Optimal Interpolation Data for Image Reconstructions

PhD. Defense Talk

Laurent Arthur Hoeltgen

March 30th, 2015

What Is Inpainting?

Recover missing data in given image through interpolation.

Image with missing data

Reconstruction with Laplace interpolation

Motivation (2)

- Standard setting:
  - impossible to choose missing data
  - requires optimal interpolation methods
- We take a different approach:
  - fix interpolation method
  - select sparse and optimised data

Is it possible to get good results with only 5% of data?

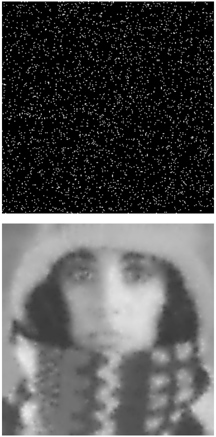
- Why do this?
  - allows image compression

Motivation (3)

How Important Is a Good Optimisation?

- Position of interpolation data matters when sparse (say 5%): (Mainberger et al. 2012)

Original image



Random positions

Optimal positions

- Pixel values at given positions equally important (Mainberger et al. 2012)

  Reconstructions from same **random** mask with 5% density:

Original image

Without grey value optimisation

With grey value optimisation

How to find all this optimal data?

Related work

Previous findings on PDE-Based inpainting and compression:

- Contour-based reconstructions (Carlsson 1988)
- Wavelet decompositions with TV (Chan et al. 2001)
- Toppoints in scale space (Kanter et al. 2005)
- Spline based representations (Orzan et al. 2008)
- Subdivision schemes (Galić et al. 2008, Schmaltz et al. 2009)
- Variational approach (Belhachmi et al. 2009)
- Stochastic optimisation (Mainberger et al. 2012)
- Bilevel optimisation (Chen et al. 2014)

Outline

- PDE-Based Image Inpainting
- Finding Optimal Data Locations
- Finding Optimal Data Values
- An Image Compression Codec

PDE-Based Image Inpainting

Which interpolation operator should be used?

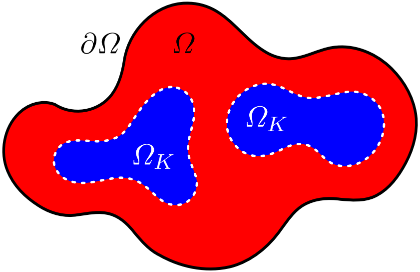
Requirements:

1. simple to analyse
2. applicable for any domain and codomain
3. able to handle arbitrarily scattered data
4. fast to carry out

Laplace interpolation fulfils all these requirements.

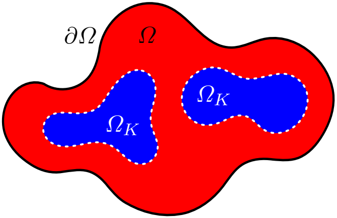
Laplace Interpolation for Image Inpainting

Consider the Laplace equation with mixed boundary conditions.



$$\begin{cases} -\Delta u = 0, & \text{on } \Omega \\ u = g, & \text{on } \Omega_K \\ \partial_n u = 0, & \text{on } \partial\Omega \end{cases}$$

- Ω_K: represents known data
- $\Omega \setminus \Omega_K$: region to be inpainted (i.e. unknown data)
- Image reconstructions are solutions u.



$$
\begin{cases} -\Delta u = 0, & \text{on } \Omega \\ u = g, & \text{on } \Omega_K \\ \partial_n u = 0, & \text{on } \partial\Omega \end{cases}
$$

Mixed boundary value problem can be rewritten as

$$
\begin{aligned} c\,(u-g) + (1-c)\,(-\Delta)u &= 0, \quad \text{on } \Omega \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \end{aligned}
$$

with $c \equiv 1$ on Ω_K and $c \equiv 0$ on $\Omega \setminus \Omega_K$.

