gipsa-lab

Sparse models and optimization methods for signals and images

Laurent Condat

research scientist of the CNRS, GIPSA-lab, Grenoble, France

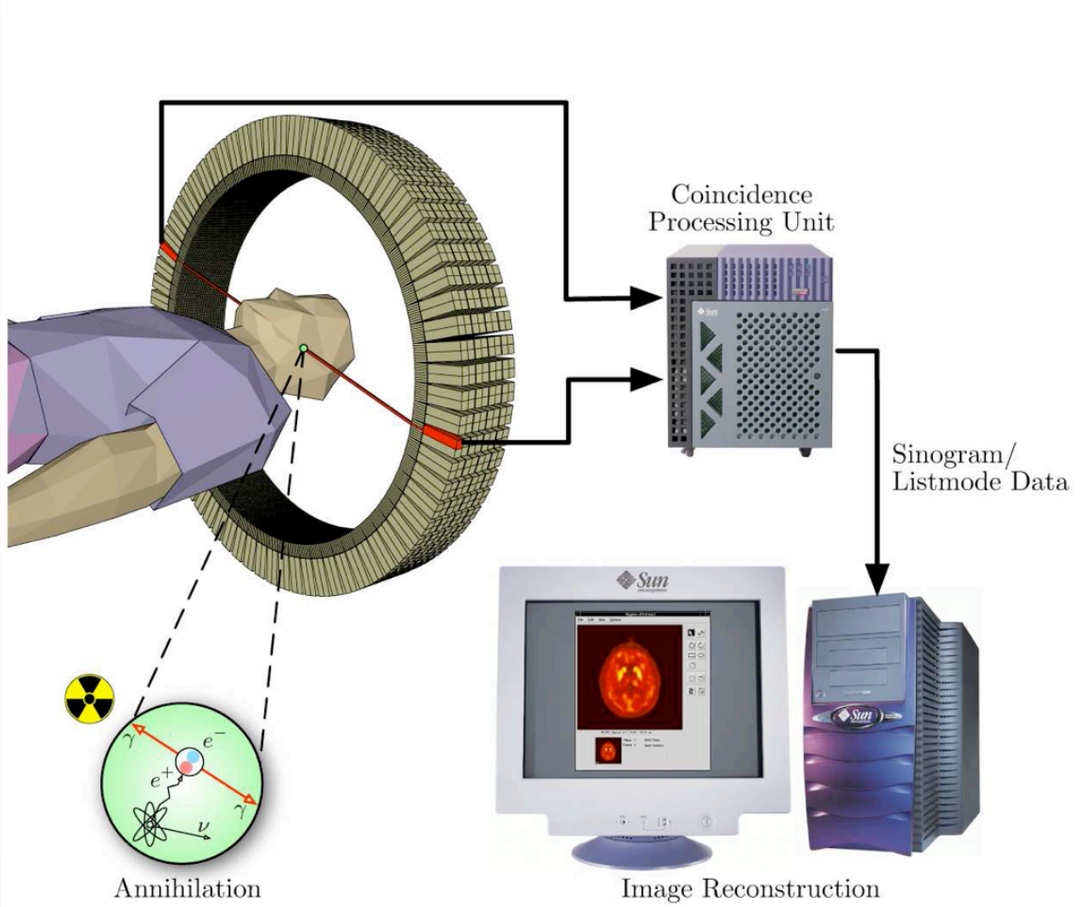
March 3, 2019

Outline

- Context
- Contributions: 2 focuses on
  - A new definition of discrete total variation
  - A new proximal splitting algorithm for convex optimization
- ~~Research plan: ...~~

Context

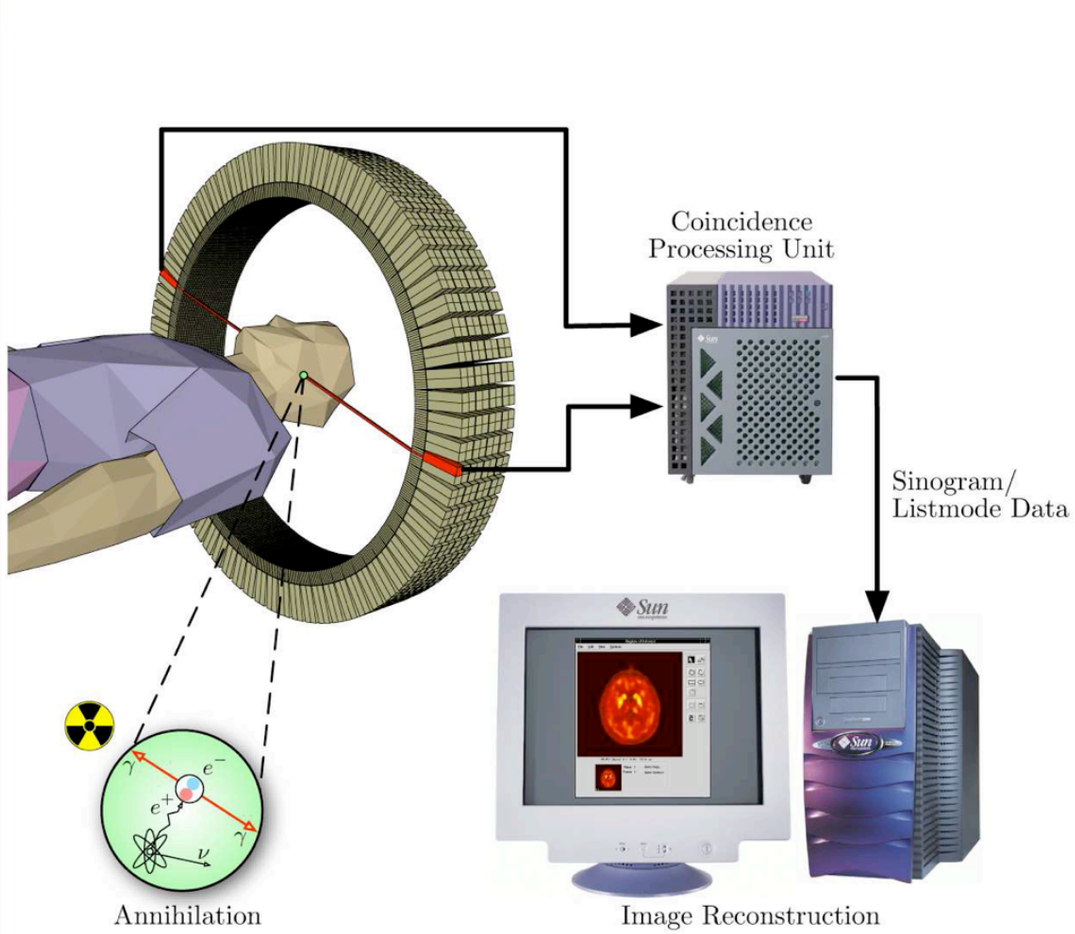
Estimation of signals / images from partial and noisy measurements



Context

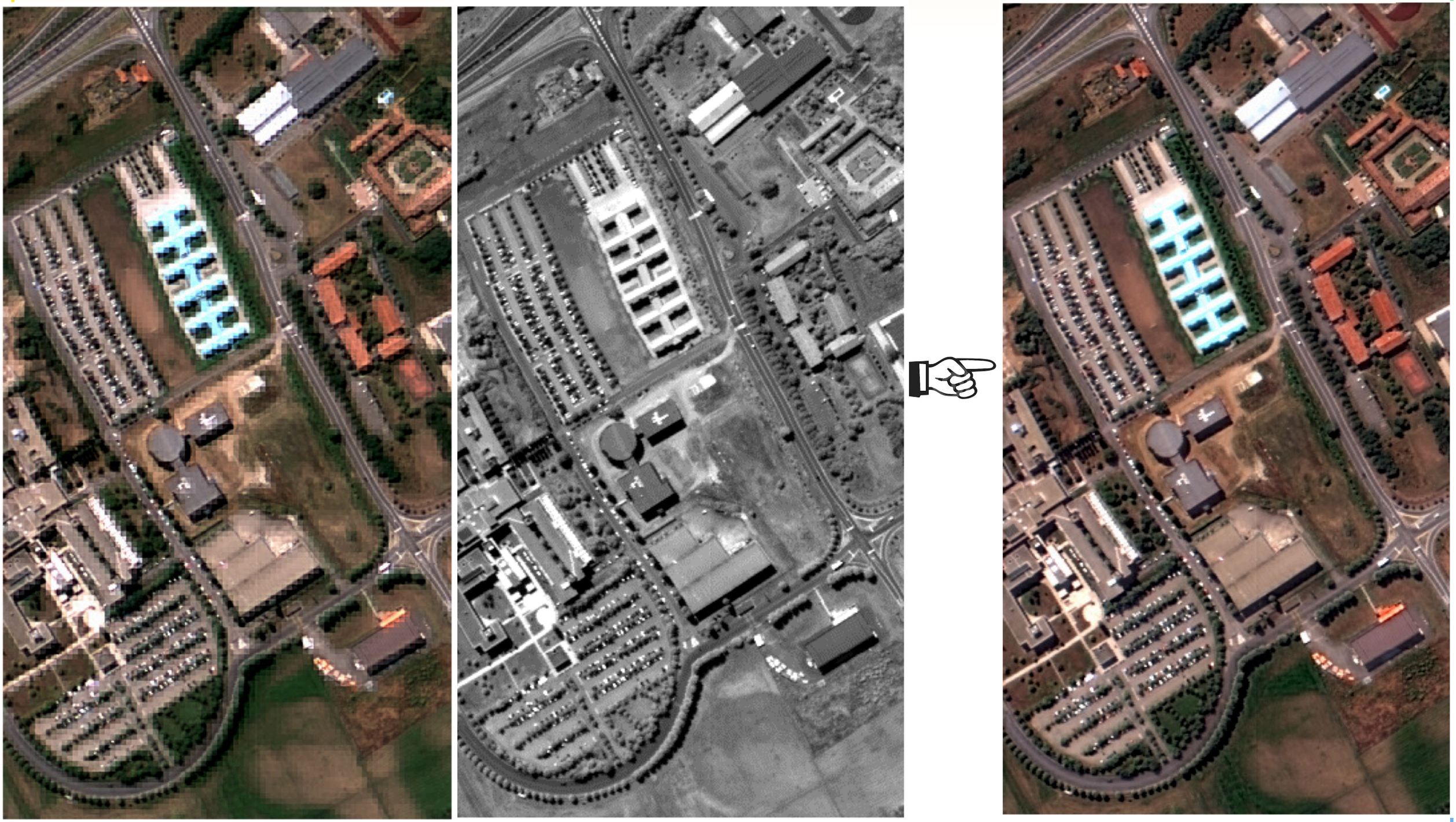
Estimation of signals / images from partial and noisy measurements

- reconstruction method?
- acquisition setting?



