Second Order Differences of Cyclic Data and Applications in Variational Denoising

Ronny Bergmann*

Image Processing Group
Departement of Mathematics
University of Technology Kaiserslautern

September 3rd, 2014

Mathematical Signal Processing and Phase Retrieval
Göttingen

FELIX KLEIN
ZENTRUM FÜR
MATHEMATIK

*joint work with A. Weinmann, G. Steidl, F. Laus

Outline

Introduction

- Employing the Rudin-Osher-Fatemi (ROF) functional [Osher, Rudin, Fatemi, 1992]

$$\sum_{i,j}(f_{i,j} - x_{i,j})^2 + \lambda \sum_{i,j} |\nabla x_{i,j}|$$

  - ∇ discrete gradient
  - $\sum_{i,j}|\nabla x_{i,j}|$ discrete total variation (TV)
  - regularization parameter $\lambda > 0$

$\Rightarrow$ edge-preserving

- stair caising-effect: reduced by adding higher order derivatives

Recently [Cremers,Strekalovski, 2012], [Lellmann et al., 2013], [Weinmann et al., 2013], [Bačák, 2013]

- TV denoising generalized to Riemannian manifolds
- several algorithms to find the minimizer x^*
- convergence for CAT(0) spaces
- $\mathbb{S}^1$ is not a CAT(0) space

First & Second Order Differences on $\mathbb{R}$

A short reminder.

Given a weight $w = (w_j)_{j=1}^d \in \mathbb{R}^d \setminus \{0\}$, i.e. $\langle w, 1_d \rangle := \sum_{j=1}^{d} w_j = 0$,

the finite difference operator is given by

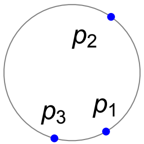
$$\Delta(x; w) := \langle x, w \rangle, \quad x \in \mathbb{R}^d$$

Examples

- $b_1 := (-1, 1)$: First order difference $\Delta(x; b_1) = x_2 - x_1$
- $b_2 := (1, -2, 1)$: Second order difference $\Delta(x; b_2) = x_1 - 2x_2 + x_3$
- $b_{1,1} := (-1, 1, 1, -1)$: ‘mixed second order difference’ $\Delta(x; b_{1,1}) = -x_1 + x_2 + x_3 - x_4$

First & Second Order Difference on $\mathbb{S}^1$

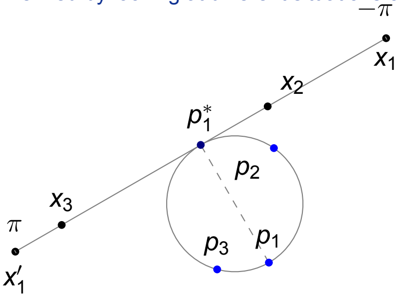
Defined by looking at different situations on $\mathbb{R}$ the points may take.



- $x_i \in [-\pi, \pi) \Leftrightarrow p_i := (\cos x_i, \sin x_i)$
- Idea: unwrap the circle onto any tangential line

First & Second Order Difference on $\mathbb{S}^1$

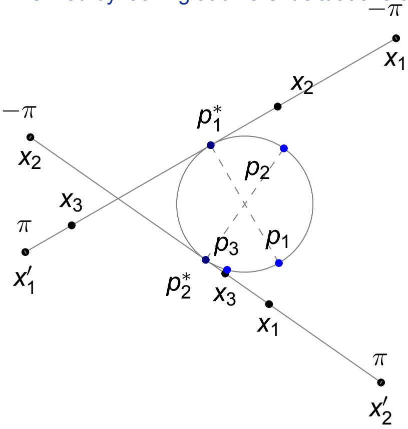
Defined by looking at different situations on $\mathbb{R}$ the points may take.



- $x_i \in [-\pi, \pi) \Leftrightarrow p_i := (\cos x_i, \sin x_i)$
- Idea: unwrap the circle onto any tangential line

First & Second Order Difference on $\mathbb{S}^1$

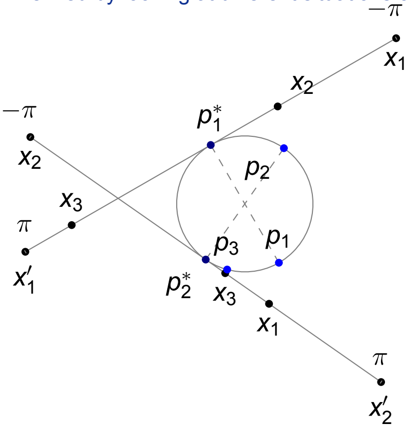
Defined by looking at different situations on $\mathbb{R}$ the points may take.



- $x_i \in [-\pi, \pi) \Leftrightarrow p_i := (\cos x_i, \sin x_i)$
- Idea: unwrap the circle onto any tangential line

First & Second Order Difference on $\mathbb{S}^1$

Defined by looking at different situations on $\mathbb{R}$ the points may take.



