Learning better discretizations for singular variational problems

Antonin Chambolle

CEREMADE, CNRS, Univ. Paris-Dauphine / PSL, France

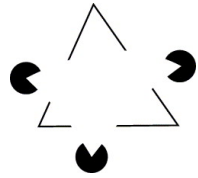
(joint with C. Caillaud (CMAP, Palaiseau), L. Kreutz (Münster), T. Pock (Graz))

Distinguished lectures for HKSIAM and Hong Kong Universities, 24th April 2021

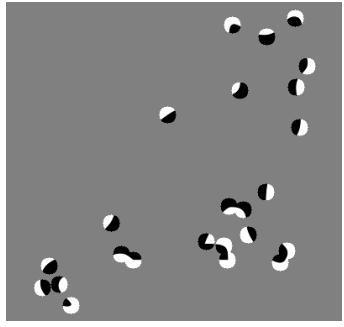
Outline:

- Energies for variational image reconstruction, singularities;
- Example: total variation (TV);
- Sharp isotropic TV via homogenization;
- Sharp isotropic TV: learning of a dual approximation.

A distant goal (but getting closer each year...)

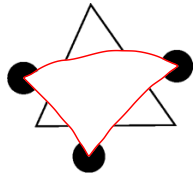


- "Inpainting" with "Elastica"



What is this? (RHS data from J. Weickert)

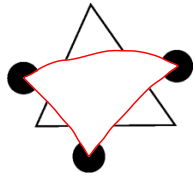
A distant goal (but getting closer each year...)



- "Inpainting" with "Elastica"

Inpainted with convexified "Elastica" [Ch, Pock, Num. Math. 19]

A distant goal (but getting closer each year...)



- "Inpainting" with "Elastica"

Inpainted with convexified "Elastica" [Ch, Pock, Num. Math. 19]

- Discretization is poor (3D lifting, boundary $\leftrightarrow$ 1-current)

A simpler problem

- "Total variation inpainting" (not a good idea)

TV Inpainting

A simpler problem

- "Total variation inpainting" (not a good idea)

TV Inpainting

- A standard discretization does not even work to properly recover a discontinuity...

Starting point

Improve, or possible "learn", discrete surface energies / total variations so that

- They are faithful, possibly precise approximations of the continuous T.V.;
- They behave "well" at the discrete level (isotropy, sharpness...)

The total variation

In this talk $TV(u) = \int |Du|$ is the "total variation" of an image u (that is, a function defined on a 2-dimensional domain). It is defined:

- as $\int |\nabla u(x)| dx = \|\nabla u\|_{L^1}$ if ∇u exists and is integrable;
- by duality for more general u's (formula comes later);
- it is well defined for *non-continuous* functions, for instance $\int |D\chi_E|$ is the *perimeter* of the set E;
- in general, minimizers (with other terms) can have discontinuities;
- in practice, TV is replaced by more or less good discrete approximations and then optimized.

Issue in this talk: build **precise** discrete approximations of the TV for discontinuous functions...

Related works

- Non-standard finite differences for anisotropic diffusion [Weickert, Welk, Wichert '13]
- Graph-based / MRFs / crystalline energies [Boykov, Kolmogorov '03], [Rother, Kolmogorov, Blake '04], [Boykov, Kolmogorov, Cremers, Delong '06], [AC '05], [Darbon-Sigelle '06] [Hochbaum '01];
- Upwind discretization [AC, Levine, Lucier '11];
- "Shannon TV" [Abergel, Moisan '17];
- Conforming P1 finite elements [Bartels '12];
- Non-conforming P1 (Crouzeix-Raviart) finite elements [AC, T. Pock 18];
- Duality based discretization using $H(\mathrm{div})$-conforming Raviart-Thomas (RT0) vector fields [Destuynder, Jaoua, Sellami 2007], [Herrmann, Herzog, Schmidt,Vidal, Wachsmuth '18], [Caillaud, AC '20];
- Approximate Raviart-Thomas [Hintermüller, Rautenberg, Hahn '14], [Condat '17].

Related works

- Non-standard finite differences for anisotropic diffusion [Weickert, Welk, Wichert '13]
- Graph-based / MRFs / crystalline energies [Boykov, Kolmogorov '03], [Rother, Kolmogorov, Blake '04], [Boykov, Kolmogorov, Cremers, Delong '06], [AC '05], [Darbon-Sigelle '06] [Hochbaum '01];
- Upwind discretization [AC, Levine, Lucier '11];
- "Shannon TV" [Abergel, Moisan '17];
- Conforming P1 finite elements [Bartels '12];
- Non-conforming P1 (Crouzeix-Raviart) finite elements [AC, T. Pock 18];
- Duality based discretization using $H(\mathrm{div})$-conforming Raviart-Thomas (RT0) vector fields [Destuynder, Jaoua, Sellami 2007], [Herrmann, Herzog, Schmidt,Vidal, Wachsmuth '18], [Caillaud, AC '20];
- Approximate Raviart-Thomas [Hintermüller, Rautenberg, Hahn '14], [Condat '17].

Here: Attempt to optimize graph-based or finite-differences/elements based methods by automatic learning.