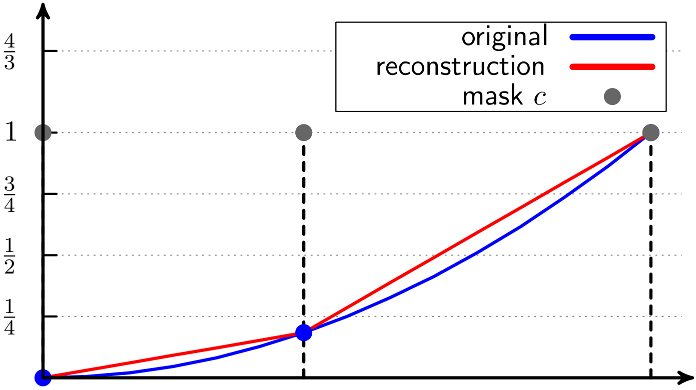
Optimising **binary** c known as **free knot problem** for 1D signals.

Previous equation makes also sense if c maps to whole $\mathbb{R}$.

- can be seen as a regularisation

Benefits of a Non-Binary Mask c

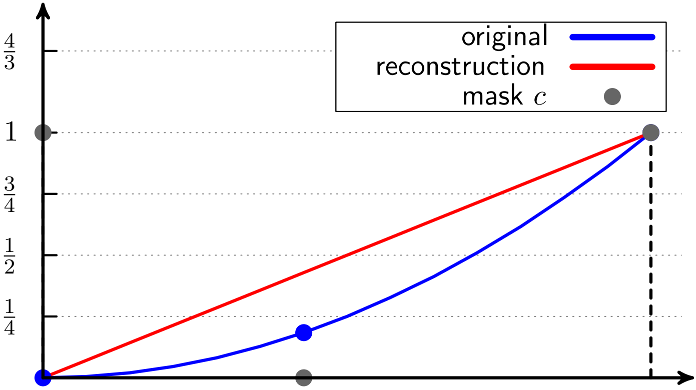
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



Reconstruction by piecewise linear inerpolation in 1D.

Benefits of a Non-Binary Mask c

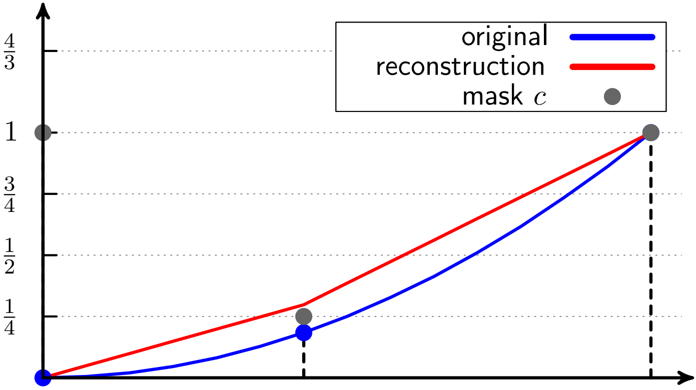
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.45

Benefits of a Non-Binary Mask c

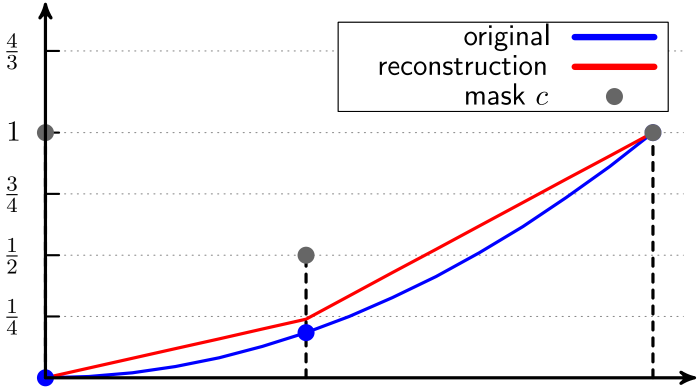
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.43