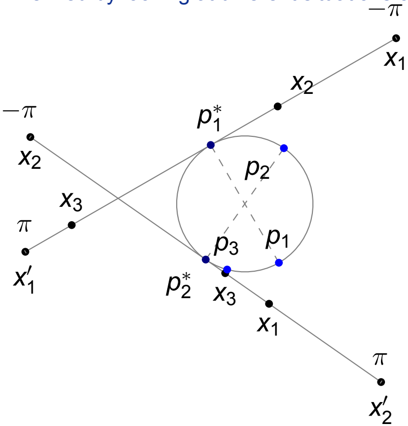
- $x_i \in [-\pi, \pi) \Leftrightarrow p_i := (\cos x_i, \sin x_i)$
- Idea: unwrap the circle onto any tangential line
- Absolute cyclic differences w.r.t w:

$$d(x; w) := \min_{\alpha \in \mathbb{R}} \left| \Delta\left([x + \alpha 1_d]_{2\pi}; w\right) \right|$$

- $[x]_{2\pi}$: element-wise mod 2π
  except $x_i = (2k+1)\pi$: take both $\pm\pi$

First & Second Order Difference on $\mathbb{S}^1$

Defined by looking at different situations on $\mathbb{R}$ the points may take.



- $x_i \in [-\pi, \pi) \Leftrightarrow p_i := (\cos x_i, \sin x_i)$
- Idea: unwrap the circle onto any tangential line
- Absolute cyclic differences w.r.t w:

$$d(x; w) := \min_{\alpha \in \mathbb{R}} \left| \Delta\left([x + \alpha 1_d]_{2\pi}; w\right) \right|$$

- $[x]_{2\pi}$: element-wise mod 2π except $x_i = (2k+1)\pi$: take both $\pm\pi$

- b_1: arc length distance $d_1(x_1, x_2) := d(x; b_1) = \left| \left(\Delta(x; b_1)\right)_{2\pi} \right|$
- b_2: $d_2(x_1, x_2, x_3) := d(x; b_2) = \left| \left(\Delta(x; b_2)\right)_{2\pi} \right|$ (similar for $b_{1,1}$)

Second Order Total Variation on the Circle

Transfer the ROF functional to the circle and extend to second order differences.

Let $f = (f_i)_{i=1}^N$ be given data on $\mathbb{S}^1$, $\alpha, \beta \geq 0$.
We are interested in the minimizers $x^* \in (\mathbb{S}^1)^N$ of

$$\varphi(x) := F(x; f) + \alpha\,\mathrm{TV}_1(x) + \beta\,\mathrm{TV}_2(x),$$

where

- data fidelity term $F(x; f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i, x_i)^2$
- first order differences $\mathrm{TV}_1(x) = \sum_{i=1}^{N-1} d_1(x_i, x_{i+1})$
- second order differences $\mathrm{TV}_2(x) = \sum_{i=2}^{N-1} d_2(x_{i-1}, x_i, x_{i+1})$
- similar with both drectional differences and a second order mixed derivative for 2D data.

Proximal Point Algorithms on $\mathbb{R}$

The proximal mapping is defined by

$$\operatorname{prox}_{\lambda\varphi}(f) := \operatorname*{arg\,min}_{x\in\mathbb{R}^N} \frac{1}{2}\|f-x\|_2^2 + \lambda\varphi(x),$$

- $\varphi : \mathbb{R}^N \to (-\infty, +\infty]$ proper, closed, convex function
- $\lambda > 0$ trade-off parameter: minimizing $\varphi(x)$ vs. “staying near” f

Proximal Point Algorithms on $\mathbb{R}$

The proximal mapping is defined by

$$\operatorname{prox}_{\lambda\varphi}(f) := \operatorname*{arg\,min}_{x\in\mathbb{R}^N} \frac{1}{2}\|f - x\|_2^2 + \lambda\varphi(x),$$

- $\varphi\colon \mathbb{R}^N \to (-\infty, +\infty]$ proper, closed, convex function
- $\lambda > 0$ trade-off parameter: minimizing $\varphi(x)$ vs. "staying near" f

Proximal Point Algorithm (PPA): Picard Iteration [Moreau, 1965; Rockafellar, 1976]

$$x^{(k+1)} = \operatorname{prox}_{\lambda\varphi}(x^{(k)}), \quad k \in \mathbb{N}$$

Proximal Point Algorithms on $\mathbb{R}$

The proximal mapping is defined by

$$\operatorname{prox}_{\lambda\varphi}(f) := \operatorname*{arg\,min}_{x\in\mathbb{R}^N} \frac{1}{2}\|f-x\|_2^2 + \lambda\varphi(x),$$

- $\varphi\colon \mathbb{R}^N \to (-\infty,+\infty]$ proper, closed, convex function
- $\lambda > 0$ trade-off parameter: minimizing $\varphi(x)$ vs. "staying near" f