0. Typical model:

Focus on problems of the form:

$$\min_{u=u^0\ \partial\Omega} \int_\Omega |Du| \quad \left(\text{or } + \int_\Omega |u-g|^2 dx \quad (ROF)\right)$$

where $u^0 \in \{0,1\}$, so that this is equivalent to finding sets E with lowest perimeter and boundary condition $\chi_E = u^0$. One expects to find (in general) sharp solutions $u \in \{0,1\}$ *a.e.*

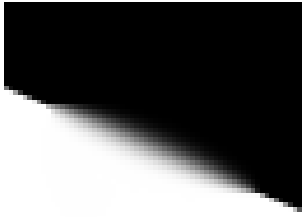
Discretize: One minimizes in practice a convex problem of the form

$$\min_{u_i=u_i^0, i\in I^0} F_h(u_i) : u \in \mathbb{R}^{N(h)}$$

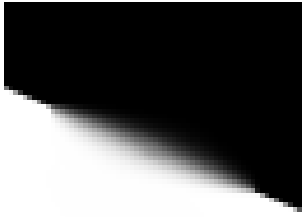
where $(u_i)_{i=1}^{N(h)}$ is supposed to be a discrete representation of u at scale $h > 0$, and F_h approximates the total variation in some sense.

Typical model:

In practice, depending on the form of F_h, one can expect more or less "nice" or "precise" results (sharp, isotropic, or not...)



"forward"



"Raviart-Thomas"

"Condat"

Typical model:

Here the discrete problem has usually the form of a convex-concave saddle-point problem:

$$\min_{u \in C_u} \sup_{w \in C_w} \langle w, Du \rangle \quad \rightsquigarrow \min_{u \in C_u} \|Du\|_*$$

for D some discrete derivative, and where C_u and C_w are convex sets. More regular versions include

$$\min_{u \in C_u} \sup_{w \in C_w} \langle w, Du \rangle + \frac{1}{2}\|u - g\|^2 \qquad (\text{“ROF”})$$

which is strongly convex wr u, or a “regularized” variant:

$$\min_{u \in C_u} \sup_{w \in C_w} \langle w, Du \rangle - \frac{\varepsilon}{2}\|w\|^2 + \frac{\varepsilon}{2}\|u\|^2$$

for $\varepsilon > 0$ a small parameter, which is strongly convex wr both u and w.

I. "discrete" discretizations: Graph-TV

One basic way to build "sharp" total variations is to consider purely discrete finite differences on a graph. The general form is as follows: Assume we are given a discrete $2D$ image $u_{i,j}$ on a square grid, we define a graph total variation as

$$\sum_{(i,j),(i',j')} \alpha_{(i,j),(i',j')}(u_{i',j'} - u_{i,j})^+$$

(here $x^+ = \max\{x, 0\}$). [*For $u \in \{0, 1\}$, this is nothing but a standard "cut" loss $\rightarrow$ max-flow algorithms.*]

Using such functions produces in practice quite anisotropic (crystalline) measures of the perimeters. This (visible) problem can be mitigated in two different ways. The most common is to increase the number of edges. For "fun" (or computational efficiency) let us consider an alternative approach (joint with L. Kreutz, Münster WWU).

Graph TV

We will try to build an "*isotropic*-ℓ_1" discretization. The simplest form would be:

$$\sum_{i,j} \alpha^+_{i+\frac{1}{2},j}(u_{i+1,j} - u_{i,j})^+ + \alpha^-_{i+\frac{1}{2},j}(u_{i,j} - u_{i+1,j})^+ \\ + \alpha^+_{i,j+\frac{1}{2}}(u_{i,j+1} - u_{i,j})^+ + \alpha^-_{i,j+\frac{1}{2}}(u_{i,j} - u_{i,j+1})^+$$

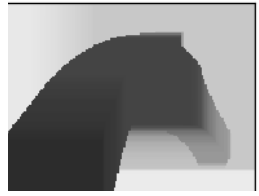
which involves only horizontal/vertical directions.

Graph TV

If all the α's are 1, this is known as an "ℓ_1" discretization of the total variation, which in a continuum limit would approximate the anisotropic functional $\int |\partial_1 u| + |\partial_2 u|$, and produces block artefacts. *(It measure the lengths only through vertical and horizontal projections like a New-York cab driver.)*

On the other hand,

- it is very easy and fast to optimize (graph cuts, or horizontal/vertical splitting...)
- one can show that it "always" produce sharp interfaces (related to *co-area formula / Lovasz' extension*)



Homogenization of graph TV

The isotropy can be improved by "homogenization". In practice, the idea is to use **periodic** oscillating weights $\alpha^{\pm}$ which produce, in the continuum limit, an "effective surface tension" ϕ ($\rightsquigarrow \int \phi(Du)$) given by an exact "cell formula", defined for $\nu \in \mathbb{R}^2$,

$$\begin{aligned}\phi(\nu) = \min_u \Big\{ \sum_{(i,j)\in Y} & \alpha^+_{i+\frac{1}{2},j}(u_{i+1,j}-u_{i,j})^+ + \alpha^-_{i+\frac{1}{2},j}(u_{i,j}-u_{i+1,j})^+ \\ & + \alpha^+_{i,j+\frac{1}{2}}(u_{i,j+1}-u_{i,j})^+ + \alpha^-_{i,j+\frac{1}{2}}(u_{i,j}-u_{i,j+1})^+ \ : \\ & u_{i,j} - \nu \cdot \begin{pmatrix} i \\ j \end{pmatrix} \ Y\text{-periodic} \Big\}\end{aligned}$$

where here Y is a periodicity cell of the form $\{1,\dots,n\}\times\{1,\dots,m\}$. (Typically, $m = n = 2, 3, 4....$) *(u is a periodic perturbation of the affine function $x \mapsto \nu \cdot x$.)*

Homogenization

... and one would be interested in solving:

$$\min_{(\alpha)} \mathcal{L}(\alpha) := \frac{1}{2} \sum_{i=1}^{k} |\phi(\nu_i) - 1|^2$$

where the "loss" $\mathcal{L}$ depends on α through the dependence of $\phi(\cdot)$ on α and ν_i are a set of given directions.
So one needs to estimate $\nabla_{(\alpha)}\phi(\nu_i)$, for each direction ν_i.

