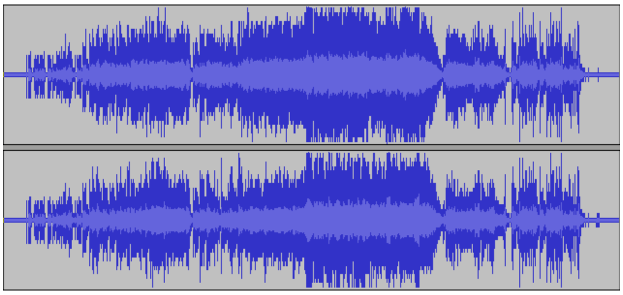


Left: Original data set. **Right:** Reconstruction with optimised 10 % of the points and geometric linear diffusion.

Adaptation to Audio Compression

(Peter et al. 2018)

- sparse representation of audio signal in sample domain
- homogeneous diffusion inpainting becomes linear interpolation
- data sparsification approach, coding with LPAQ
- competitive to mp3, AAC, and Vorbis in terms of SNR, inferior w.r.t. perceptual quality measures such as PEAQ



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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

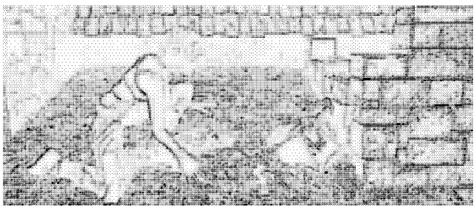
Extension to Video Compression

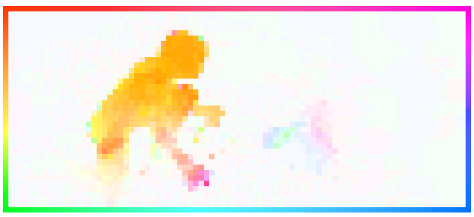
(Andris et al., PCS 2016 Best Poster Award)

- inpainting-based coding of selected images (intraframes)
- prediction of images inbetween (interframes) with variational optic flow
- additional coding of the residual errors

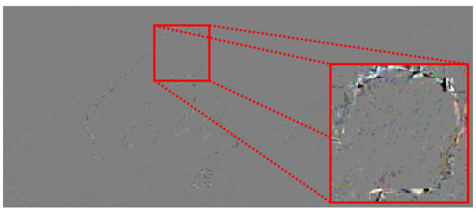
original frame

inpainting mask (intraframe)





subsampled motion field (interframe)



interframe residual

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Example: MPI Sintel Data Set

stored data prediction

residue final reconstruction

MI 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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Steganographic Application: Censoring

(Mainberger et al. 2012)

original

censored

restored

Left: Original image. **Middle:** Censored version. The censored part is EED-encoded and hidden in the least significant bits of the non-censored part. **Right:** Reconstruction.

- Try it with your own image: `stego.mia.uni-saarland.de`

MI 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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Outline

- Continuous Spatial Data Optimisation
- Discrete Spatial Data Optimisation
- Discrete Tonal Data Optimisation
- Better PDEs
- Optimising the Encoding
- Connections, Extensions, and Applications
- Summary and Outlook

Summary

- inpainting offers novel and promising ways for compressing visual data
- store subset of pixels and inpaint inbetween
- can achieve results of competitive quality,
  if data, inpainting process, and coding are optimised
- flexible and widely applicable

Outlook

- theoretical underpinning, quality guarantees
- learning the optimal data and inpainting operators
- treating all optimisation problems simultaneously
- inpainting-based codecs for highly textured images
- highly efficient numerical algorithms for parallel architectures

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 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65

Thank you!

www.mia.uni-saarland.de/Research/IP_Compress.shtml