Proximal Point Algorithm (PPA): Picard Iteration [Moreau, 1965; Rockafellar, 1976]

$$x^{(k+1)} = \operatorname{prox}_{\lambda\varphi}(x^{(k)}), \quad k \in \mathbb{N}$$

For $\varphi = \sum_{i=1}^{c} \varphi_i$, where the proximal mappings of φ_i are "easier":

Cyclic Proximal Point Algorithm (CPPA) [Bertsekas, 2011]

$$x^{(k+\frac{i+1}{c})} = \operatorname{prox}_{\lambda_k\varphi_i}(x^{(k+\frac{i}{c})}), \quad i = 0,\ldots,c-1, k \in \mathbb{N}.$$

Converges to a minimizer if $\{\lambda_k\} \in \ell_2(\mathbb{Z})\backslash\ell_1(\mathbb{Z})$

Proximal Mappings on $\mathbb{S}^1$

For cyclic data: $\operatorname{prox}_{\lambda\varphi}(g) = \operatorname*{arg\,min}_{x\in[-\pi,\pi)^N} \frac{1}{2} d_1(g,x)^2 + \lambda\varphi(x)$ [Ferreira, Oliveira, 2002]

Theorem I [B., Laus, Steidl, Weinmann, 2014]

The unique minimizer x^* of $\operatorname{prox}_{\lambda d_1(f,\circ)^2}(g)$ is

$$x^* = \left(\frac{g+\lambda f}{1+\lambda} + \frac{\lambda}{1+\lambda} 2\pi v\right)_{2\pi}, \quad v = \begin{cases} 0 & \text{for } |g-f| \leq \pi, \\ \operatorname{sgn}(g-f) & \text{for } |g-f| > \pi \end{cases}$$

Proximal Mappings on $\mathbb{S}^1$

For cyclic data: $\operatorname{prox}_{\lambda\varphi}(g) = \operatorname*{arg\,min}_{x\in[-\pi,\pi)^N} \frac{1}{2} d_1(g,x)^2 + \lambda\varphi(x)$ [Ferreira, Oliveira, 2002]

Theorem I [B., Laus, Steidl, Weinmann, 2014]

The unique minimizer x^* of $\operatorname{prox}_{\lambda d_1(f,\circ)^2}(g)$ is

$$x^* = \left(\frac{g+\lambda f}{1+\lambda} + \frac{\lambda}{1+\lambda} 2\pi\, v\right)_{2\pi}, \quad v = \begin{cases} 0 & \text{for } |g-f| \leq \pi, \\ \operatorname{sgn}(g-f) & \text{for } |g-f| > \pi \end{cases}$$

Theorem II [B., Laus, Steidl, Weinmann, 2014]

The minimizers of $\operatorname{prox}_{\lambda d(\circ;w)}(g)$, $\quad w \in \{b_1, b_2, b_{1,1}\}$, are given by

$$x^* = \left(g - \operatorname{sgn}([\langle g,w\rangle]_{2\pi}) \cdot \min\left\{\lambda, \frac{|(\langle g,w\rangle)_{2\pi}|}{\|w\|_2^2}\right\} w\right)_{2\pi}$$

For $|(\langle g,w\rangle)_{2\pi}| = \pi$, there are two minimizers, otherwise it is unique.

CPPA with Second Order TV for 1D Data on $\mathbb{S}^1$

How to split the higher order TV functional φ?

- $F(x; f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i, x_i)^2 =: \varphi_1(x)$
  proximal mapping I (applied simultaneously element-wise)
- first order differences

$$\alpha\,\mathrm{TV}_1(x) = \alpha \sum_{i=1}^{N-1} d_1(x_i, x_{i+1})$$

- second order differences

$$\beta\,\mathrm{TV}_2(x) = \beta \sum_{i=2}^{N-1} d_2(x_{i-1}, x_i, x_{i+1})$$

CPPA with Second Order TV for 1D Data on $\mathbb{S}^1$

How to split the higher order TV functional φ?

- $F(x;f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i,x_i)^2 =: \varphi_1(x)$
  proximal mapping I (applied simultaneously element-wise)
- first order differences

$$\alpha\,\mathrm{TV}_1(x) = \sum_{l=0}^{1}\alpha\sum_{i=1}^{\left\lfloor\frac{N-1}{2}\right\rfloor} d_1(x_{2i-1+l},x_{2i-l}) =: \sum_{l=0}^{1}\varphi_{2+l}$$

- second order differences

$$\beta\,\mathrm{TV}_2(x) = \beta\sum_{i=2}^{N-1} d_2(x_{i-1},x_i,x_{i+1})$$

CPPA with Second Order TV for 1D Data on $\mathbb{S}^1$

How to split the higher order TV functional φ?