Derivative of the energy

In our case the minimal energy $\phi(\nu)$ can be found by solving a saddle-point problem:

$$\phi(\nu) = \min_{u \in C_i(\nu)} \sup_{w \in C_w} \langle D(\alpha)u, w \rangle$$

Derivative of the energy

In our case the minimal energy $\phi(\nu)$ can be found by solving a saddle-point problem:

$$\phi(\nu) = \min_{u \in C_i(\nu)} \sup_{w \in C_w} \langle D(\alpha)u, w \rangle - \frac{\varepsilon}{2}\|w\|^2 + \frac{\varepsilon}{2}\left\|u - \binom{i}{j} \cdot \nu\right\|^2$$

which we regularize in order to have a unique solution $(u(D), w(D))$ for a given discrete derivative operator D.

Derivative of the energy

Thanks to the regularization one easily sees that

- $D \mapsto (u(D), w(D))$ is continuous and
- $D \mapsto \phi(\nu) =: \mathcal{E}_\nu(D)$ is $C^{1,1}$.

Indeed:

$$\sup_{w \in C_w} \langle Du, w \rangle - \frac{\varepsilon}{2}\|w\|^2$$

- is convex with $(1/\varepsilon)$-Lipschitz gradient with respect to Du
- is convex with (C/ε)-Lipschitz gradient with respect to D in a neighborhood of D, for $C > \|u(D)\|^2$.
- so its $\inf_u$ has Hessian bounded from above.

Derivative of the energy

Thanks to the regularization one easily sees that

- $D \mapsto (u(D), w(D))$ is continuous and
- $D \mapsto \phi(\nu) =: \mathcal{E}_\nu(D)$ is $C^{1,1}$.

Indeed:

$$\sup_{w \in C_w} \langle Du, w \rangle - \frac{\varepsilon}{2}\|w\|^2$$

- is convex with $(1/\varepsilon)$-Lipschitz gradient with respect to Du
- is convex with (C/ε)-Lipschitz gradient with respect to D in a neighborhood of D, for $C > \|u(D)\|^2$.
- so its $\inf_u$ has Hessian bounded from above.
- symetrically (taking first $\inf_u$ then $\sup_w$) one gets a bound from below.

Derivative of the energy

Then, computing the differential is quite standard (one can for instance estimate $\mathcal{E}_\nu(D + tL)$, t small, from above and below using the optimal values u_t, w_t, and pass to the limit...) and one finds

$$\nabla_D \mathcal{E}_\nu(D) = w(D) \otimes u(D)$$

So here one just needs to solve (with some precision) the saddle point to evaluate the derivative from the optimal solutions (u, w). Then one can implement a gradient descent and optimize the main criterion $\mathcal{L}(\alpha)$.

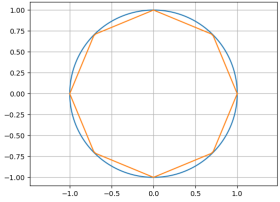
Derivative of the energy

Then, computing the differential is quite standard (one can for instance estimate $\mathcal{E}_\nu(D + tL)$, t small, from above and below using the optimal values u_t, w_t, and pass to the limit...) and one finds

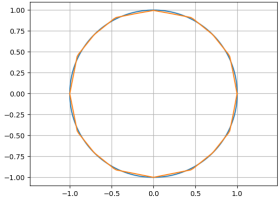
$$\nabla_D \mathcal{E}_\nu(D) = w(D) \otimes u(D)$$

So here one just needs to solve (with some precision) the saddle point to evaluate the derivative from the optimal solutions (u, w). Then one can implement a gradient descent and optimize the main criterion $\mathcal{L}(\alpha)$.