Example: Multispectral pansharpening

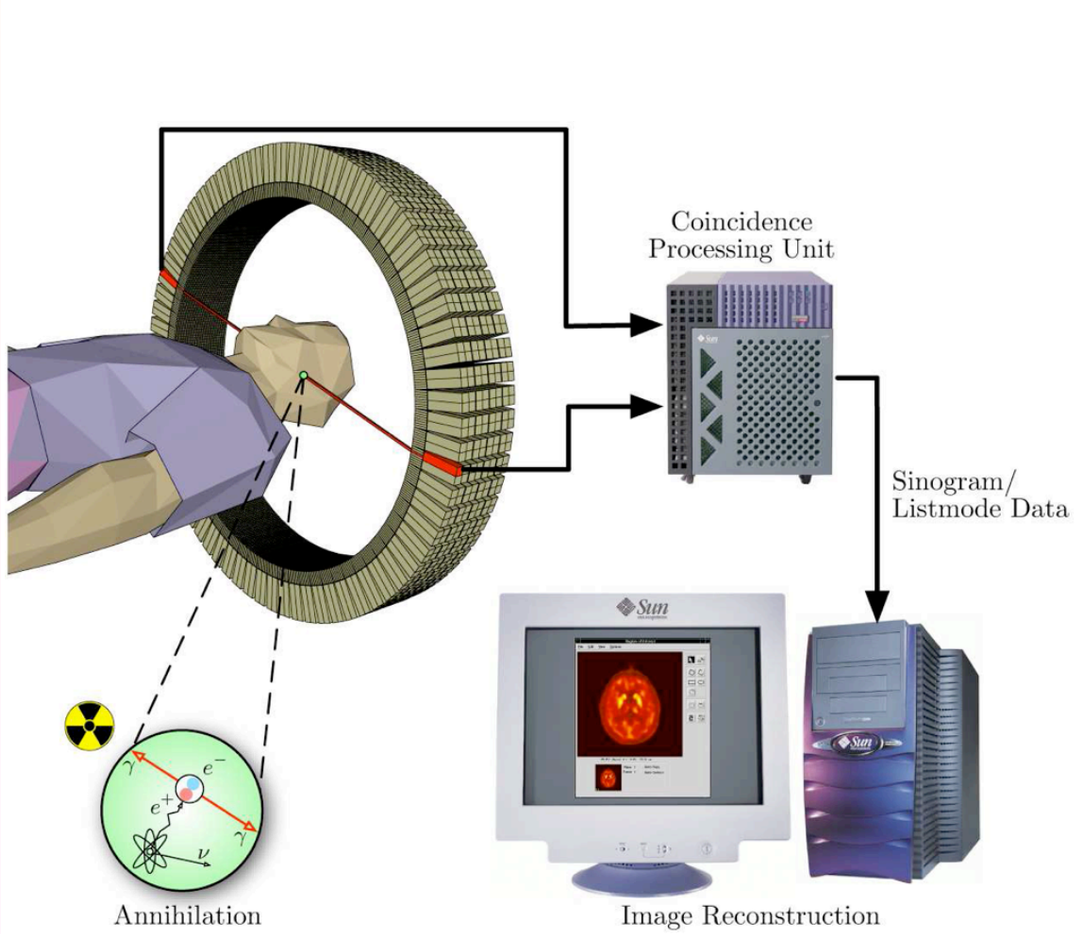
Example: Multispectral pansharpening



X. He, L. C. et al., *IEEE Trans. Image Process.*, 2014

Inverse problems

Given data $y \approx Ax^\sharp$

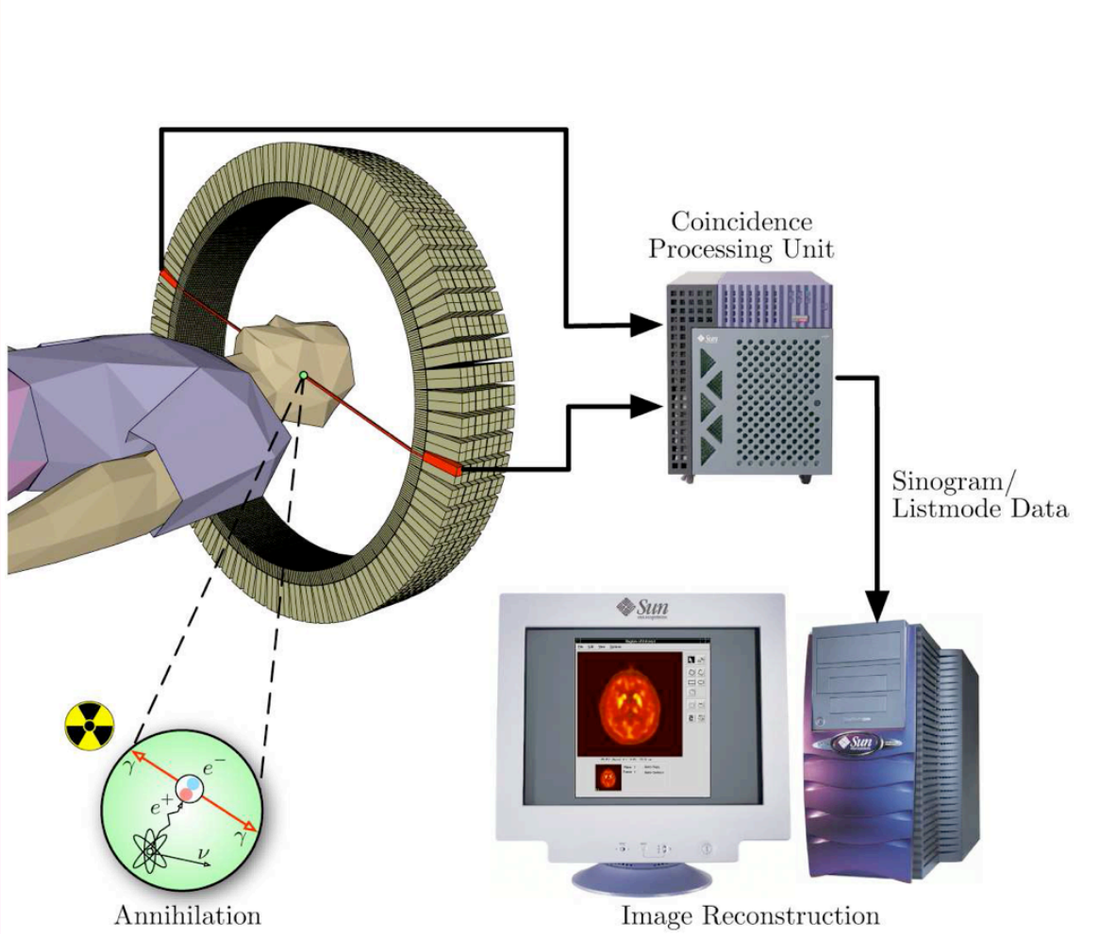


Inverse problems

Given data $y \approx Ax^\sharp$

estimate the unknown image $x^\sharp$ by solving

$$\text{Find } \tilde{x} \in \underset{x \in \mathbb{R}^{N_1 \times N_2}}{\arg\min} \left\{ D(Ax, y) + R(x) \right\}$$



1.

Total variation

L. Condat, “Discrete total variation: New definition and minimization,” *SIAM Journal on Imaging Sciences*, 2017

The total variation

Total variation (TV) of a continuously-defined function $s(t_1, t_2)$:

$$\mathrm{TV}(s) = \int_\Omega \|\nabla s\|$$

The total variation

Total variation (TV) of a continuously-defined function $s(t_1, t_2)$:

$$\mathrm{TV}(s) = \int_\Omega \|\nabla s\|$$

isotropic functional

Discrete TV: Classical Definitions

For an image x with domain $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$,

'anisotropic' TV:

$$\mathrm{TV}_{\mathrm{a}}(x) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \left| x[n_1+1, n_2] - x[n_1, n_2] \right| + \left| x[n_1, n_2+1] - x[n_1, n_2] \right|$$

Discrete TV: Classical Definitions

For an image x with domain $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$,

'anisotropic' TV:

$$\mathrm{TV}_{\mathrm{a}}(x) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \left| x[n_1+1, n_2] - x[n_1, n_2] \right| + \left| x[n_1, n_2+1] - x[n_1, n_2] \right|$$

☞ favors horizontal and vertical edges

Discrete TV: Classical Definitions

For an image x with domain $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$,

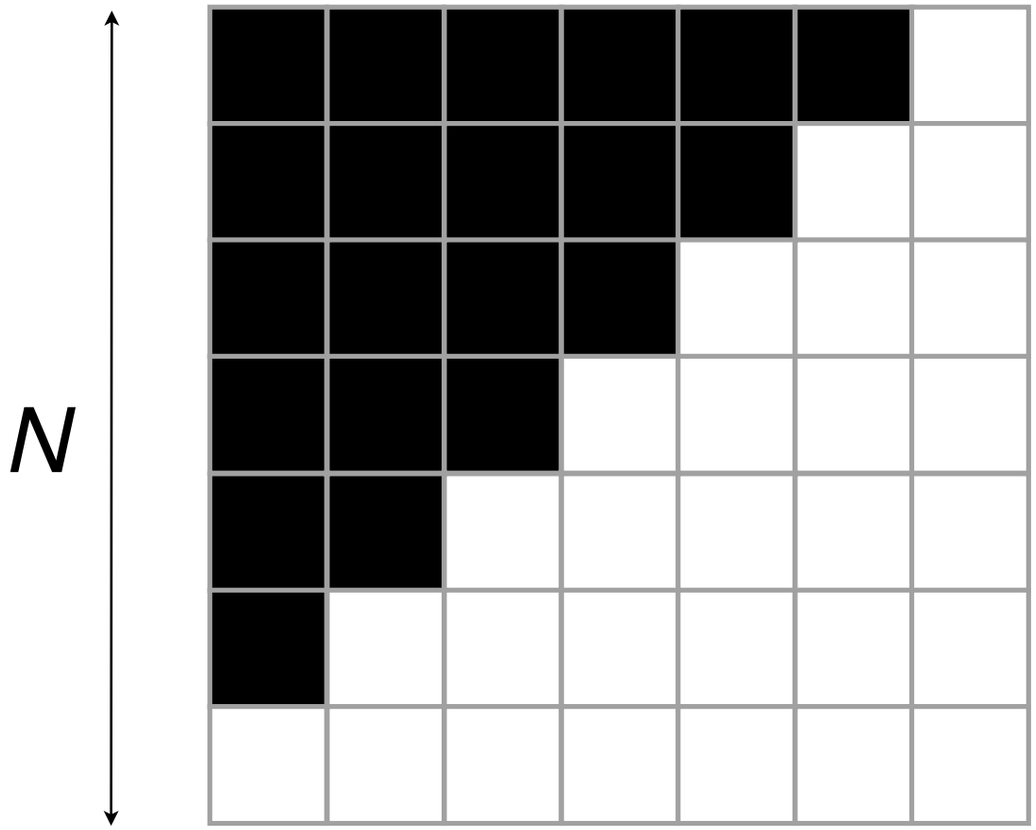
'anisotropic' TV:

$$\mathrm{TV}_{\mathrm{a}}(x) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \left| x[n_1+1, n_2] - x[n_1, n_2] \right| + \left| x[n_1, n_2+1] - x[n_1, n_2] \right|$$

'isotropic' TV:

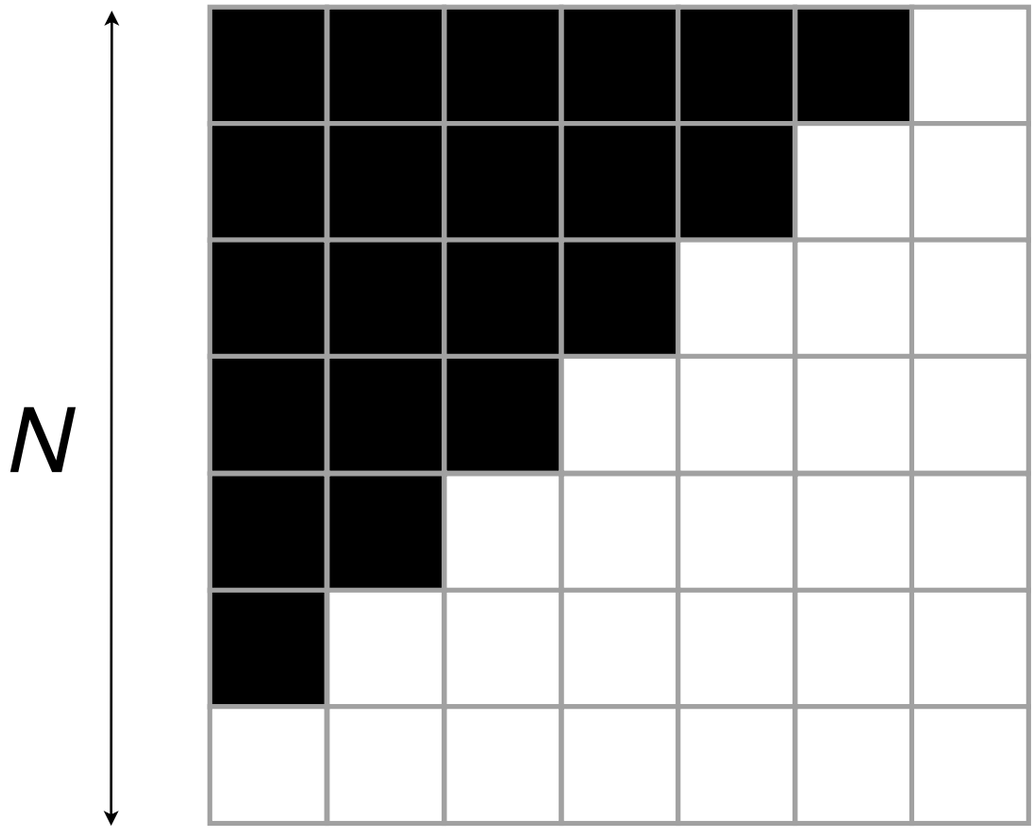
$$\mathrm{TV}_{\mathrm{i}}(x) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \sqrt{\left(x[n_1+1, n_2] - x[n_1, n_2] \right)^2 + \left(x[n_1, n_2+1] - x[n_1, n_2] \right)^2}$$