- $F(x; f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i, x_i)^2 =: \varphi_1(x)$
  proximal mapping I (applied simultaneously element-wise)
- first order differences

$$\alpha\,\mathrm{TV}_1(x) = \sum_{l=0}^{1} \alpha \sum_{i=1}^{\lfloor \frac{N-1}{2} \rfloor} d_1(x_{2i-1+l}, x_{2i-l}) =: \sum_{l=0}^{1} \varphi_{2+l}$$

  inner sum: distinct data $\Rightarrow$ proximal mapping II with $w = b_1$
- second order differences

$$\beta\,\mathrm{TV}_2(x) = \beta \sum_{i=2}^{N-1} d_2(x_{i-1}, x_i, x_{i+1})$$

CPPA with Second Order TV for 1D Data on $\mathbb{S}^1$

How to split the higher order TV functional φ?

- $F(x;f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i,x_i)^2 =: \varphi_1(x)$
  proximal mapping I (applied simultaneously element-wise)
- first order differences
  $$\alpha\,\mathrm{TV}_1(x) = \sum_{l=0}^{1}\alpha\sum_{i=1}^{\left\lfloor\frac{N-1}{2}\right\rfloor} d_1(x_{2i-1+l},x_{2i-l}) =: \sum_{l=0}^{1}\varphi_{2+l}$$
  inner sum: distinct data $\Rightarrow$ proximal mapping II with $w=b_1$
- second order differences
  $$\beta\,\mathrm{TV}_2(x) = \sum_{l=0}^{2}\beta\sum_{i=1}^{\left\lfloor\frac{N-1}{3}\right\rfloor} d_2(x_{3i-2+l},x_{3i-1+l},x_{3i+l}) =: \sum_{l=0}^{2}\varphi_{4+l}(x)$$
  inner sum: distinct data $\Rightarrow$ proximal mapping II with $w=b_2$

CPPA with Second Order TV for 1D Data on $\mathbb{S}^1$

How to split the higher order TV functional φ?

- $F(x; f) = \frac{1}{2}\sum_{i=1}^{N} d_1(f_i, x_i)^2 =: \varphi_1(x)$
  proximal mapping I (applied simultaneously element-wise)
- first order differences

$$\alpha\,\mathrm{TV}_1(x) = \sum_{l=0}^{1} \alpha \sum_{i=1}^{\lfloor \frac{N-1}{2} \rfloor} d_1(x_{2i-1+l}, x_{2i-l}) =: \sum_{l=0}^{1} \varphi_{2+l}$$

inner sum: distinct data $\Rightarrow$ proximal mapping II with $w = b_1$

- second order differences

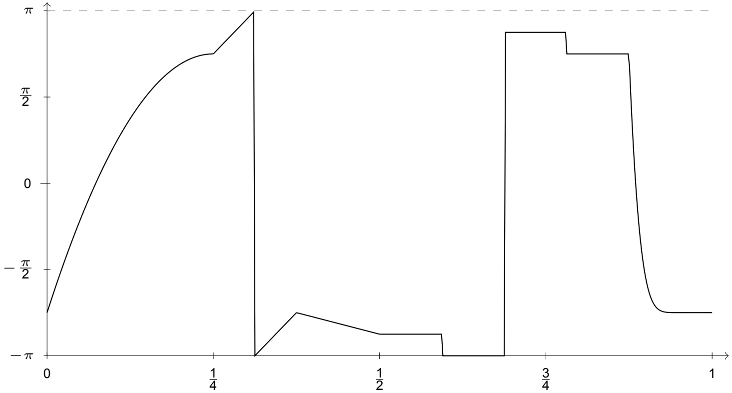
$$\beta\,\mathrm{TV}_2(x) = \sum_{l=0}^{2} \beta \sum_{i=1}^{\lfloor \frac{N-1}{3} \rfloor} d_2(x_{3i-2+l}, x_{3i-1+l}, x_{3i+l}) =: \sum_{l=0}^{2} \varphi_{4+l}(x)$$

inner sum: distinct data $\Rightarrow$ proximal mapping II with $w = b_2$

$\Rightarrow$ $\varphi(x) = \sum_{l=1}^{6} \varphi_l(x)$ $\Rightarrow$ cycle length $c = 6$

Example

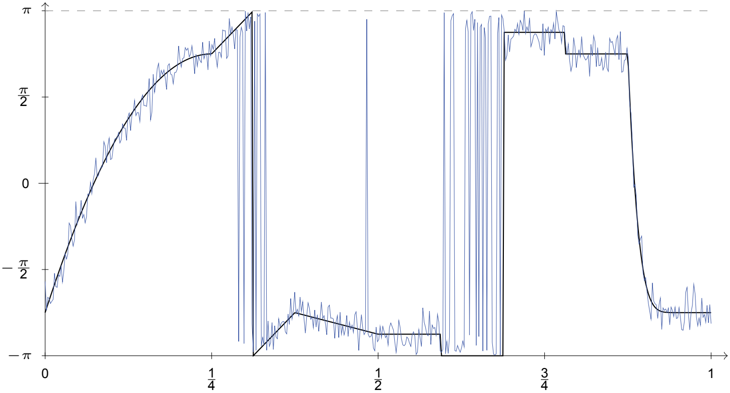
Denoising a 1D phase valued signal.