Benefits of a Non-Binary Mask c

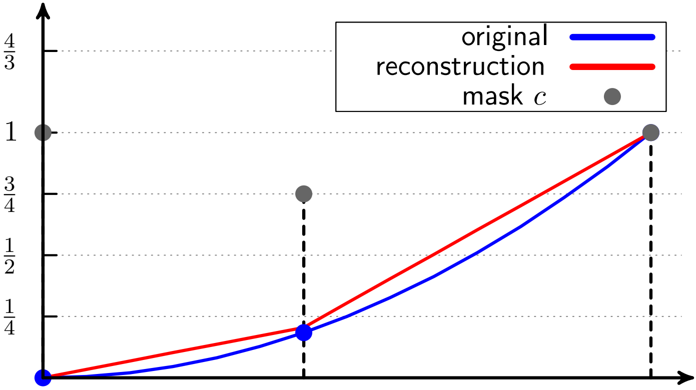
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.41

Benefits of a Non-Binary Mask c

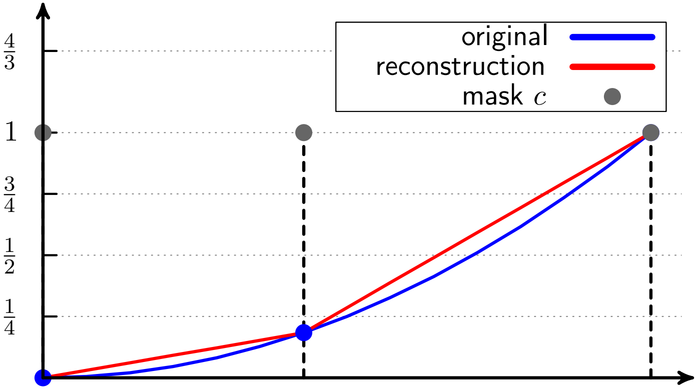
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.39

Benefits of a Non-Binary Mask c

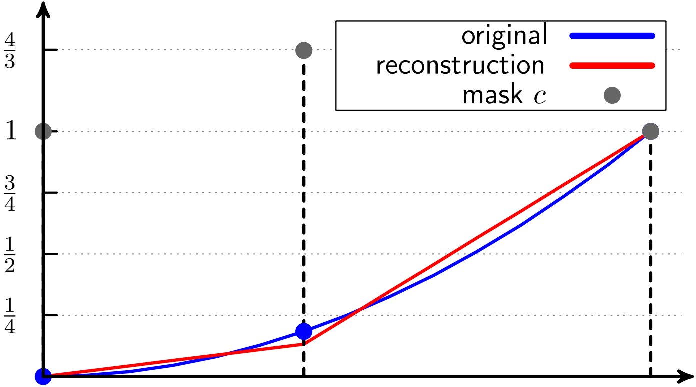
- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.37

Benefits of a Non-Binary Mask c

- Finding binary masks is a non-convex, combinatorial task.
- Non-binary masks allow better fits to the data.



L_2 Error: 0.34

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

New Findings

Standard finite difference schemes in 2D for

$$\begin{aligned} c\ (u-g)+(1-c)\,(-\Delta)u &= 0, \quad \text{on } \Omega \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \end{aligned}$$

yield a linear system of equations.

The system matrix has the following properties:

1. All eigenvalues are real if c is bounded by 1.
2. Matrix is invertible if c maps to $\left[0, \frac{8}{7}\right]$.
3. Solutions obey max-min principle if c maps to $[0, 1]$.

Generalisation of Mainberger et al. (2011) to the non-binary case.

A Novel Optimal Control Model (EMMCVPR 2013)

Optimal **non-binary** masks c for recovering g obtained from

$$\underset{u,\, c}{\arg\min}\{\int_\Omega \frac{1}{2}\,(u-g)^2 + \lambda|c| + \frac{\varepsilon}{2}|c|^2\,\mathrm{d}x\}$$

$$\text{subject to:} \begin{cases} c\,(u-g) + (1-c)(-\Delta)u = 0, & \text{on } \Omega \\ \partial_n u = 0, & \text{on } \partial\Omega \end{cases}$$

- PDE enforces that we only get suitable solutions.
  - one solution $u(c)$ for each valid choice of c
- Cost function optimises reconstructions u and masks c.
  - $\frac{1}{2}(u-g)^2$ favours accurate reconstructions.
  - $\lambda|c|$ prefers sparse data sets.