TV of classical patterns

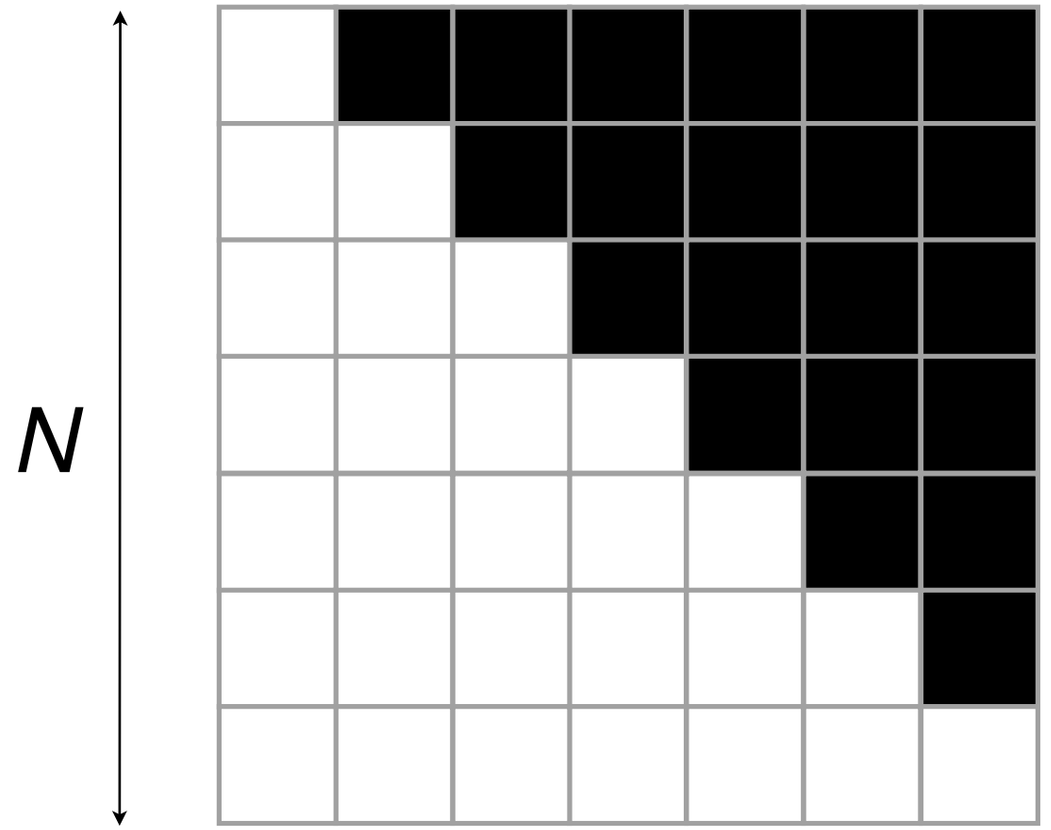


$$TV_i = \sqrt{2}N$$

TV of classical patterns



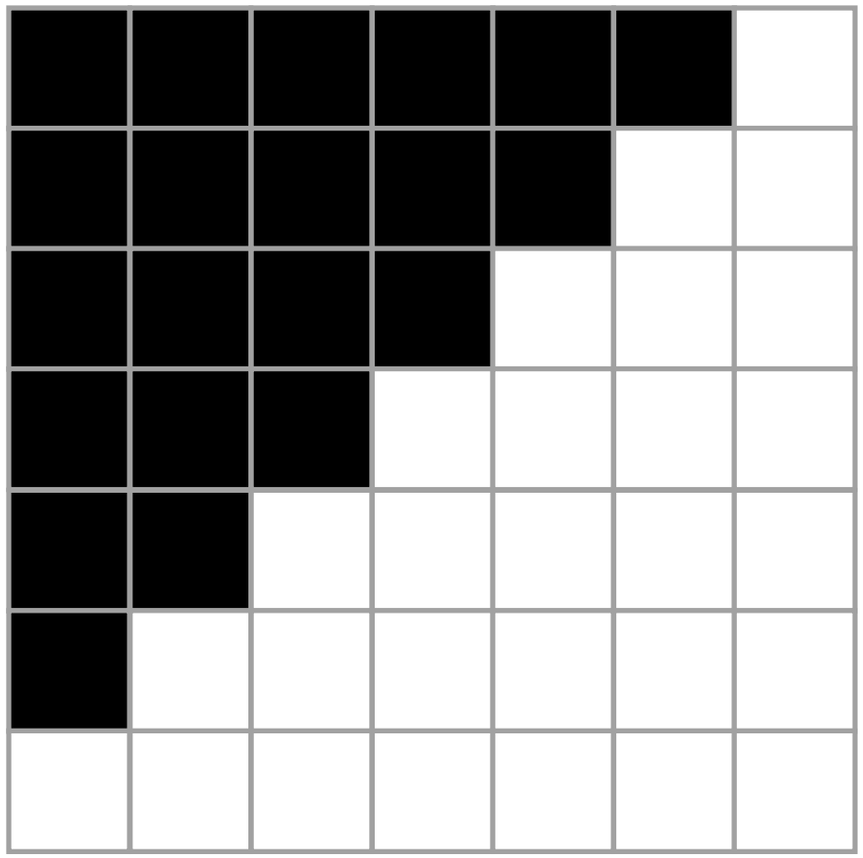
$TV_i = \sqrt{2}N$

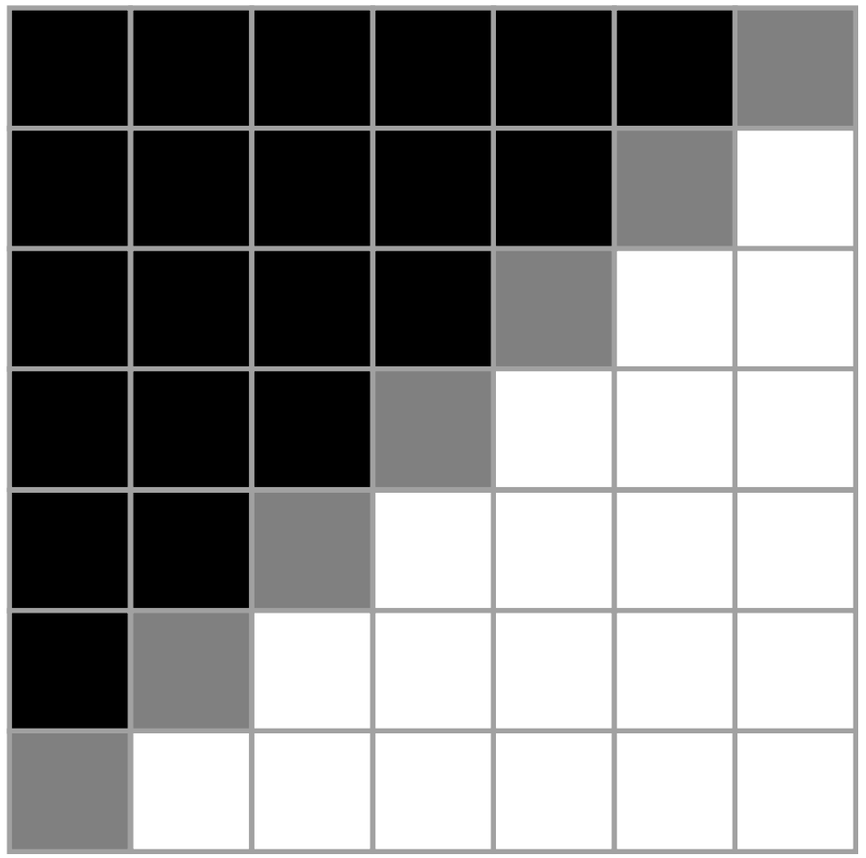


$TV_i = 2N$

The discrete gradient

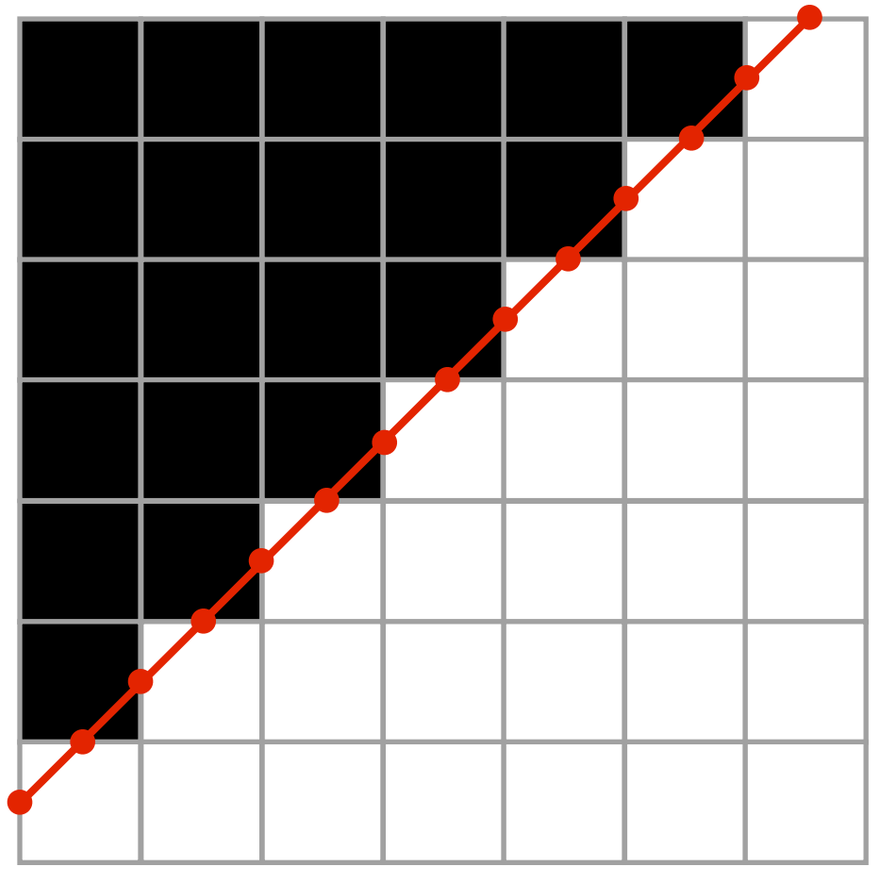
Where is the edge?