- function $f\colon [0,1] \to \mathbb{S}^1$ sampled to obtain data $f_\mathrm{o} = (f_{\mathrm{o},i})_{i=1}^{500}$
- jumps $> \pi$ at $\frac{5}{16}$ and $\frac{11}{16}$ are due to the representation system

Example

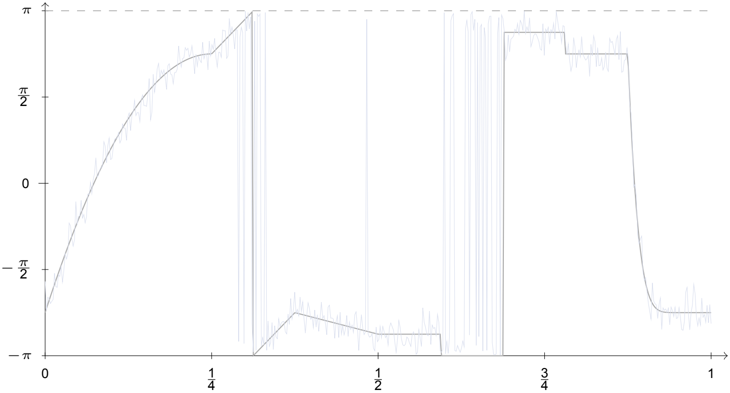
Denoising a 1D phase valued signal.



- function $f\colon [0,1] \to \mathbb{S}^1$ sampled to obtain data $f_o = (f_{o,i})_{i=1}^{500}$
- adding wrapped Gaussian noise, $\sigma = 0.2$
- noisy data $f_n = (f_0 + \eta)_{2\pi}$

Example

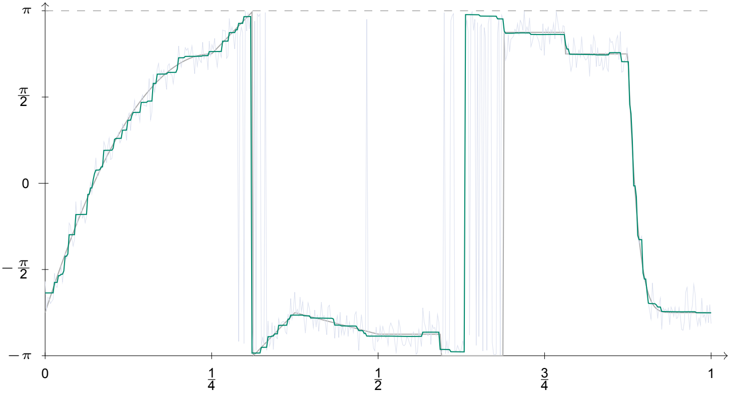
Denoising a 1D phase valued signal.



- comparison of f_o & f_n with

Example

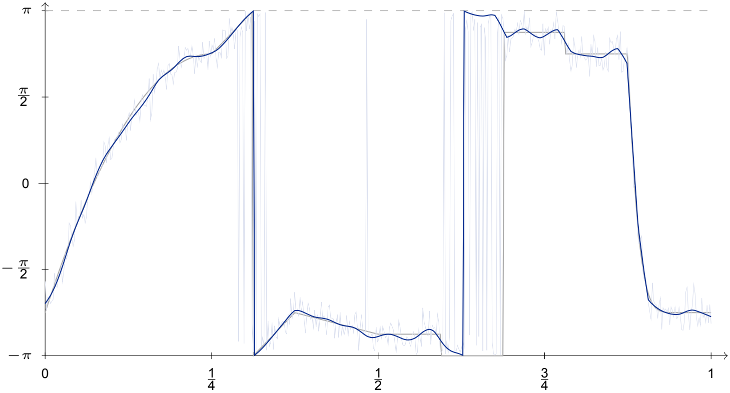
Denoising a 1D phase valued signal.



- comparison of f_o & f_n with f_1
- denoising: just TV_1: $\alpha = \frac{3}{4}$, $\beta = 0$
- but: stair casing

Example

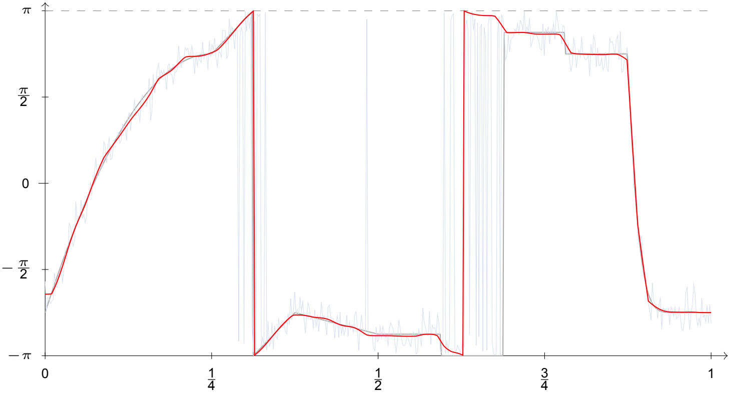
Denoising a 1D phase valued signal.