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

Interpretation

$$\underset{u,\, c}{\arg\min}\{\int_{\Omega} \frac{1}{2}\,(u-g)^2 + \boldsymbol{\lambda}|c| + \frac{\varepsilon}{2}|c|^2 \,\mathrm{d}x\}$$

$$\text{subject to: } \begin{cases} c\,(u-g) + (1-c)(-\Delta)u = 0, & \text{on } \Omega \\ \partial_n u = 0, & \text{on } \partial\Omega \end{cases}$$

- Energy reflects trade-off between accuracy and sparsity.
  - Objectives cannot be fulfilled simultaneously.
- $\boldsymbol{\lambda}$ steers sparsity of the interpolation data.
  - Small, positive $\boldsymbol{\lambda}$: many data points, good reconstruction
  - Large, positive $\boldsymbol{\lambda}$: few data points, bad reconstruction

Properties

$$\operatorname*{arg\,min}_{u,\,c}\left\{\int_{\Omega} \frac{1}{2}\,(u-g)^2 + \lambda|c| + \frac{\varepsilon}{2}|c|^2 \,\mathrm{d}x\right\}$$

$$\text{subject to: } \begin{cases} c\,(u-g) + (1-c)(-\Delta)u = 0, & \text{on } \Omega \\ \partial_n u = 0, & \text{on } \partial\Omega \end{cases}$$

- Optimal control problem
- Model has similarites to Belhachmi et al. (2009)
- Large scale optimisation (often $\geqslant 100\,000$ unknowns)
- $\lambda|c|$ is non-differentiable
- Constraint is non-convex due to mixed products in c and u.

A Solution Strategy

- Linearise constraint to handle non-convexity:

$$T(u,c) \coloneqq c\,(u-g) + (1-c)(-\Delta)u$$
$$T(u,c) \approx T(u^k,c^k) + D_u T(u^k,c^k)(u-u^k) + D_c T(u^k,c^k)(c-c^k)$$

- Add proximal term and solve iteratively

$$\underset{u,\,c}{\arg\min} \left\{ \int_\Omega \frac{1}{2}(u-g)^2 + \lambda |c| + \frac{\varepsilon}{2}|c|^2 + \frac{\mu}{2}\left(u-u^k\right)^2 + \frac{\mu}{2}\left(c-c^k\right)^2 \mathrm{d}x \right\}$$

$$T(u^k,c^k) + D_u T(u^k,c^k)(u-u^k) + D_c T(u^k,c^k)(c-c^k) = 0$$

until fixed point is reached.

Algorithmic Details

- Approach is similar to
  - LCL algorithm (Murthagh et al. 1982),
  - EM/MM method (Orthega et al. 1970).
- Linearised problem is convex and easier to solve.
- Derivation of the conjugate dual problem is possible.
  - unconstrained convex optimisation problem
  - solvable via gradient descent
  - may yield more accurate solutions and faster convergence

Theoretical Findings

Algorithm yields several interesting results:

1. Energy is decreasing as long as:

$$\frac{1}{2}\left(\|u^{k+1}-g\|_2^2-\|u^k-g\|_2^2\right)\leqslant \lambda\left(\|c^{k+1}\|_1-\|c^k\|_1\right)+\frac{\varepsilon}{2}\left(\|c^{k+1}\|_2^2-\|c^k\|_2^2\right)$$

   Gain in sparsity must outweigh loss in accuracy.

2. Fixed-points fulfil necessary optimality conditions:

$$\begin{aligned} u-g-D_uT(u,c)^\top p &= 0\\ \lambda\partial\left(\|\cdot\|_1\right)(c)+\varepsilon c+D_cT(u,c)\,p &\ni 0\\ T(u,c) &= 0 \end{aligned}$$

Grey Scale Image Example

Input
(256×256)

Non-binary mask
(5% entries)

Reconstruction
(MSE: 16.9)

Even textured areas are recovered!