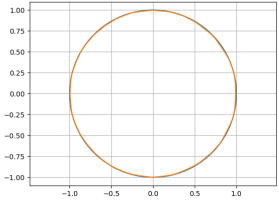
Application / Results



2×2 periodicity cell

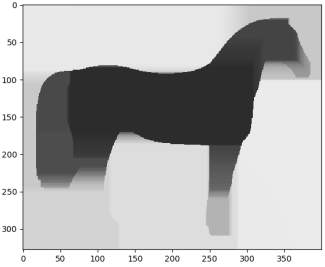


4×4 cell

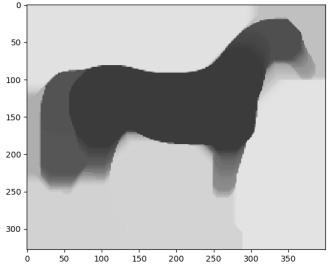


8×8

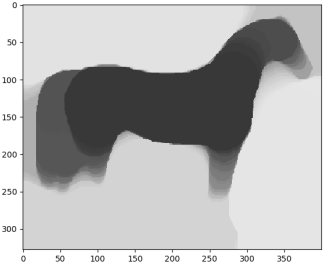
Application / Results



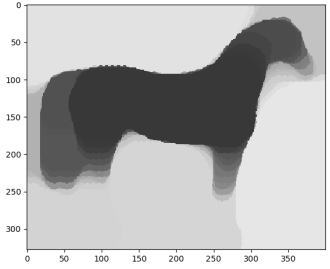
ℓ_1-TV



2×2

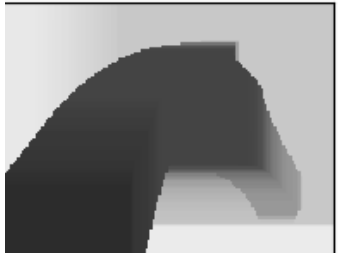


4×4

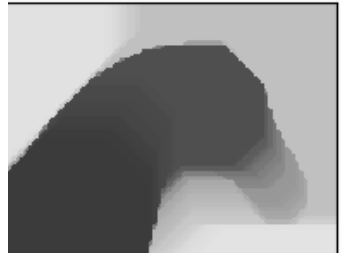


8×8

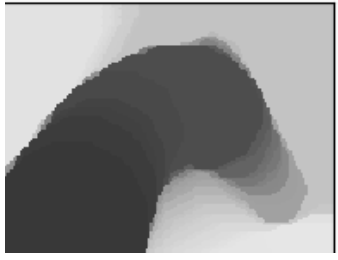
Application / Results



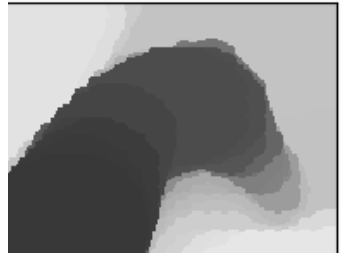
ℓ_1-TV



2×2



4×4

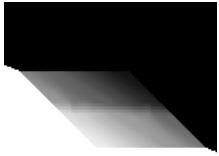


8×8

Application / Results: inpainting

Boundary datum

ℓ_1-TV



2×2

4×4

6×6

8×8

It does NOT work at all!! (main reason: *non-uniqueness* $\rightarrow$ threshold)

II. "Continuous" discretizations

(More in the spirit of finite differences or finite element discretizations.)

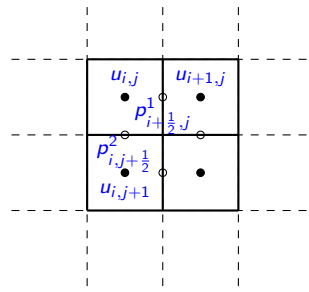
A quite general approach consists in discretizing the *dual* definition of the Total Variation. One has given a domain $\Omega \subset \mathbb{R}^d$ and $u \in L^1(\Omega)$:

$$TV(u;\Omega) = \int_\Omega |Du| = \sup \left\{ -\int_\Omega u \operatorname{div} \phi dx : \phi \in C_c^\infty(\Omega;\mathbb{R}^d), \|\phi(x)\| \leq 1 \ \forall x \right\}$$

Indeed: $-\int u \operatorname{div} \phi \, dx = \int \phi \cdot Du$ and the supremum gives back $\int |Du|$. *(This makes sense as soon as the* sup *above is finite.)*

$\rightsquigarrow$ Discretize u, ϕ, and the norm constraint.

General discrete model



Our general discrete total variations are given by:

$$TV(u) := \sup\{\langle \boldsymbol{p}, \boldsymbol{D}u\rangle_Y : \|\boldsymbol{F}\boldsymbol{p}\|_{Z^*} \leq 1\} = \min_{q:\boldsymbol{F}^* q = \boldsymbol{D}u} \|q\|_Z$$

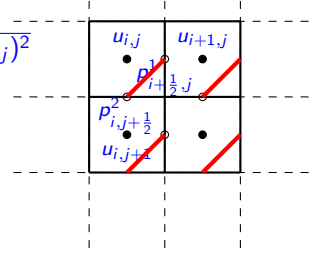
where $\boldsymbol{p} = (p^1, p^2)$ are the dual variables and $\boldsymbol{F} = (\boldsymbol{F}^1, ..., \boldsymbol{F}^L)$ are interpolation kernels defined by convolutions:

$$(\boldsymbol{F}^l \boldsymbol{p})_{i,j} = \begin{pmatrix} (F^{l,1}p^1)_{i,j} \\ (F^{l,2}p^2)_{i,j} \end{pmatrix} = \begin{pmatrix} \sum_{m,n=-\nu}^{\nu} \xi^l_{m,n} p^1_{i+\frac{1}{2}-m,j-n} \\ \sum_{m,n=-\nu}^{\nu} \eta^l_{m,n} p^2_{i-m,j+\frac{1}{2}-n} \end{pmatrix}$$

Here the discrete gradient is $\boldsymbol{D}u = (D^1 u, D^2 u)$, given by:

$$\begin{cases} (D^1 u)_{i+\frac{1}{2},j} = u_{i+1,j} - u_{i,j} & i = 1, \ldots, M-1,\ j = 1, \ldots, N, \\ (D^2 u)_{i,j+\frac{1}{2}} = u_{i,j+1} - u_{i,j} & i = 1, \ldots, M,\ j = 1, \ldots, N-1. \end{cases}$$

Example: Forward differences $\sum_{i,j} \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2}$



- Interpolation kernels (Nearest neighbor interpolation):

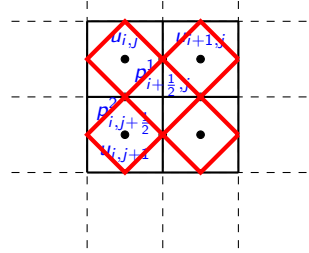
$$(Fp)_{i,j} = \begin{pmatrix} p^1_{i+\frac{1}{2},j} \\ p^2_{i,j+\frac{1}{2}} \end{pmatrix}.$$

Interpolation kernels F

- The Z-norm is given by

$$\|z\|_Z = \sum_{i,j} \sqrt{(z^1_{i+\frac{1}{2},j})^2 + (z^2_{i,j+\frac{1}{2}})^2}, \text{ with dual } \|z\|_{Z^*} = \max_{i,j} \sqrt{(z^1_{i+\frac{1}{2},j})^2 + (z^2_{i,j+\frac{1}{2}})^2}$$