- comparison of f_o & f_n with f_2
- denoising: just TV_2: $\alpha = 0$, $\beta = \frac{3}{2}$
- but: no plateaus

Example

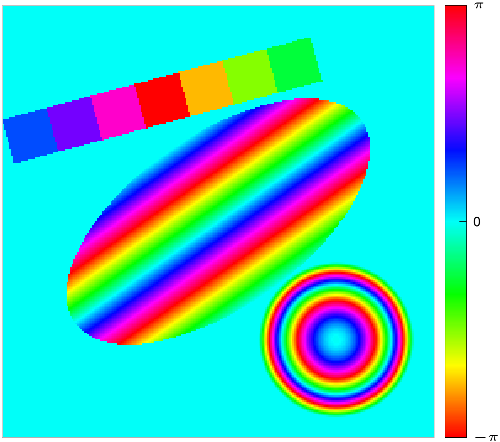
Denoising a 1D phase valued signal.



- comparison of f_o & f_n with f_3
- denoising: TV_1 & TV_2: $\alpha = \frac{1}{2}$, $\beta = 1$
- smallest mean squared error

Example

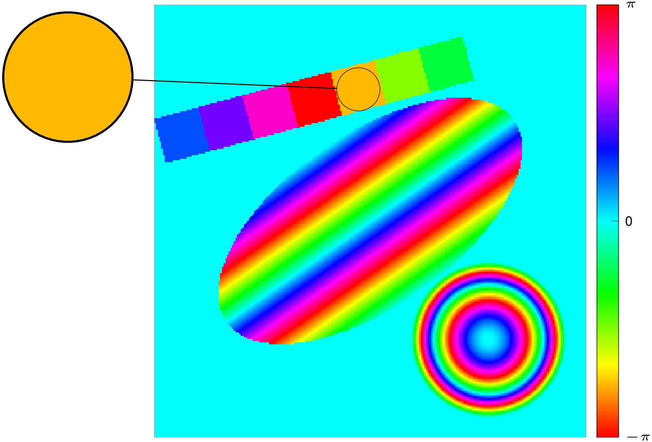
Denoising artificial phase valued data.



original data f_o, 256×256 pixel image

Example

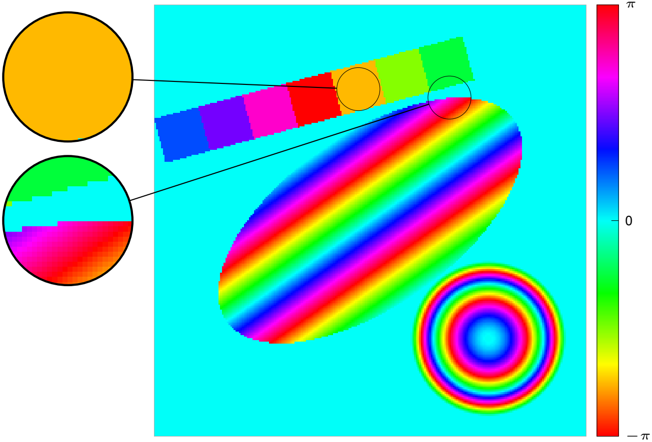
Denoising artificial phase valued data.



original data f_o, 256×256 pixel image

Example

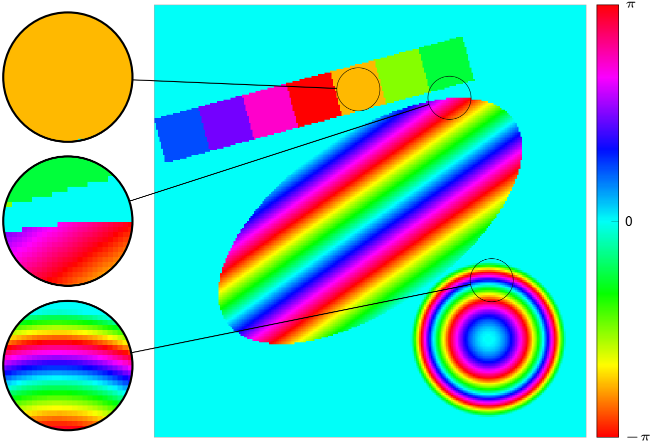
Denoising artificial phase valued data.