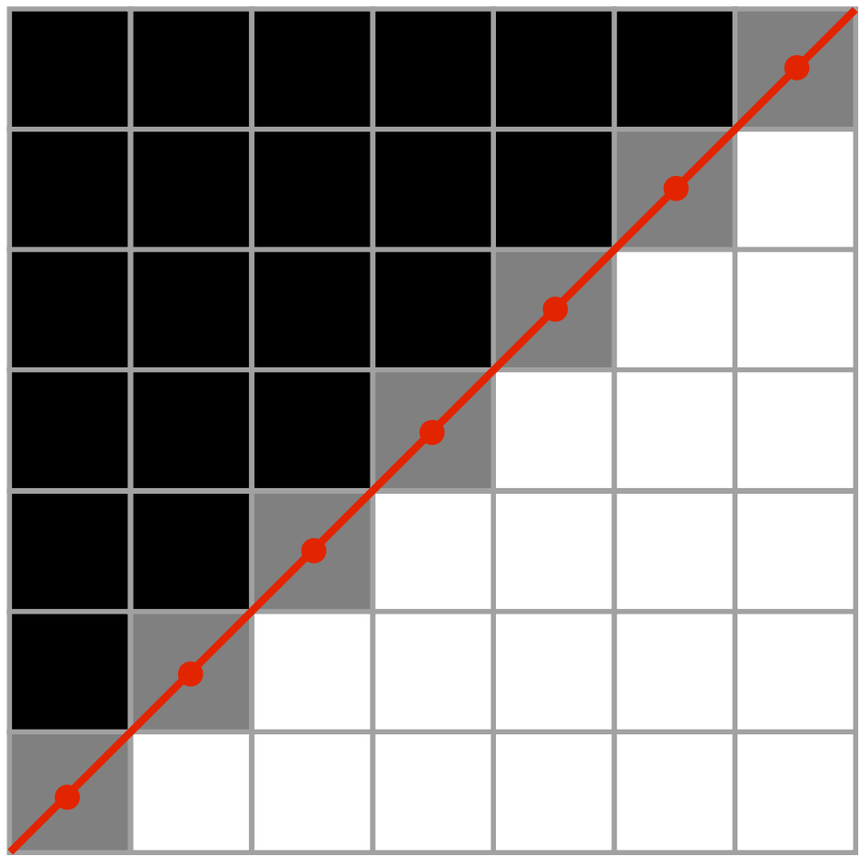




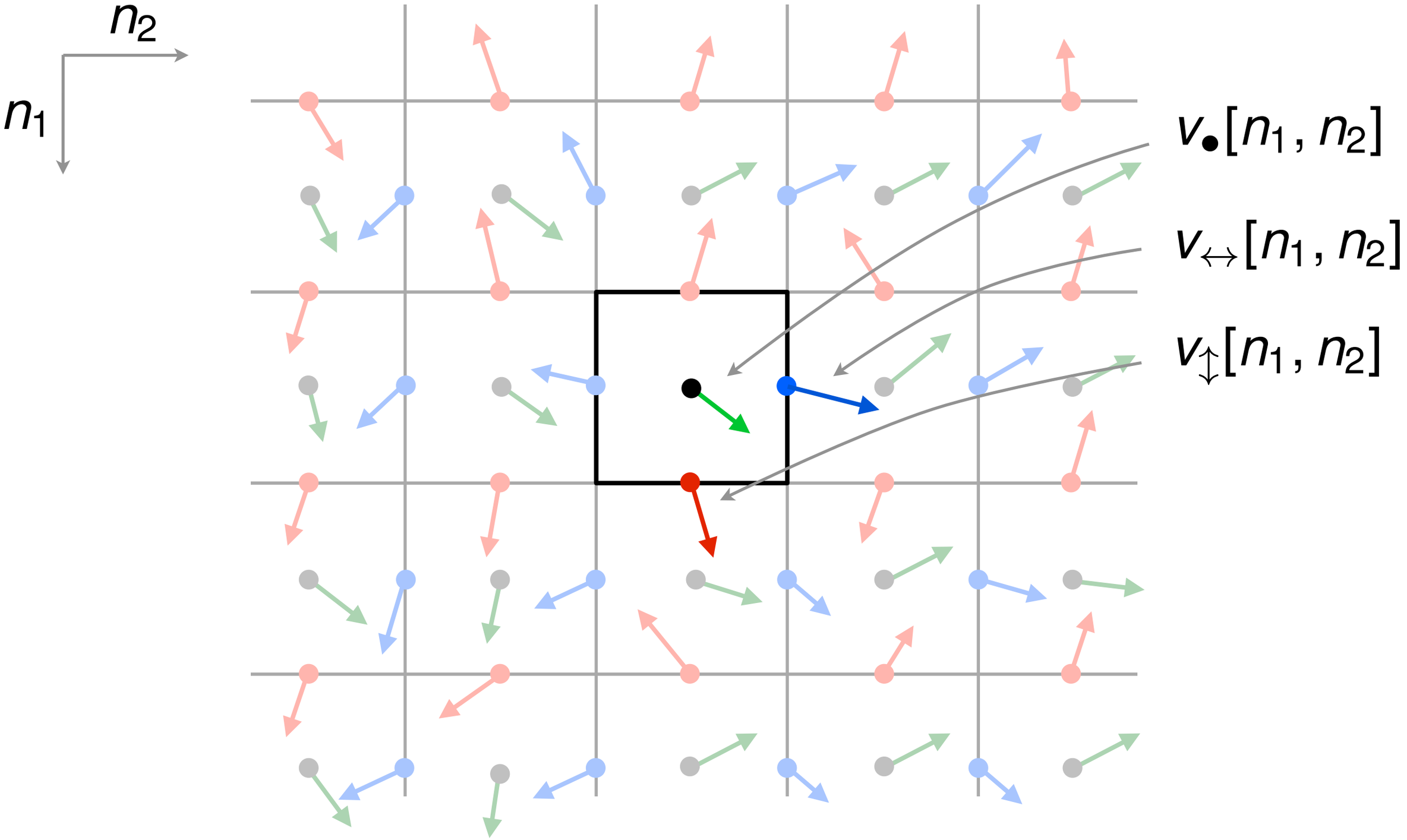
The discrete gradient

Where is the edge?

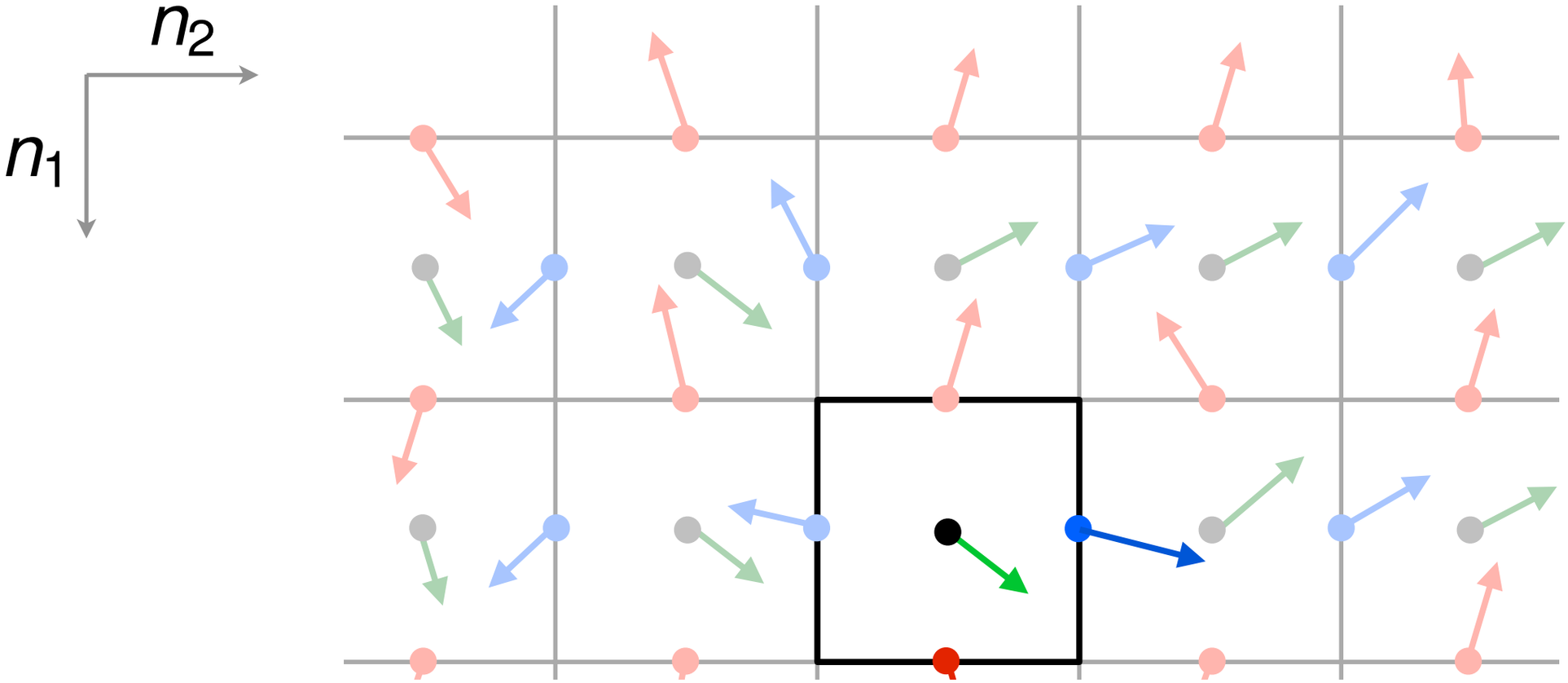




Proposed method: a 2x finer grid for the gradient field



Proposed method: a 2x finer grid for the gradient field



Remember Shannon's theorem:

$$s(t) = \cos(\langle t, c \rangle) \quad \text{☞} \quad \|\nabla s(t)\|^2 \propto \left(1 - \cos(2\langle t, c \rangle)\right)$$

Fundamental theorem of calculus

In 1-D: $\int_a^b s'(t)dt = s(b) - s(a)$

Fundamental theorem of calculus

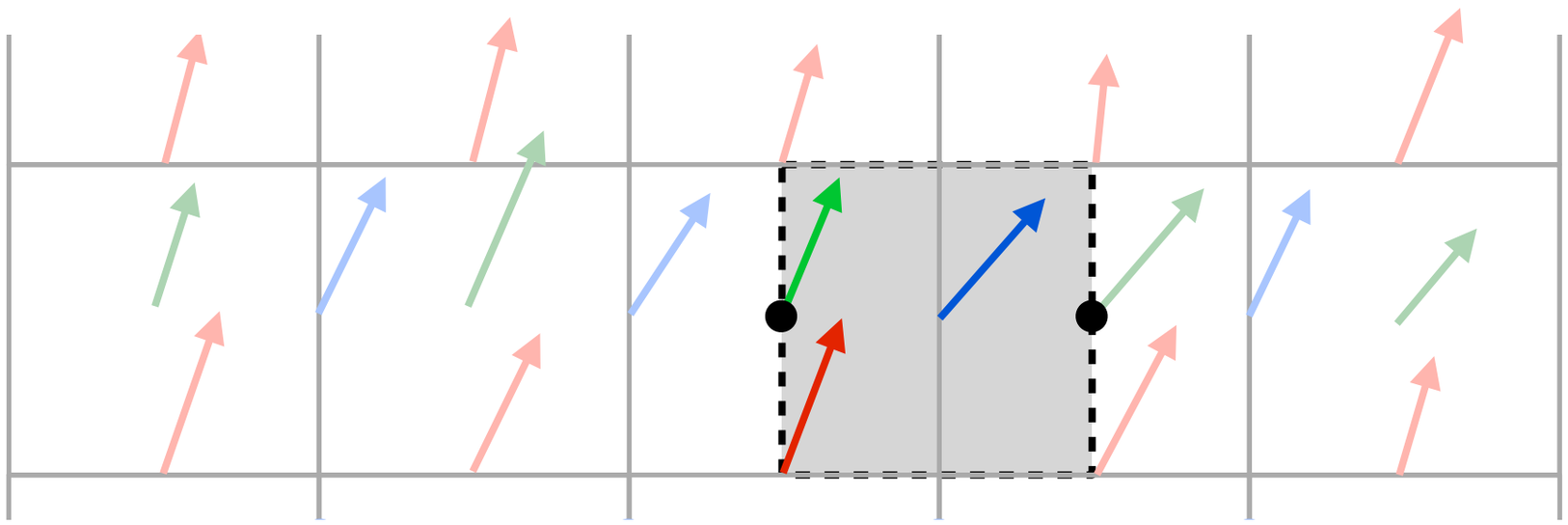
In 1-D: $\int_a^b s'(t)dt = s(b) - s(a)$

By analogy, integrating the gradient field should yield the image (up to a constant)

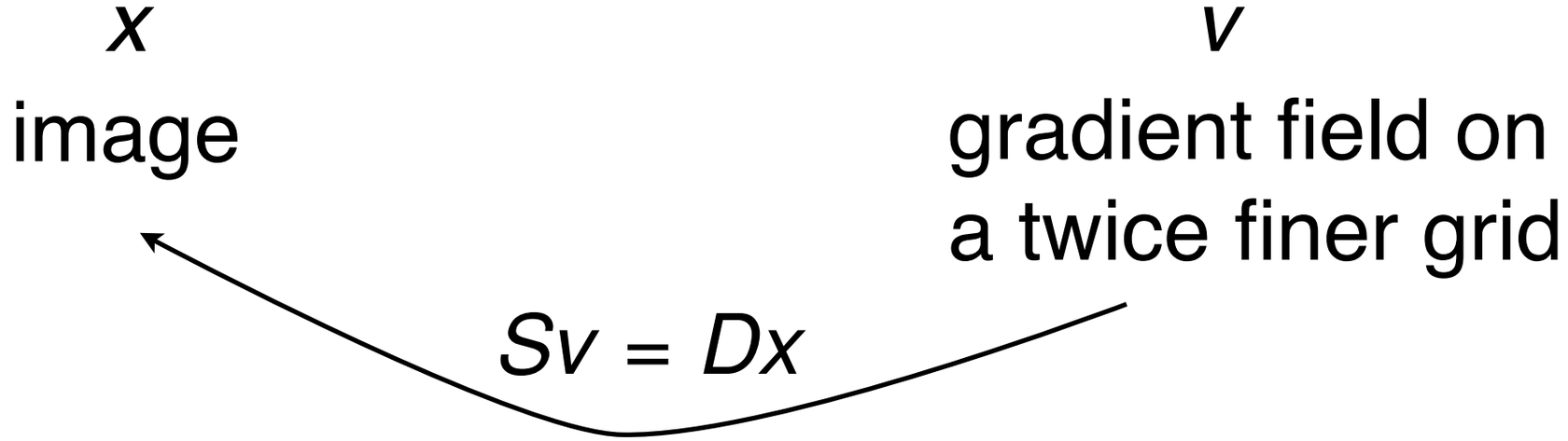
Fundamental theorem of calculus

In 1-D: $\int_a^b s'(t)dt = s(b) - s(a)$

$$x[n_1, n_2 + 1] - x[n_1, n_2] =$$
$$v_{\leftrightarrow,2}[n_1, n_2] + (v_{\updownarrow,2}[n_1, n_2] + v_{\updownarrow,2}[n_1, n_2 + 1] +$$
$$v_{\updownarrow,2}[n_1 - 1, n_2] + v_{\updownarrow,2}[n_1 - 1, n_2 + 1])/4 +$$
$$(v_{\bullet,2}[n_1, n_2] + v_{\bullet,2}[n_1, n_2 + 1])/2$$