Colour Image Example

- Colour images are handled in YCbCr space.
- compute mask on Y channel
- use same mask to inpaint all channels

Input
(511×511)

Non-binary mask
(4% entries)

Reconstruction
(MSE: 24.4)

Differences are barely visible!

From Mask to Grey Value Optimisation (GVO)

- Control model optimises mask values and positions in c.
- Grey values g can be optimised, too.
- Does this help to reduce the error even further?
- Can it be done efficiently?

Grey Value Optimisation (GVO)

- Let mask c be fixed and $M(c)$ the inpainting operator.

  Mainberger et al. (2012) suggest to optimise the grey values g:

Grey Value Optimisation

$$f = \arg\min_{x} \{ \frac{1}{2} \| M\,(c)\, x - g \|_2^2 \}$$

Experiments suggest:

- GVO yields huge improvements for **binary** masks.
- GVO has no effect for **optimised non-binary** masks c.
- Both strategies yield **same** error.

Grey Value Optimisation (GVO)

- Let mask c be fixed and $M(c)$ the inpainting operator.

  Mainberger et al. (2012) suggest to optimise the grey values g:

Grey Value Optimisation

$$f = \arg\min_{x} \{ \frac{1}{2} \| M(c)\, x - g \|_2^2 \}$$

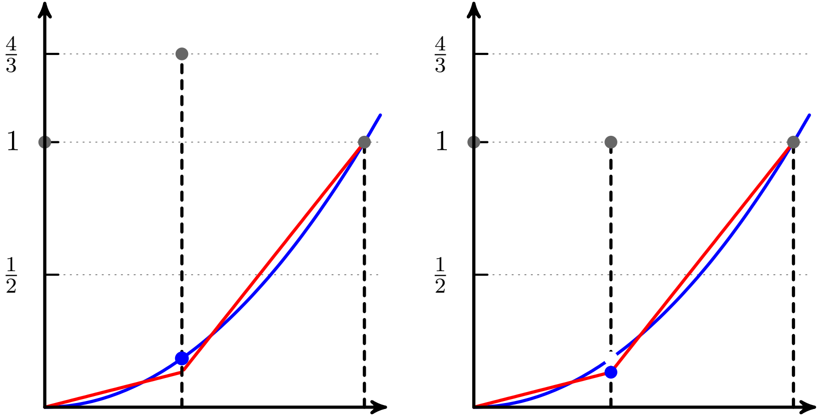
Experiments suggest:

- GVO yields huge improvements for **binary** masks.
- GVO has no effect for **optimised non-binary** masks c.
- Both strategies yield **same** error.

Coincidence?

Which Data Can Be Optimised?

Optimising grey values and mask values can give the same results.



Mask Value Optimisation   Grey Value Optimisation

New Findings

- Complex relationship between inpainting data g and mask c:

$$\begin{aligned} c\,(u-g) + (1-c)(-\Delta)u &= 0, \quad \text{on } \Omega \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \end{aligned}$$

- **Important:** Mask locations coincide with data locations.

Equivalence in the data optimisation (EMMCVPR 2015)

If mask positions are **fixed** by set K:
Mask value optimisation is equivalent to grey value optimisation.

$$\min_{c_i,\, i\in K} \{\|M\,(c)\,g - g\|_2^2\} = \min_{x_i,\, i\in K} \{\|M\,(\bar{c})\,x - g\|_2^2\}$$

where $\bar{c}$ is a binary mask for K.

Benefits from This Equivalence

- GVO is much simpler than mask value optimisation.
  - least squares vs. non-convex optimisation task
- Less memory requirements allow image compression schemes.
  - mask values need not be saved
  - reduces file size by approximately 30%
  - vital to develop feasible image compression codec
- heuristic to speed up convergence for the optimal control solver
  - threshold mask values during iterative scheme
  - may reduce run time from 15 hours to 5 minutes

Fast Numerics for Grey Value Optimisation

- finding optimal mask values is time consuming
  - non-convex and non-smooth optimisation task
- finding optimal grey values is a least squares problem
  - can be solved efficiently
- use specialised algorithms for target environment
  - LSQR (Paige & Saunders, 1982) based solver for CPUs
  - primal dual (Chambolle & Pock, 2011) solver for GPUs

Benchmarks

Run times with comparison to Mainberger et al. (2012)

Image Size, (5% mask)	CPU [s]			GPU [s]
	Mainberger et al.	LSQR	PD	PD
64×64	156.33	2.69	5.82	1.28
128×128	3116.70	18.73	52.57	3.33
256×256	95 832.64	113.07	260.26	9.01

GPU results by Sebastian Hoffmann

Speedup factor 850 on CPU and 10 000 on GPU.