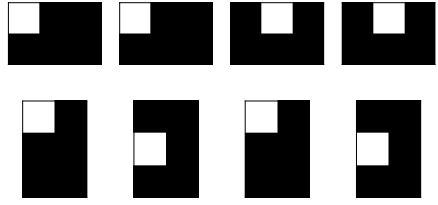
Example: Raviart-Thomas



- Interpolation kernels (Nearest neighbor interpolation):

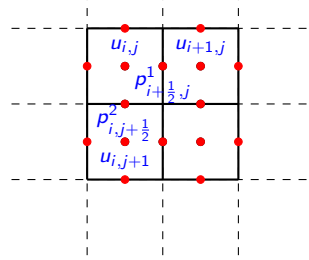
$$(F^1\boldsymbol{p})_{i-\frac{1}{2},j-\frac{1}{2}} = \begin{pmatrix} p^1_{i-\frac{1}{2},j} \\ p^2_{i,j-\frac{1}{2}} \end{pmatrix}, \quad (F^2\boldsymbol{p})_{i-\frac{1}{2},j+\frac{1}{2}} = \begin{pmatrix} p^1_{i-\frac{1}{2},j} \\ p^2_{i,j+\frac{1}{2}} \end{pmatrix},$$

$$(F^3\boldsymbol{p})_{i+\frac{1}{2},j-\frac{1}{2}} = \begin{pmatrix} p^1_{i+\frac{1}{2},j} \\ p^2_{i,j-\frac{1}{2}} \end{pmatrix}, \quad (F^4\boldsymbol{p})_{i+\frac{1}{2},j+\frac{1}{2}} = \begin{pmatrix} p^1_{i+\frac{1}{2},j} \\ p^2_{i,j+\frac{1}{2}} \end{pmatrix}.$$



- $\|(z^1, z^2, z^3, z^4)\|_Z := \sum_{i,j} |z^1_{i-\frac{1}{2},j-\frac{1}{2}}|_2 + |z^2_{i-\frac{1}{2},j+\frac{1}{2}}|_2 + |z^3_{i+\frac{1}{2},j-\frac{1}{2}}|_2 + |z^4_{i+\frac{1}{2},j+\frac{1}{2}}|_2$

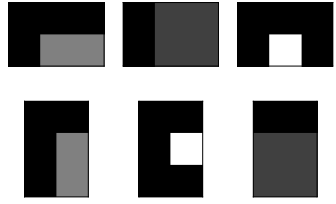
Example: Hintermüller et al/Condat's discretization



- Interpolation kernels (bilinear interpolation):

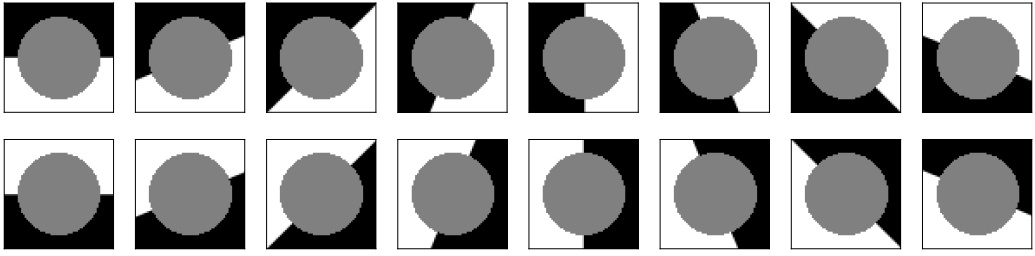
$$(F^1\boldsymbol{p})_{i,j} = \begin{pmatrix} \frac{p^1_{i-\frac{1}{2},j}+p^1_{i+\frac{1}{2},j}}{2} \\ \frac{p^2_{i,j-\frac{1}{2}}+p^2_{i,j+\frac{1}{2}}}{2} \end{pmatrix},$$

$$(F^2\boldsymbol{p})_{i+\frac{1}{2},j} = \begin{pmatrix} p^1_{i+\frac{1}{2},j} \\ \frac{p^2_{i,j-\frac{1}{2}}+p^2_{i,j+\frac{1}{2}}+p^2_{i+1,j-\frac{1}{2}}+p^2_{i+1,j+\frac{1}{2}}}{4} \end{pmatrix}, \quad (F^3\boldsymbol{p})_{i,j+\frac{1}{2}} = \begin{pmatrix} \frac{p^1_{i-\frac{1}{2},j}+p^1_{i+\frac{1}{2},j}+p^1_{i-\frac{1}{2},j+1}+p^1_{i+\frac{1}{2},j+1}}{4} \\ p^2_{i,j+\frac{1}{2}} \end{pmatrix}.$$



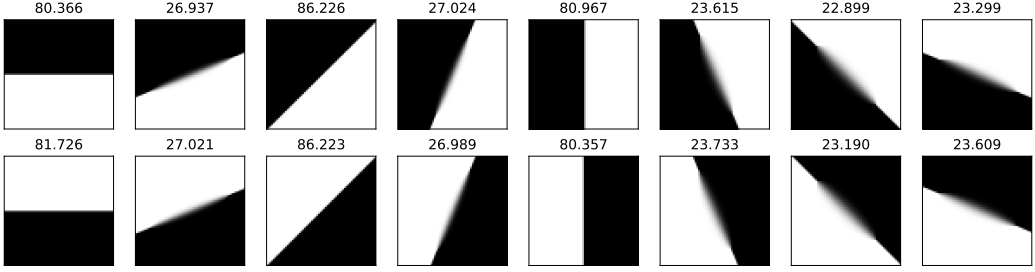
- $\|(\boldsymbol{z}^1,\boldsymbol{z}^2,\boldsymbol{z}^3)\|_Z := \sum_{i,j} |\boldsymbol{z}^1_{i,j}|_2 + |\boldsymbol{z}^2_{i+\frac{1}{2},j}|_2 + |\boldsymbol{z}^3_{i,j+\frac{1}{2}}|_2$