Proposed TV


x
image
v
gradient field on
a twice finer grid
Sv = Dx
linear consistency (integration)

Proposed TV

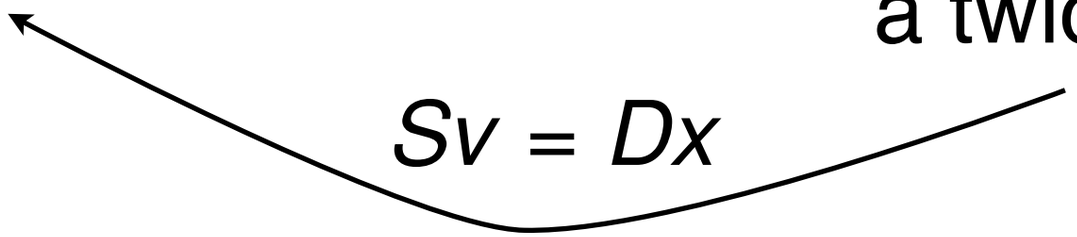
$$\mathrm{TV}_p(x) := \min_v \|v\|_{1,1,2} \quad \text{s.t.} \quad Sv = Dx$$

x

image

v

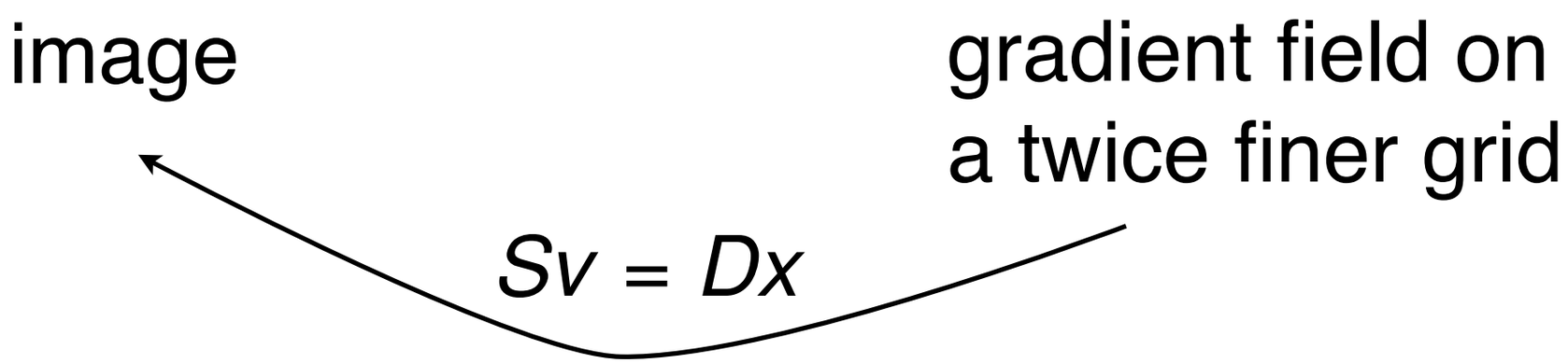
gradient field on a twice finer grid



linear consistency (integration)

Proposed TV

$$\mathrm{TV}_p(x) := \min_v \|v\|_{1,1,2} \quad \text{s.t.} \quad Sv = Dx$$



Note: the mapping $x \mapsto \arg\min_v \|v\|_{1,1,2}$ s.t. $Sv = Dx$ is nonlinear

Proposed TV

$$\mathrm{TV}_p(x) := \min_v \|v\|_{1,1,2} \quad \text{s.t.} \quad Sv = Dx$$

x
image

v
gradient field on a twice finer grid

$Sv = Dx$
linear consistency (integration)


inverse problem philosophy
(think of image enlargement)

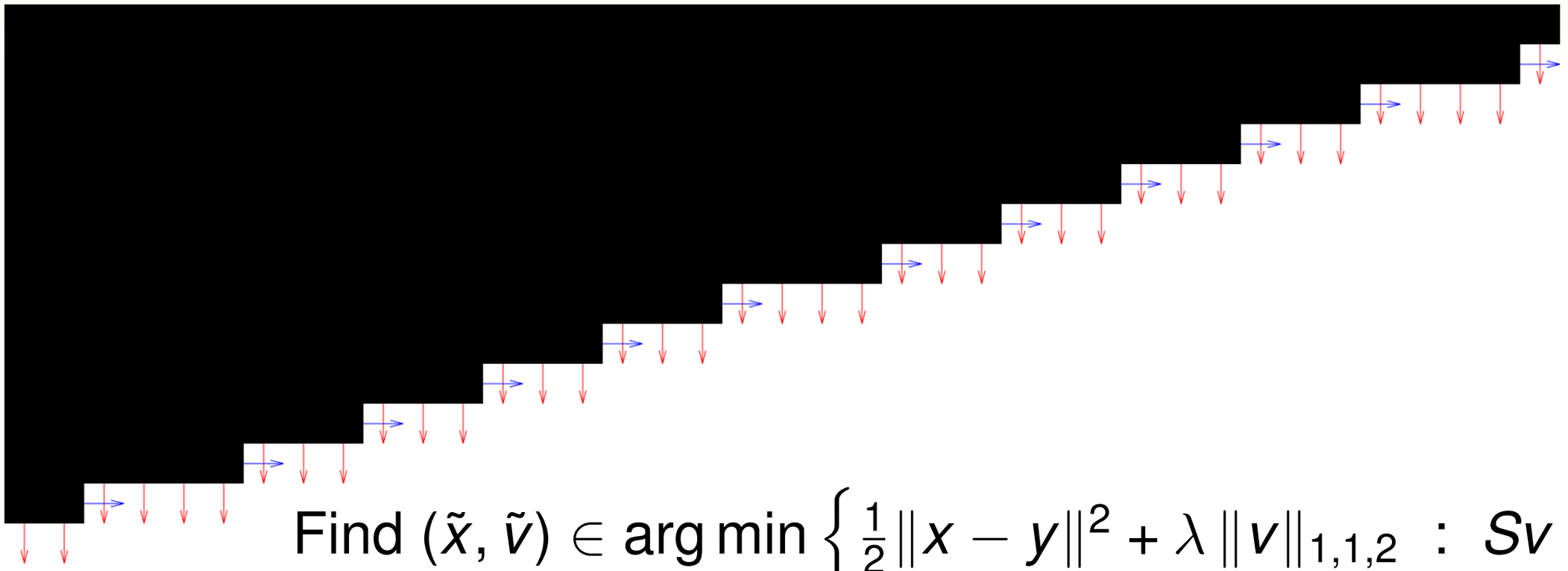
TV minimization

$$\text{Find } \tilde{x} \in \underset{x \in \mathbb{R}^{N_1 \times N_2}}{\arg\min} \left\{ F(x) + \lambda \, \mathrm{TV}_p(x) \right\}$$

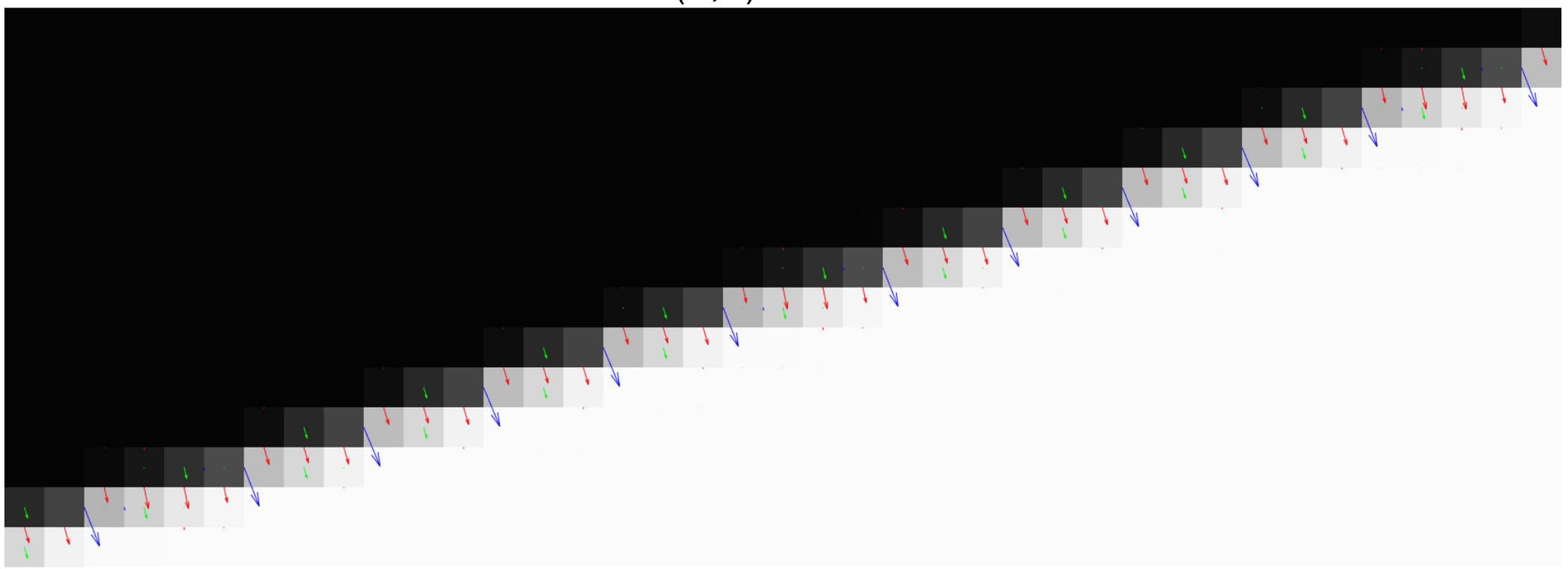
$$\equiv$$

$$\text{Find } (\tilde{x}, \tilde{v}) \in \underset{x \in \mathbb{R}^{N_1 \times N_2}, v \in (\mathbb{R}^2)^{3 \times N_1 \times N_2}}{\arg\min} \left\{ F(x) + \lambda \, \|v\|_{1,1,2} \ : \ Sv = Dx \right\}$$