Sequential vs. Combined Data Optimisation

- Tuning of mask and grey values happens sequentially.
- How much better would a combined optimisation be?
- Combined optimisation is difficult in general.
- Experiments suggest:
  - Slightly better quality
  - Significantly higher run time
- Gain in quality does not justify computational burden.

Image Encoding

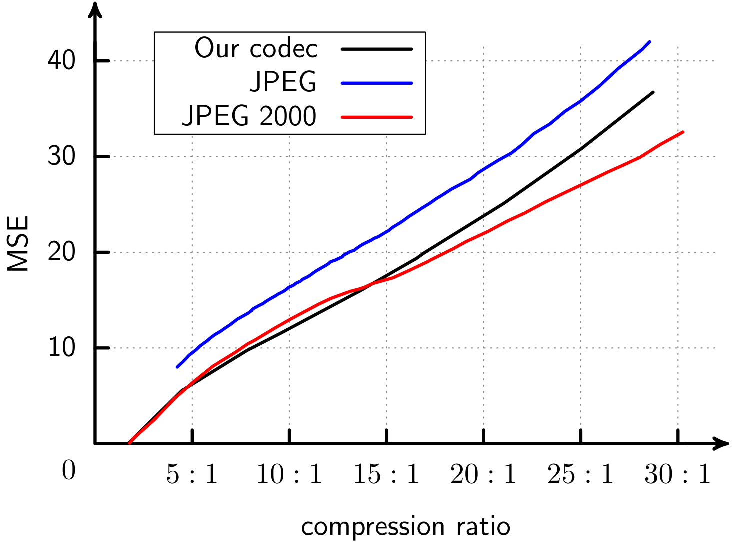
Joint work with Pascal Peter.

We compress a given (grey scale) image as follows.

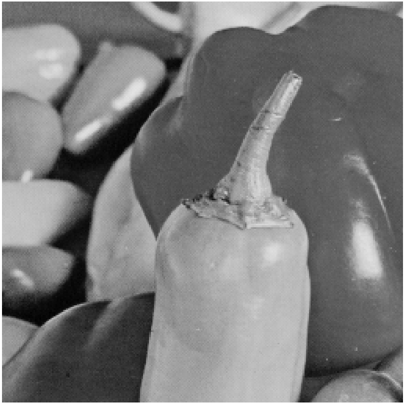
1. computation of a binary mask with optimal control model
2. computation of optimal grey values
3. quantisation optimisation to a finite number of colours
   uses algorithm of Schmaltz et al. (2009)
4. encoding of all the data in a container file
5. application of a high performing entropy coder (PAQ)

Decompression is done in reverse order with a final inpainting.

Comparison to Industry Standards



Visual Comparison



Original image (256×256)

Compressed image (13.6 : 1)
(MSE: 15.97)

Positives and Negatives

- ✔ rather simple approach
- ✔ mathematically well founded
- ✔ competitive to state-of-the-art methods
- ✔ extensions to videos and colour images are straightforward
- ✗ extremely slow (runtime of hours/days)
- ✗ unable to handle textures
- ✗ parameter tuning is difficult

Summary

- generalisation of previous results on PDE-based inpainting
- extension of free knot optimisation to 2D
- interesting equivalence between mask and pixel optimisation
- a novel approach to find optimal inpainting data
- new and efficient numerical schemes

Outlook

- complete convergence theory
- faster numerics
- handling of textured images
- extensions to higher order and non-linear operators

Thank you for your attention