Comparison



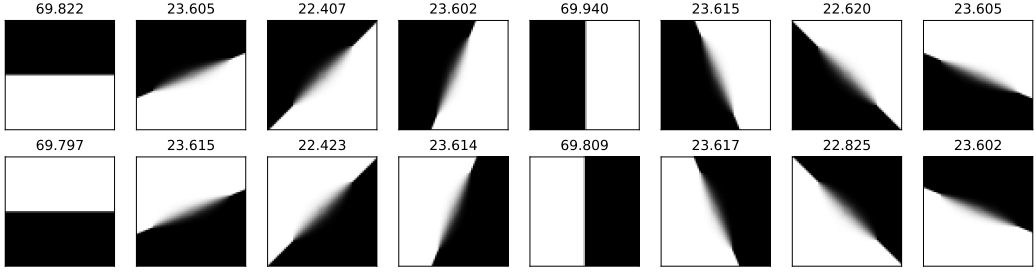
Input

Comparison



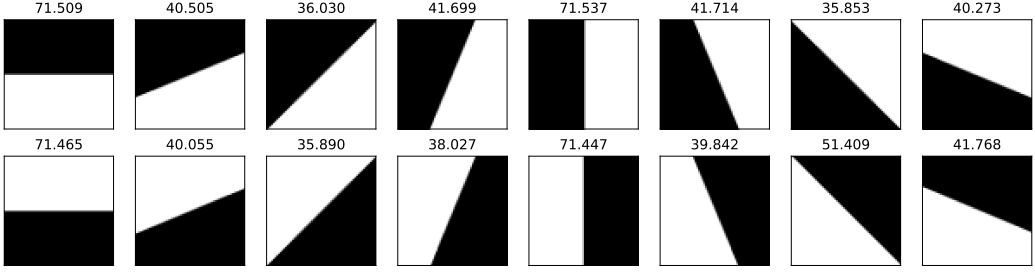
Forward differences

Comparison



Raviart-Thomas $\rightarrow$why???

Comparison



Condat

Theoretical results

1. For the "Raviart-Thomas" variant:
   - "(ROF) denoising problem": optimal error bound:

     Theorem. Assume that a dual field for $\bar{u}$ solving (ROF) is Lipschitz: then $\|\bar{u} - \bar{u}_h\|_{L^2} \leq C\sqrt{h}$ (for some constant essentially depending on $\|g\|_\infty$ and the Lipschitz constant of the dual field).
     Remark: the condition is probably not often satisfied. It is satisfied if $g = \chi_B$.

   - "Inpainting": assume $g_{\nu,a}(x) = \chi_{\{x \cdot \nu \geq a\}}$ is the characteristic of a half-plane (or -space), let V_h be the space of discrete piecewise constant functions at scale $h > 0$:

     Proposition. Let $g^h_{\nu,a}$ be an appropriate discretization of $g_{\nu,a}$ on $\partial\Omega$, at scale h. Then $u^h = \Pi_h g_{\nu,a}$ (the projection onto V_h) is such that

     $$TV_h(u^h) \leq TV_h(v^h) \quad \forall v^h \in V_h\,,\, v^h = g^h_{\nu,a} \text{ on } \partial\Omega$$

Theoretical results

Question: why is the inpainting result with RT not "perfect"? most probably: non uniqueness (*cf* Graph TV or Crouzeix-Raviart).

Theoretical results

2. Consistency.
Given a scale $h > 0$, define a family of discrete total variations for pixels of size $h \times h$:

$$TV_h(u) = \min \left\{ h^2 \|\boldsymbol{q}\|_{Z_h} : \boldsymbol{F}_h^* \boldsymbol{q} = \boldsymbol{D}_h u \right\} = \sup \left\{ h^2 \langle \boldsymbol{p}, \boldsymbol{D}_h u \rangle_{Y_h} : \|\boldsymbol{F}_h \boldsymbol{p}\|_Z^* \leq 1 \right\}$$

(with $\boldsymbol{D}_h u = (1/h)\boldsymbol{D}u$).

Theorem. Assume the weights of the convolutions defining $\boldsymbol{F}_h$ satisfy

$$\sum_{m,n} \xi_{m,n}^l = \sum_{m,n} \eta_{m,n}^l = 1$$

for all $h > 0$ (in addition, uniformly bounded and with bounded support).
Then TV_h "Γ-converges" to the total variation $TV(u)$ as $h \to 0$.

$\rightsquigarrow$ convergence of the discrete minimizers to solutions of the continuous problem.
So all the examples mentioned earlier "are", in some sense, consistent total variations.
Now, by tuning appropriately ξ, η, one can hope to "**learn**" new ones in order to solve best a given task.

Learning setting

We consider the class of total variation minimization problems (for various terms $G(u, g)$ corresponding to the "(ROF)" denoising or the inpainting problem):

$$\min_{\boldsymbol{D}u=\boldsymbol{F}^*\boldsymbol{q}} \lambda\|\boldsymbol{q}\|_Z + G(u, g),$$

with the saddle-point formulation

$$\min_{u,\boldsymbol{q}} \max_{\boldsymbol{p}} \langle \boldsymbol{D}u - \boldsymbol{F}^*\boldsymbol{q}, \boldsymbol{p}\rangle + \lambda\|\boldsymbol{q}\|_Z + G(u, g)$$

We want to optimize a "Loss":

$$\mathcal{L}(\boldsymbol{F}) = \frac{1}{MNS}\sum_{s=1}^{S} \ell(u_s^*(\boldsymbol{F}), t_s),$$

which measures the error between some targets t_s (*e.g.*, "ground truth solutions") and the computed solutions u_s^* for the data g_s.