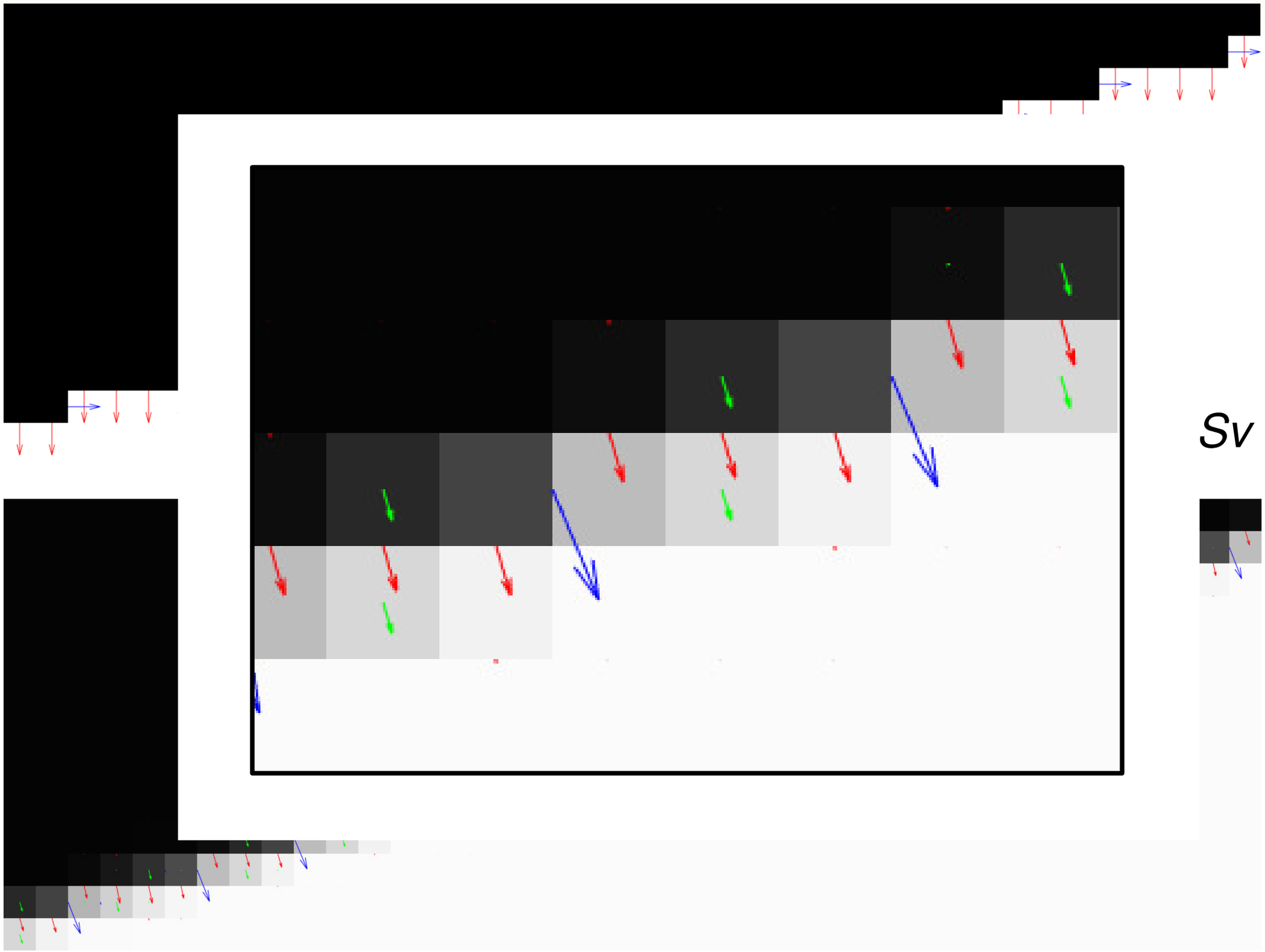
Smoothing an edge



$$\text{Find } (\tilde{x}, \tilde{v}) \in \underset{(x,v)}{\arg\min} \left\{ \tfrac{1}{2} \|x - y\|^2 + \lambda \|v\|_{1,1,2} \ : \ Sv = Dx \right\}$$

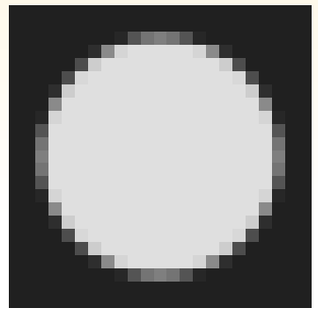


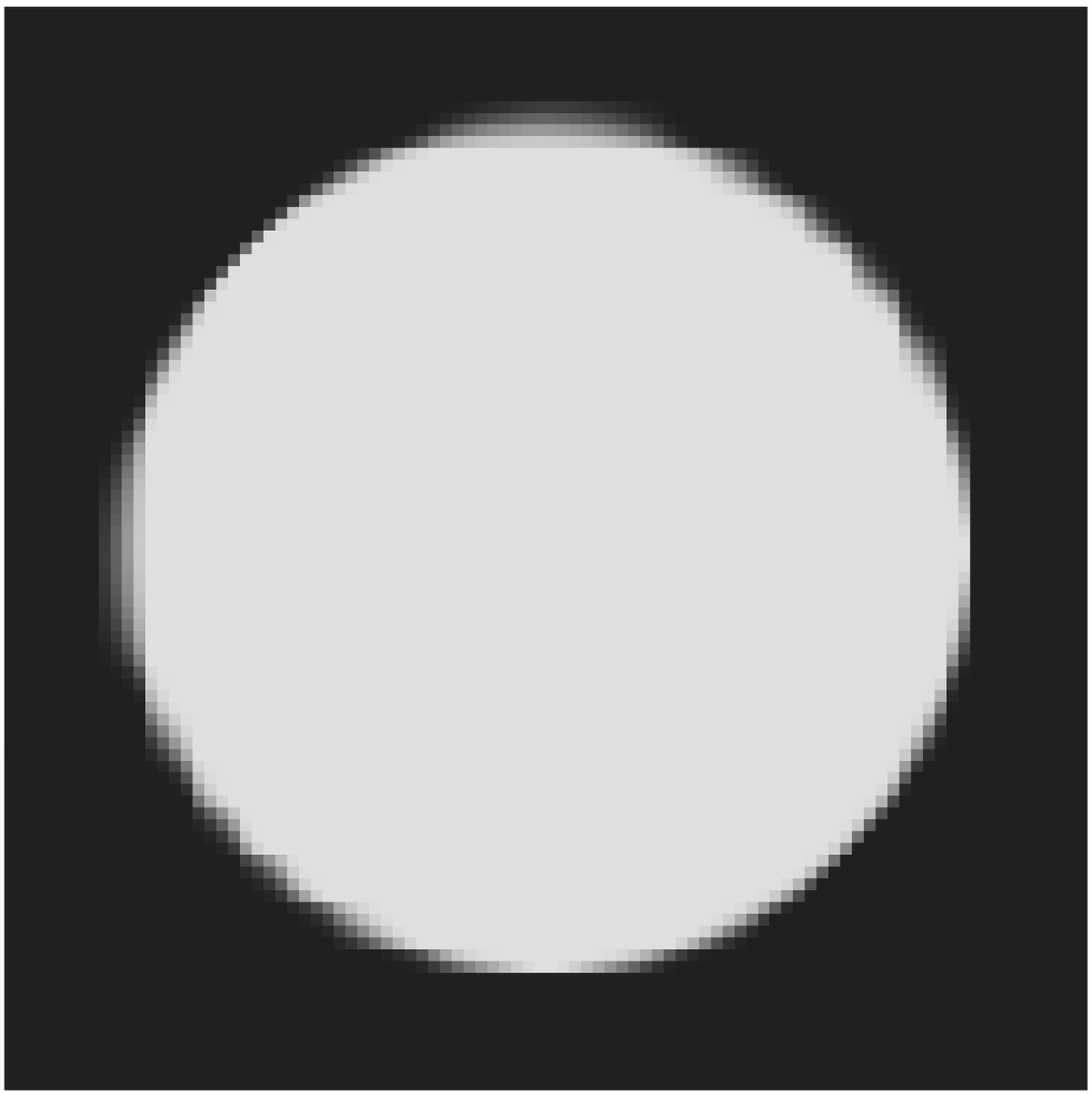
Smoothing an edge



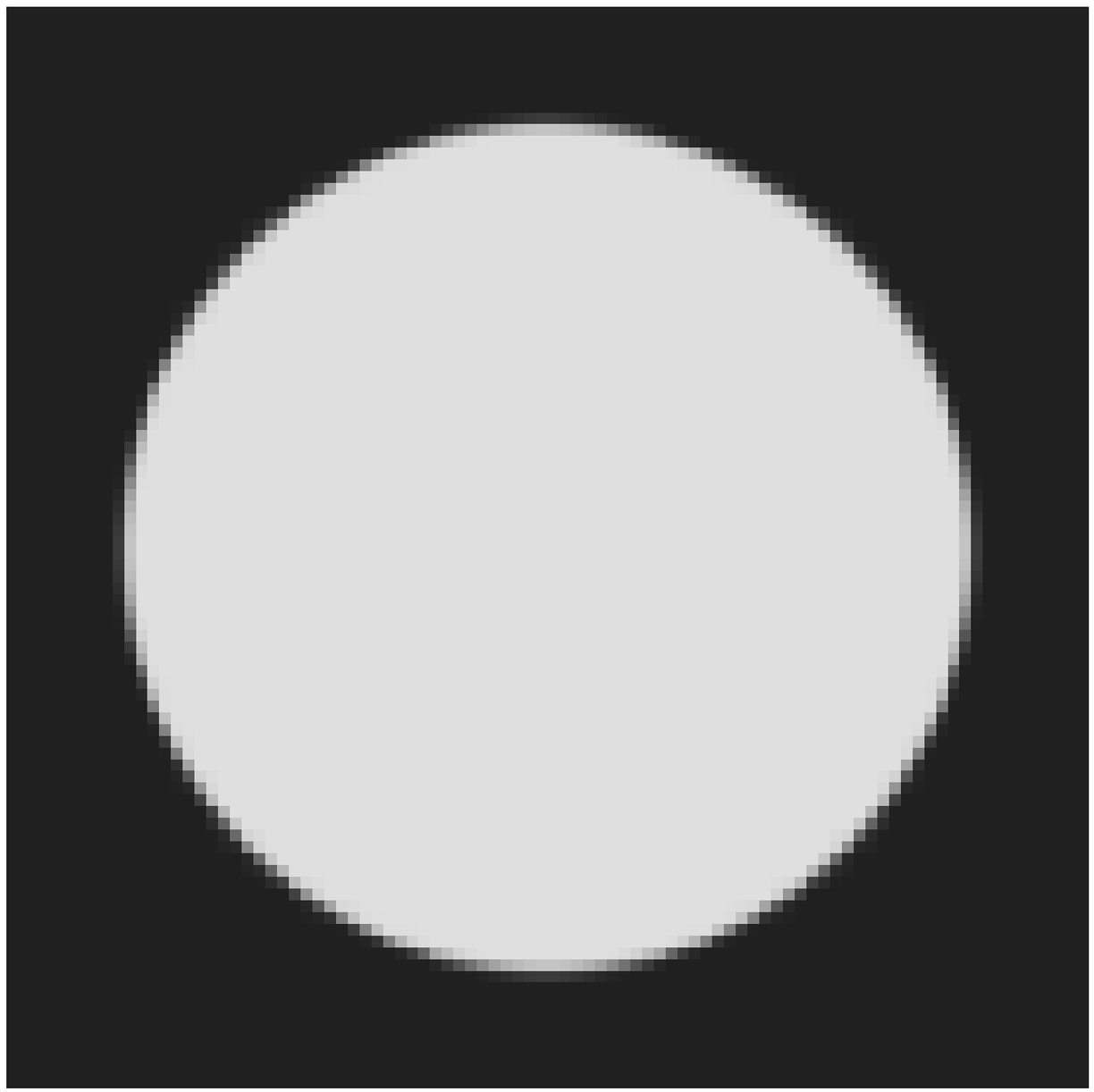
$$Sv = Dx$$

Enlargement





isotropic TV



proposed TV

Summary

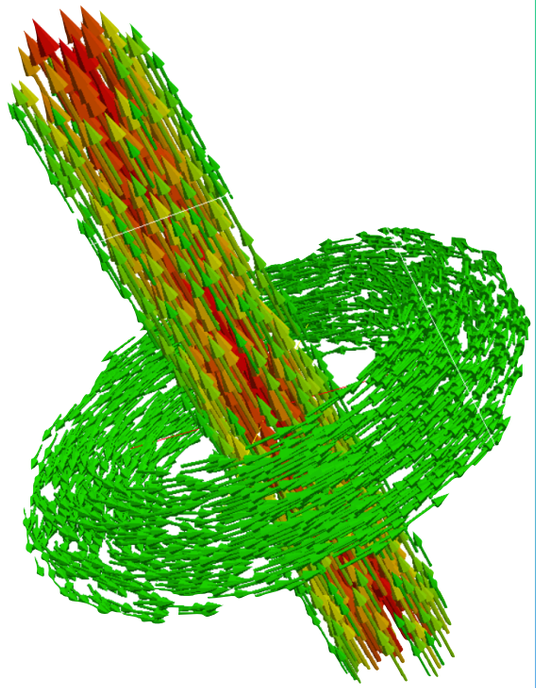
A thorough definition of the gradient field of an image and of the discrete TV

Summary

A thorough definition of the gradient field of an image and of the discrete TV

Perspectives

- extension to a discrete calculus framework
- applications to PDEs and simulation
- application to volume rendering

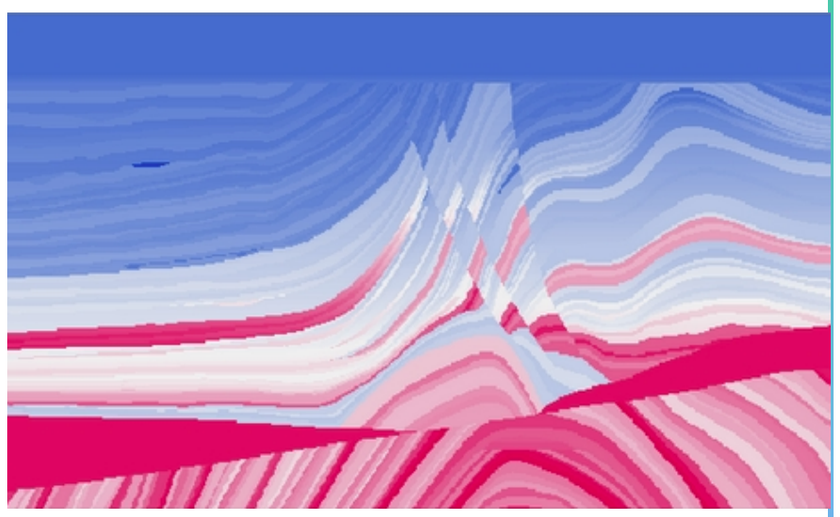


Summary

A thorough definition of the gradient field of an image and of the discrete TV

Perspectives

- extension to a discrete calculus framework
- applications to PDEs and simulation
- application to volume rendering