original data f_o, 256×256 pixel image

Example

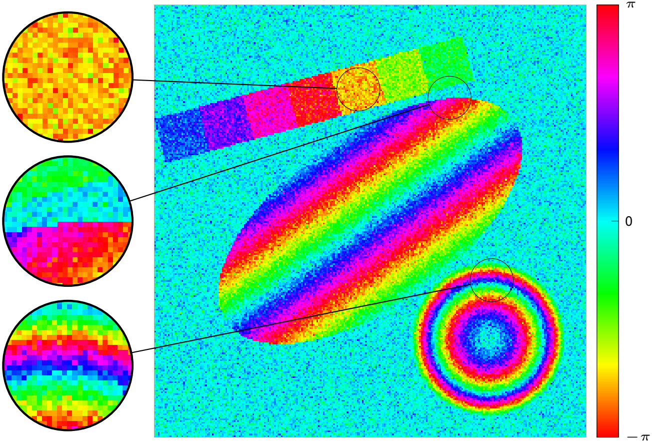
Denoising artificial phase valued data.



original data f_o, 256×256 pixel image

Example

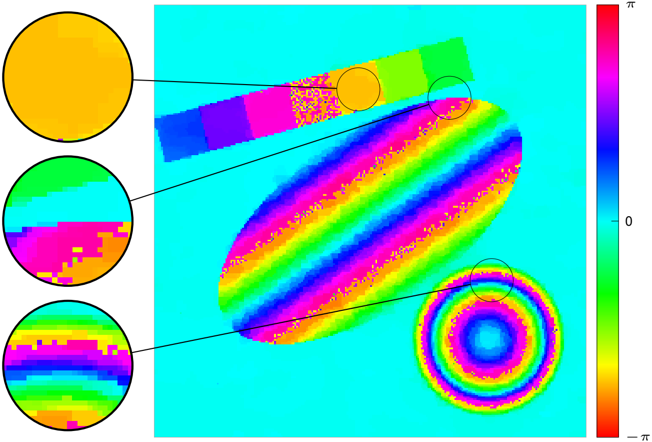
Denoising artificial phase valued data.



noisy data f_n, $\sigma = 0.3$

Example

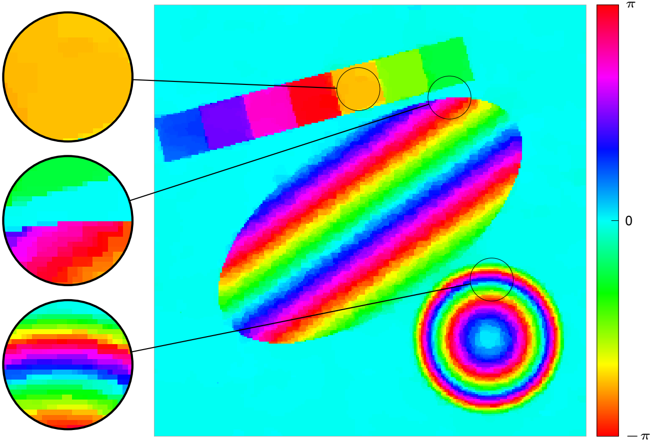
Denoising artificial phase valued data.



denoising with real valued TV
$\alpha_1 = \frac{3}{8}$, $\alpha_2 = \frac{1}{4}$: red part destroyed.

Example

Denoising artificial phase valued data.

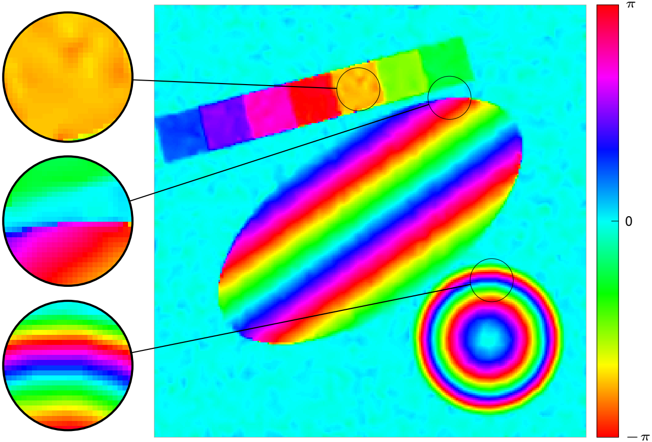


denoising f_n: f_1 with just TV_1

$\alpha_1 = \frac{3}{8}$, $\alpha_2 = \frac{1}{4}$, $\beta_1 = \beta_2 = \gamma = 0$: stair casing

Example

Denoising artificial phase valued data.

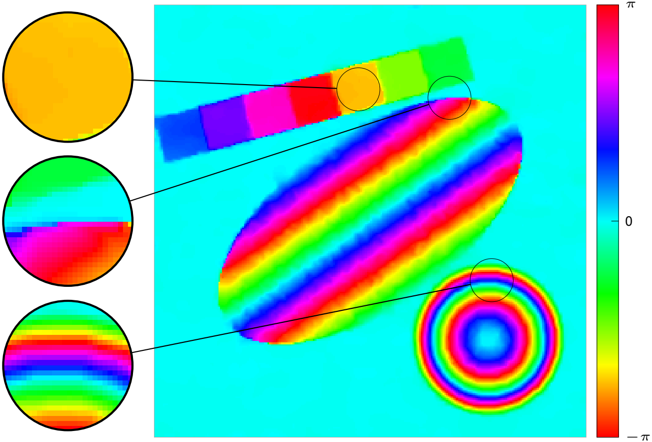


denoising f_n: f_2 with just TV_2

$\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = \gamma = \frac{1}{8}$: no plateaus

Example

Denoising artificial phase valued data.