Learning setting

In the general form, we have a bilinear saddle point problem:

$$\min_x \sup_y g(x) + \langle Kx, y \rangle - f^*(y)$$

with g, f^* convex. To simplify we assume g, f^* strongly convex so that $(x(K), y(K))$ is uniquely defined (and continuous). We need to *differentiate a Loss* $\mathcal{L}(K) = \ell(x, y)$ *with respect to* K.

Learning setting

In the general form, we have a bilinear saddle point problem:

$$\min_x \sup_y g(x) + \langle Kx, y\rangle - f^*(y)$$

with g, f^* convex. To simplify we assume g, f^* strongly convex so that $(x(K), y(K))$ is uniquely defined (and continuous). We need to *differentiate a Loss* $\mathcal{L}(K) = \ell(x, y)$ *with respect to* K.

The main difficulty with respect to the previous Homogenization setting is that before, the Loss was depending on the minimal energy, whose derivative wr. K is simply $x \otimes y$.

Learning setting

In the general form, we have a bilinear saddle point problem:

$$\min_x \sup_y g(x) + \langle Kx, y\rangle - f^*(y)$$

with g, f^* convex. To simplify we assume g, f^* strongly convex so that $(x(K), y(K))$ is uniquely defined (and continuous). We need to *differentiate a Loss* $\mathcal{L}(K) = \ell(x, y)$ *with respect to* K.

The main difficulty with respect to the previous Homogenization setting is that before, the Loss was depending on the minimal energy, whose derivative wr. K is simply $x \otimes y$.

Classical method (now): Implement a 1st order algorithm to approximate $x(K)$ with some x^n, $n \geq 1$. Then "unroll" the iterations $(x^0, \dots, x^n)$ and use automatic differentiation and back-propagation to estimate $\nabla_K x^n$.
Issues: strange dependence on x^0, and difficult if the problem is large or requires too many iterations (costs a lot of memory).

Learning setting

In the general form, we have a bilinear saddle point problem:

$$\min_x \sup_y g(x) + \langle Kx, y\rangle - f^*(y)$$

with g, f^* convex. To simplify we assume g, f^* strongly convex so that $(x(K), y(K))$ is uniquely defined (and continuous). We need to *differentiate a Loss* $\mathcal{L}(K) = \ell(x, y)$ *with respect to* K.

The main difficulty with respect to the previous Homogenization setting is that before, the Loss was depending on the minimal energy, whose derivative wr. K is simply $x \otimes y$.

Classical method (now): Implement a 1st order algorithm to approximate $x(K)$ with some x^n, $n \geq 1$. Then "unroll" the iterations $(x^0, \dots, x^n)$ and use automatic differentiation and back-propagation to estimate $\nabla_K x^n$.
Issues: strange dependence on x^0, and difficult if the problem is large or requires too many iterations (costs a lot of memory).

Alternative: *even more* **classical method**: sensivity analysis, adjoint state computed by a "Piggyback" algorithm [Griewank, Faure '03].

Piggyback Algorithm

First choose starting points (x^0, y^0, X^0, Y^0), $\theta \in (0, 1]$, τ, σ. Then for each $k \geq 0$:

1. $\tilde{x} = x^k - \tau K^* y^k$, $\tilde{X} = X^k - \tau(K^* Y^k + \nabla_x \ell(x^k, y^k))$;
2. compute using automatic differentiation $x^{k+1} = \text{prox}_{\tau g}(\tilde{x})$, $X^{k+1} = \nabla \text{prox}_{\tau g}(\tilde{x}) \cdot \tilde{X}$;
3. $\bar{x}^{k+1} := x^{k+1} + \theta(x^{k+1} - x^k)$, $\bar{X}^{k+1} := X^{k+1} + \theta(X^{k+1} - X^k)$,
4. $\tilde{y} = y^k + \sigma K \bar{x}^{k+1}$, $\tilde{Y} = Y^k + \sigma(K \bar{X}^{k+1} + \nabla_y \ell(x^k, y^k))$;
5. compute using a.d. again $y^{k+1} = \text{prox}_{\sigma f^*}(\tilde{y})$, $Y^{k+1} = \nabla \text{prox}_{\sigma f^*}(\tilde{y}) \cdot \tilde{Y}$;
6. return to 1.

The **blue** iterations are a classical primal-dual method for computing x, y. The **red** iterations are parallel iterations solving for the adjoint states.

Theoretical results

(AC., Pock, in preparation)

Theorem *Assume that g, f^* are strongly convex and let (x, y, X, Y) be a fixed point of the algorithm, for which $\nabla \mathrm{prox}_{\tau g}(x - \tau K^* y)$ and $\nabla \mathrm{prox}_{\tau f^*}(y + \sigma K x)$ exist. Then $\mathcal{L}$ is differentiable at K and:*

$$\nabla \mathcal{L}(K) = y \otimes X + Y \otimes x.$$

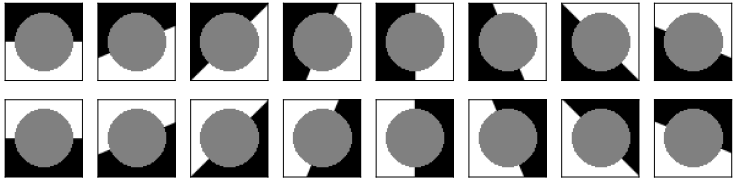
The convergence of the algorithm requires slightly more regularity:

Theorem *Assume that g, f^* are strongly convex, and in addition that g^*, f are locally $C^{2,\alpha}$ for some $\alpha > 0$. Then for τ, σ, θ properly chosen, the iterates (x^k, y^k, X^k, Y^K) converge linearly to a fixed point where the previous result holds.*

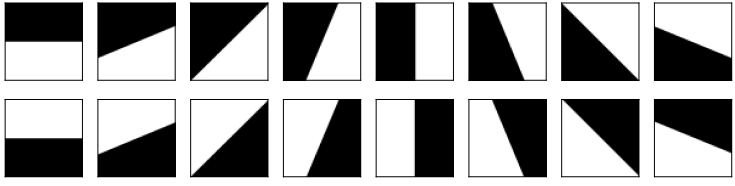
In practice: seems to work with less regularity...