Summary

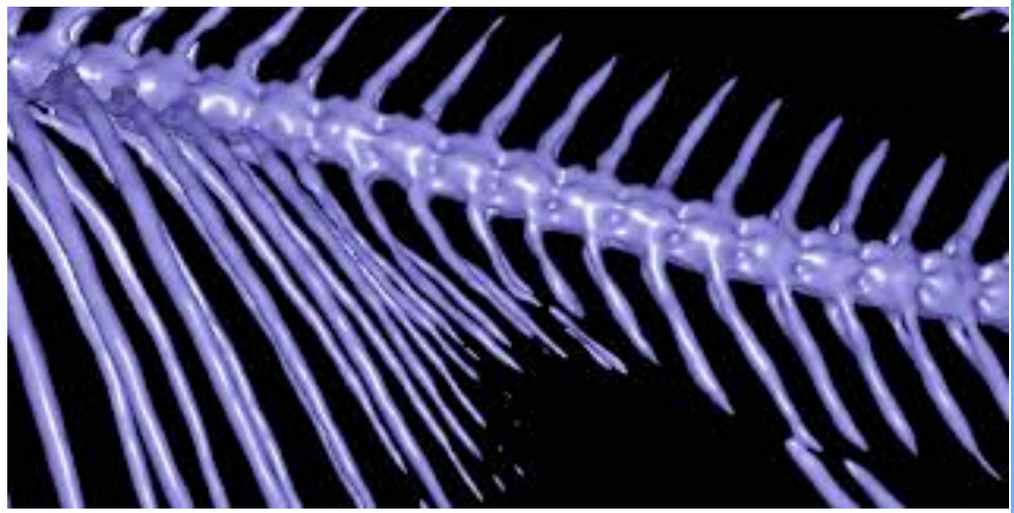
A thorough definition of the gradient field of an image and of the discrete TV

Perspectives

- extension to a discrete calculus framework
- applications to PDEs and simulation
- application to volume rendering

U. R. Alim, T. Möller, and L. C., *IEEE Trans. Visualization and Computer Graphics, 2010*

[Csébfalvi and Domonkos, 2010]



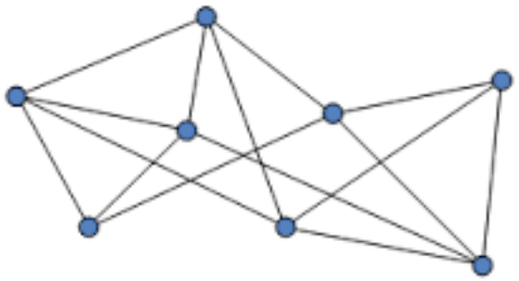
2. Convex optimization

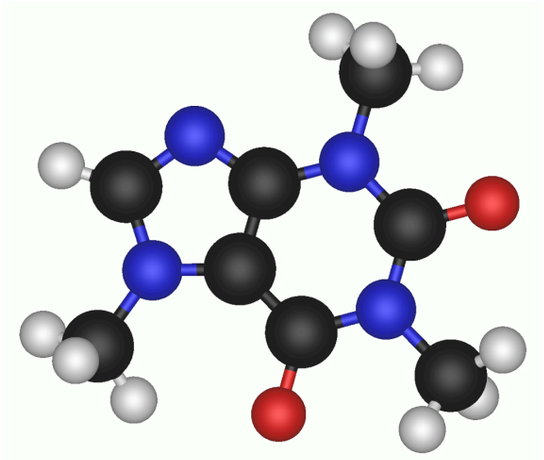
L. Condat, “A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms,” *Journal of Optimization Theory and Applications*, 2013

Setting

We place ourselves in a real Hilbert space $\mathcal{X}$

☞ We look for an object $\tilde{x} \in \mathcal{X}$, for instance,



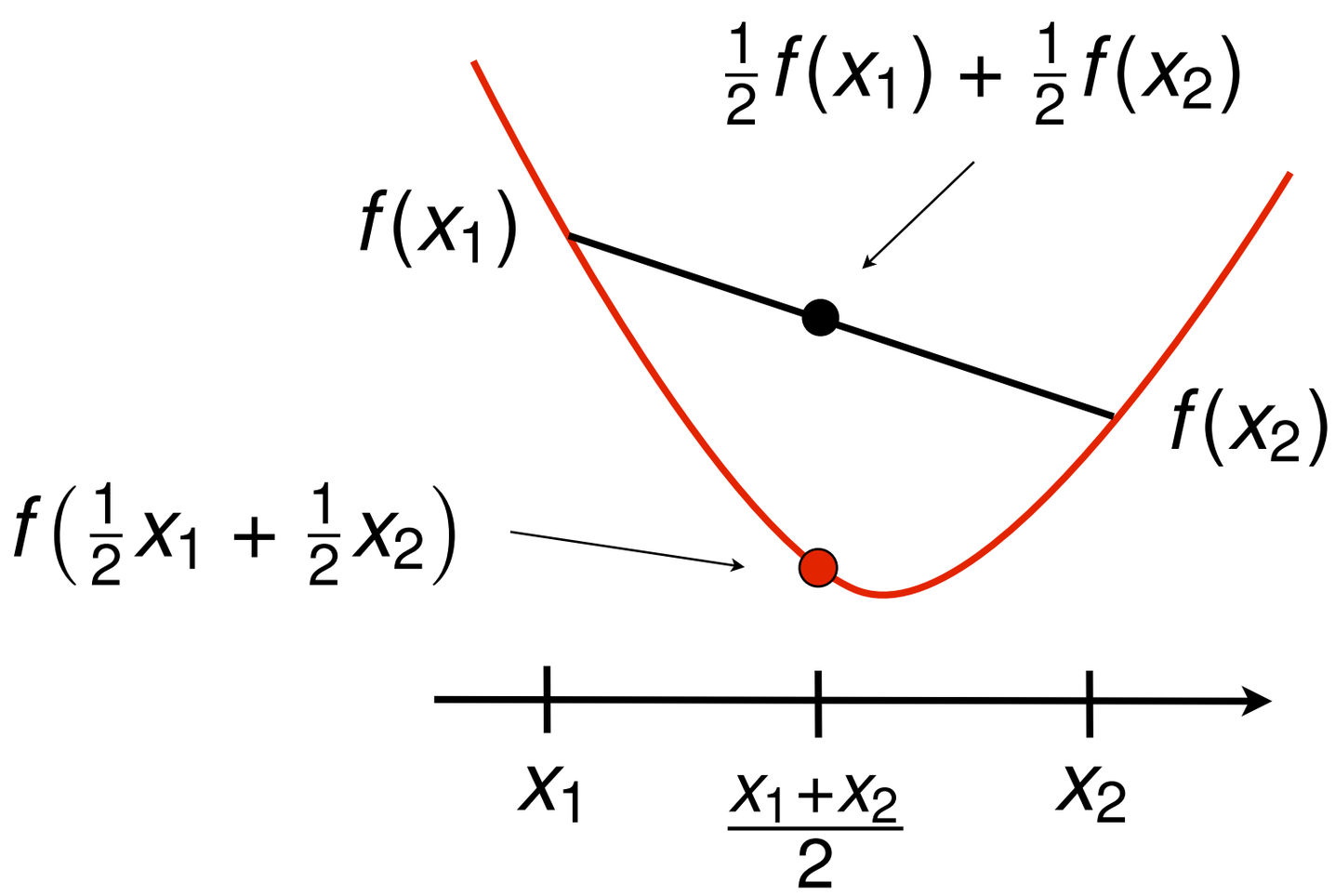


Goal

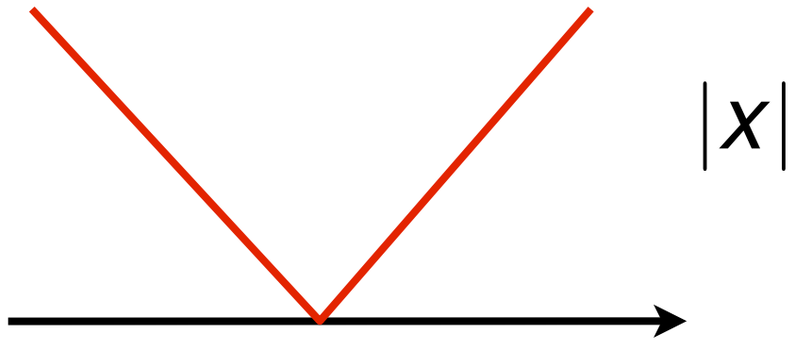
Find $\tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min}\, \Psi(x)$

for a convex, lower semicontinuous,
(cost) function $\Psi : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$