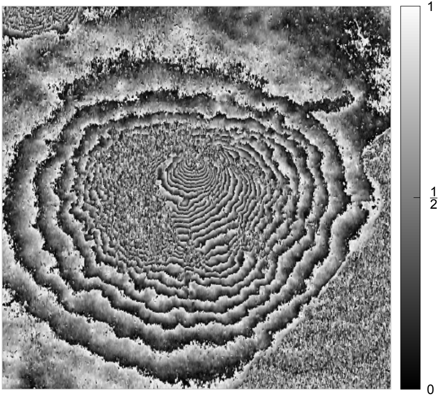


denoising f_n: f_3 with TV_1 & TV_2

$\alpha_1 = \frac{1}{4}$, $\alpha_2 = \beta_1 = \beta_2 = \frac{1}{8}, \gamma = 0$: smallest mean squared error

Mount Vesuvius

The following image is InSAR data from Mount Vesuvius, Italy.[1]

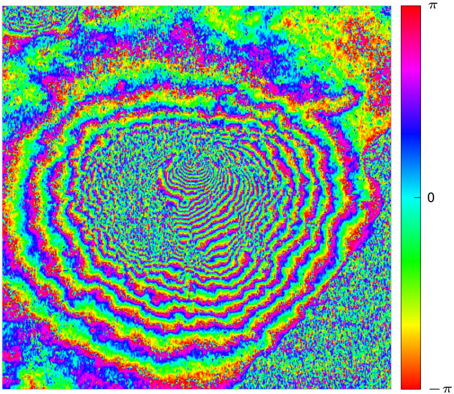


original data, 432×426 pixel

[1] https://earth.esa.int/workshops/ers97/program-details/speeches/rocca-et-al/

Mount Vesuvius

The following image is InSAR data from Mount Vesuvius, Italy.[1]

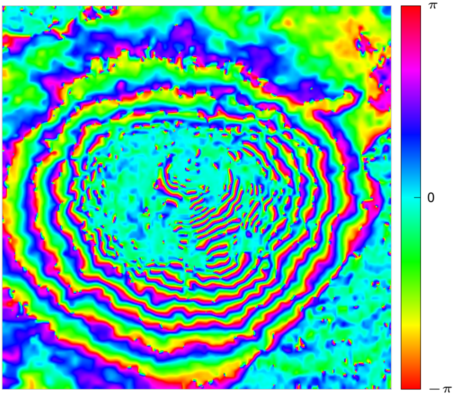


adapted just the coloring

[1] https://earth.esa.int/workshops/ers97/program-details/speeches/rocca-et-al/

Mount Vesuvius

The following image is InSAR data from Mount Vesuvius, Italy.[1]



denoised: $\alpha_1 = \alpha_2 = \frac{1}{4}, \beta_1 = \beta_2 = \gamma = \frac{3}{4}$

[1] https://earth.esa.int/workshops/ers97/program-details/speeches/rocca-et-al/

Summary

We derived for $\mathbb{S}^1$-valued 1D & 2D data f

- higher order differences
- proximal mappings for first and second order differences
- higher order TV functional φ
- an efficient CPPA to minimize φ
- applications:
  - InSAR data denoising
  - hue denoising
  - habituation data
  - Fourier optics, ground based astronomy

We can proof convergence of the algorithm to a minimizer under certain conditions, which are due to the nonconvexity of φ.

Literature

M. Bačák. Computing medians and means in Hadamard spaces. accepted to J. SIAM Opt.

B., F. Laus, G. Steidl, A. Weinmann Second order differences of cyclic data and applications in variational denoising, Preprint, 2014.

D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. Math. Program., Ser. B, 129(2):163–195, 2011.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D., 60(1):259–268, 1992.

E. Strekalovskiy and D. Cremers. Total cyclic variation and generalizations. J. Math. Imaging Vis., 47(3):258–277, 2013.

A. Weinmann, L. Demaret, and M. Storath. Total variation regularization for manifold-valued data. Preprint, 2013.

Literature

M. Bačák. Computing medians and means in Hadamard spaces. accepted to J. SIAM Opt.

B., F. Laus, G. Steidl, A. Weinmann Second order differences of cyclic data and applications in variational denoising, Preprint, 2014.

D. P. Bertsekas. Incremental proximal methods for large scale convex optimization. Math. Program., Ser. B, 129(2):163–195, 2011.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D., 60(1):259–268, 1992.

E. Strekalovskiy and D. Cremers. Total cyclic variation and generalizations. J. Math. Imaging Vis., 47(3):258–277, 2013.

A. Weinmann, L. Demaret, and M. Storath. Total variation regularization for manifold-valued data. Preprint, 2013.

Thank you for your attention.