Example: Learning for inpainting

- We train on 64 images of size 64×64 with directions uniformly sampled between $[0, 2\pi]$ and we include random subpixel shifts. Inertial gradient descent.
- Learn on a training set, evaluate on a test set.
- Experiments with different numbers of filters and different symmetry constraints for the filters.

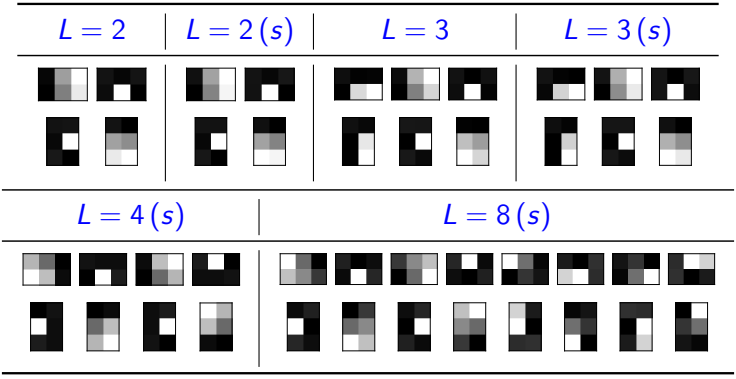


(a) Input images g_s



(b) Target images t_s

Results

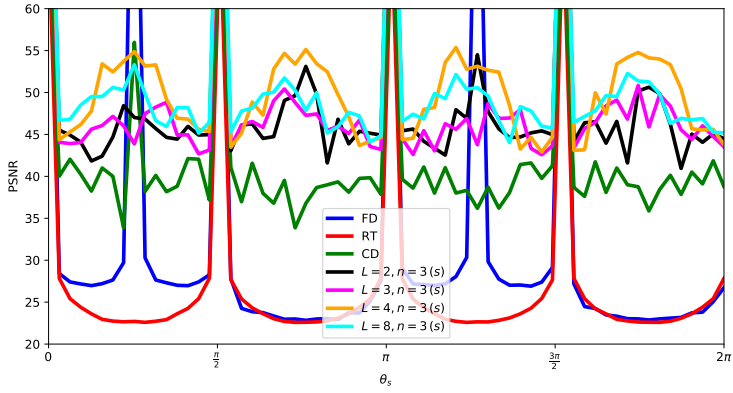


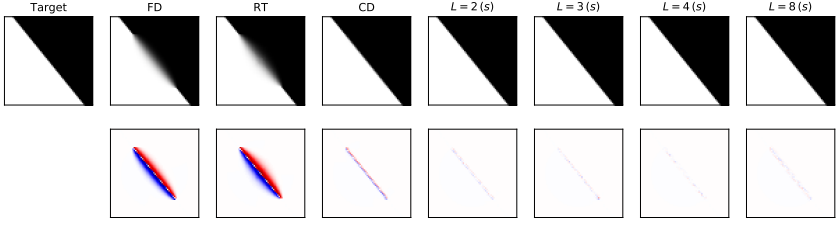
Data	FD	RT	CD	$L = 2$	$L = 2\,(s)$	$L = 3$	$L = 3\,(s)$	$L = 4\,(s)$	$L = 8\,(s)$
Train	135	195	6.69	1.26	1.22	1.19	1.27	0.85	0.77
Test	134	194	6.33	1.63	1.45	1.29	1.29	0.87	0.82

Table: $10^5 \times$MSE of handcrafted and learned filters for both training and test data.

(Note that transpose symmetry is almost automatically learned)

Comparison


PSNR
θ_s
FD
RT
CD
$L = 2, n = 3\,(s)$
$L = 3, n = 3\,(s)$
$L = 4, n = 3\,(s)$
$L = 8, n = 3\,(s)$


Target
FD
RT
CD
$L = 2\,(s)$
$L = 3\,(s)$
$L = 4\,(s)$
$L = 8\,(s)$

Example: denoising

(a) noisy

(b) forward diff.

(c) RT

(d) Condat

(e) "Shannon"

(f) learned (natural images)

(g) learned (disk denoising)

(h) learned (inpainting)

Conclusion

- Discretization of problems with discontinuous solution is very sensitive to the setting and the choice of parameters;
- We can propose a framework where these parameters can be tuned to solve a given task;
- We can implement a saddle-point algorithm which computes in parallel adjoint states for derivation;
- Still needed/ongoing: understand "Piggyback" method for non-smooth problems. Extend the framework to more singular energies (with constraints, curvature...)

Conclusion

- Discretization of problems with discontinuous solution is very sensitive to the setting and the choice of parameters;
- We can propose a framework where these parameters can be tuned to solve a given task;
- We can implement a saddle-point algorithm which computes in parallel adjoint states for derivation;
- Still needed/ongoing: understand "Piggyback" method for non-smooth problems. Extend the framework to more singular energies (with constraints, curvature...)

Thank you for your attention