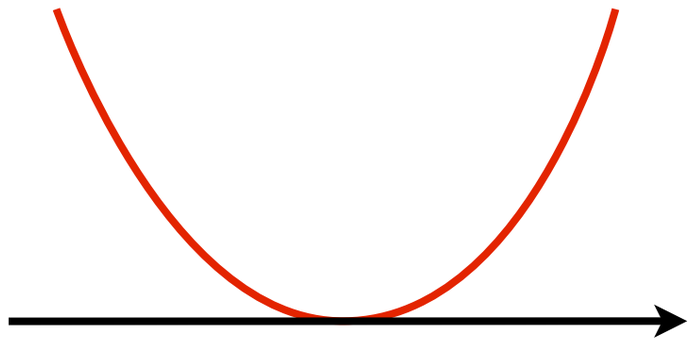
Convex functions



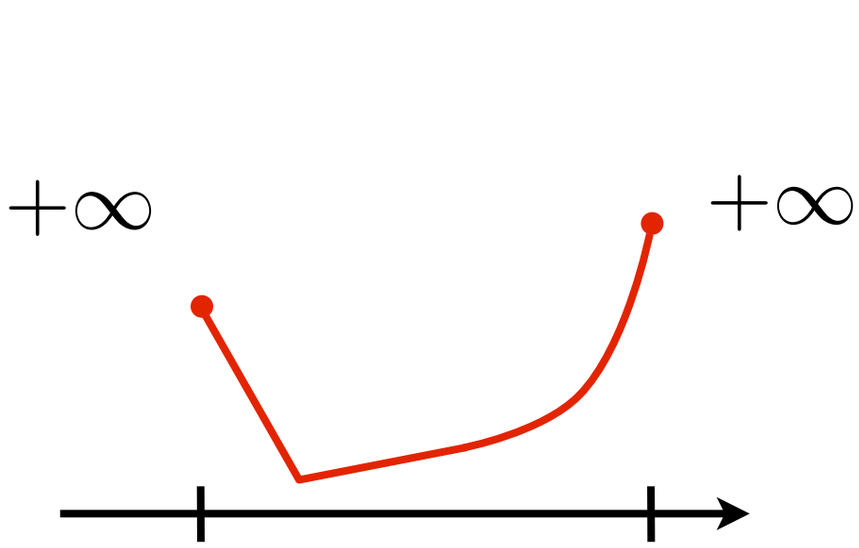
Convex functions

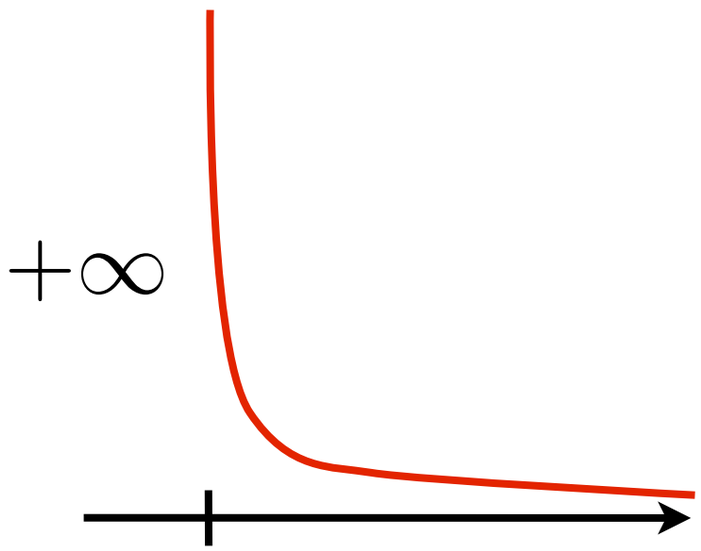


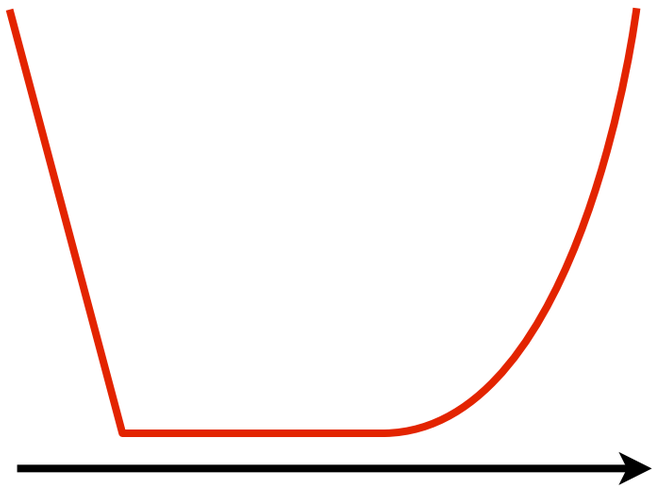
$|x|$



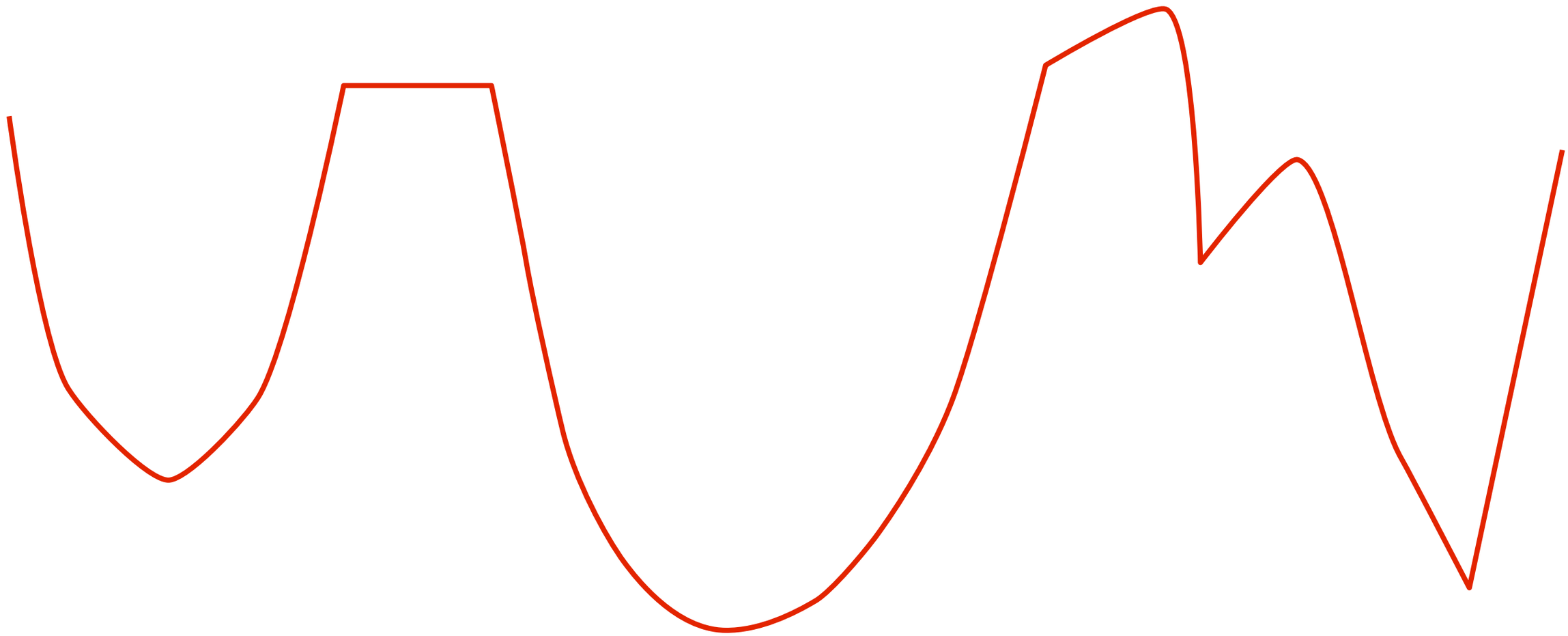
x^2







Convex optimization



A nonconvex function

Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min} \left\{ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x) \right\}$$

with

- convex functions g_m
- linear operators L_m

Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min} \left\{ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x) \right\}$$

We want full splitting, with individual activation of L_m, L_m^*, the gradient or proximity operator of g_m.

Goal

$$\text{Find } \tilde{x} \in \arg\min_{x \in \mathcal{X}} \left\{ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x) \right\}$$

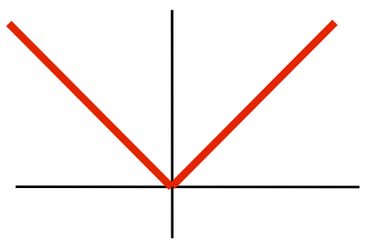
We want full splitting, with individual activation of L_m, L_m^*, the gradient or proximity operator of g_m.

- no implicit operation (inner loop or linear system to solve)
- only fast operations in $O(N)$ or $O(N \log N)$, with $N = \dim$.

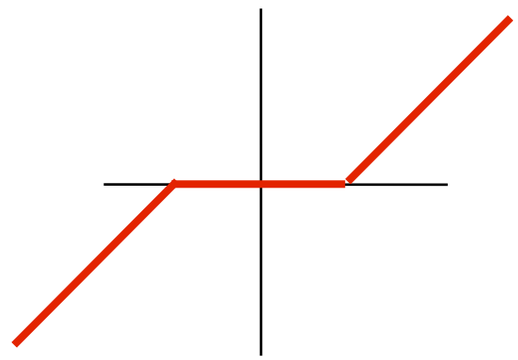
typically, $N \sim 10^6$–10^9

The proximity operator

$$\text{prox}_f : \mathcal{X} \to \mathcal{X} : x \mapsto \underset{x' \in \mathcal{X}}{\arg\min}\ f(x') + \frac{1}{2}\|x - x'\|^2$$



$f(x) = |x|$



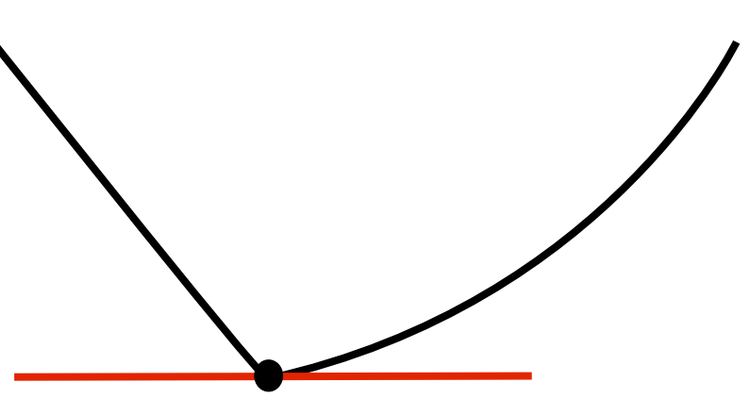
$$\text{prox}_f(x) = \text{sgn}(x)\max(|x| - 1, 0)$$

Goal

$$\text{Find} \quad \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min} \left\{ f(x) + g(Lx) + h(x) \right\}$$

Optimality conditions

$$\text{Find} \quad \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min} \left\{ f(x) + g(Lx) + h(x) \right\}$$



Optimality conditions

$$\text{Find} \quad \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min} \left\{ f(x) + g(Lx) + h(x) \right\}$$

$$\equiv$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial f(\tilde{x}) + L^* \tilde{u} \\ -L\tilde{x} + (\partial g)^{-1} \tilde{u} \end{pmatrix} + \begin{pmatrix} \nabla h(\tilde{x}) \\ 0 \end{pmatrix}$$

Proposed algorithm

$$\text{Find} \quad \tilde{x} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\{ f(x) + g(Lx) + h(x) \right\}$$

$$\equiv$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial f(\tilde{x}) + L^* \tilde{u} \\ -L\tilde{x} + (\partial g)^{-1} \tilde{u} \end{pmatrix} + \begin{pmatrix} \nabla h(\tilde{x}) \\ 0 \end{pmatrix}$$

☞ forward-backward iteration
in the primal-dual product space

Proposed algorithm

$$\text{Find} \quad \tilde{x} \in \arg\min_{x \in \mathcal{X}} \left\{ f(x) + g(Lx) + h(x) \right\}$$

$$\equiv$$

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial f(\tilde{x}) + L^* \tilde{u} \\ -L\tilde{x} + (\partial g)^{-1} \tilde{u} \end{pmatrix} + \begin{pmatrix} \nabla h(\tilde{x}) \\ 0 \end{pmatrix}$$

☞

$$\left\lfloor \begin{array}{l} x^{(i+1)} := \text{prox}_{\tau f}\left(x^{(i)} - \tau \nabla h(x^{(i)}) - \tau L^* u^{(i)} \right) \\ u^{(i+1)} := \text{prox}_{\sigma g^*}\left(u^{(i)} + \sigma L (2x^{(i+1)} - x^{(i)}) \right) \end{array} \right.$$

Proposed algorithm

$$\text{Find } \tilde{x} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\{ f(x) + \sum_{m=1}^{M} g_m(L_m x) + h(x) \right\}$$

Proposed algorithm. Choose the parameters $\tau > 0$, $\sigma > 0$, and the initial estimates $x^{(0)}$, $u_1^{(0)}, \ldots, u_M^{(0)}$.

Then iterate, for $i = 0, 1, \ldots$

$$x^{(i+1)} := \text{prox}_{\tau f}\left(x^{(i)} - \tau \nabla h(x^{(i)}) - \tau \sum_{m=1}^{M} L_m^* u_m^{(i)}\right)$$

For $m = 1, \ldots, M$,

$$u_m^{(i+1)} := \text{prox}_{\sigma g_m^*}\left(u_m^{(i)} + \sigma L_m (2x^{(i+1)} - x^{(i)})\right)$$

Proposed algorithm

$$\text{Find } \tilde{x} \in \arg\min_{x \in \mathcal{X}} \left\{ f(x) + \sum_{m=1}^{M} g_m(L_m x) + h(x) \right\}$$

Proposed algorithm. Choose the parameters $\tau > 0$, $\sigma > 0$, and the initial estimates $x^{(0)}$, $u_1^{(0)}, \ldots, u_M^{(0)}$.
Then iterate, for $i = 0, 1, \ldots$

$$x^{(i+1)} := \text{prox}_{\tau f}\left(x^{(i)} - \tau \nabla h(x^{(i)}) - \tau \sum_{m=1}^{M} L_m^* u_m^{(i)}\right)$$

For $m = 1, \ldots, M$,

$$u_m^{(i+1)} := \text{prox}_{\sigma g_m^*}\left(u_m^{(i)} + \sigma L_m (2x^{(i+1)} - x^{(i)})\right)$$

Convergence if $\frac{1}{\tau} - \sigma \left\| \sum_{m=1}^{M} L_m^* L_m \right\| \geq \frac{\beta}{2}$

Particular cases

- $g \circ L = 0 \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + h(x) \right\}$

☞ **forward–backward** algorithm

- $h = 0 \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + g(Lx) \right\}$

☞ **Chambolle–Pock** algorithm

- $h = 0$ and $L = \mathrm{Id} \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + g(x) \right\}$

☞ **Douglas–Rachford** ($\equiv$ **ADMM**) algorithm

Low-level routines

- L. Condat, “A direct algorithm for 1D total variation denoising,” *IEEE Signal Proc. Letters*, 2013.

- L. Condat, “Fast projection onto the simplex and the l1 ball,” *Mathematical Programming*, 2016.

- N. Pustelnik and L. Condat, “Proximity operator of a sum of functions; application to depth map estimation,” *IEEE Signal Processing Letters*, 2017.

- Code for more-or-less classical proxs. on my webpage

Future work

- New fixed-point iterations, e.g. with momentum
- Other low-level routines, e.g. projection of a matrix on a $\ell_{1,\infty}$ norm ball
- Distributed / parallel optimization. GPUs

  F. Iutzeler and L. C., “Distributed projection on the simplex and l1 ball via ADMM and gossip,” *IEEE Signal Process. Letters*, 2018

slides 45-51

Conclusion

Higher-dimensional formulations are often the key

- (image, gradient) pair
- (primal, dual) variables
- joint (spatial, spectral) information
- vectors mapped to Toeplitz matrices

...

Thank you!