Chapter 4

Primal-Dual Methods

Convex Optimization for Computer Vision
SS 2017

Michael Moeller
Visual Scene Analysis
Department of Computer Science
and Electrical Engineering
University of Siegen

Visual
Scene
Analysis

Primal-Dual Methods

Motivation

We do not have a method to solve problems of the form

$$\min_u \|u - f\|_1 + \alpha\|Du\|_1$$

although the proximal mapping of the ℓ^1-norm is easy to compute.

Can we build an algorithm around

$$\min_u \max_p G(u) + \langle p, Ku\rangle - F^*(p)?$$

Proximal mapping as implicit gradient descent

Interesting observation for differentiable E:

$$u^{k+1} = \text{prox}_{\tau E}(u^k) \quad \Rightarrow u^{k+1} = u^k - \tau\nabla E(u^{k+1})$$

The proximal mapping does an implicit gradient step!

The primal-dual hybrid gradient algorithm

Let us define

$$\mathrm{PD}(u,p) := G(u) + \langle p, Ku\rangle - F^*(p)$$

and try to alternate implicit accent steps in p with implicit descent steps in u:

$$\begin{aligned} p^{k+1} &= \ \mathrm{prox}_{-\sigma PD(u^k,\cdot)}(p^k) \\ u^{k+1} &= \ \mathrm{prox}_{\tau PD(\cdot,p^{k+1})}(u^k) \end{aligned}$$

One finds

$$\begin{aligned} p^{k+1} =& \mathrm{prox}_{-\sigma PD(u^k,\cdot)}(p^k), \\ =& \operatorname*{argmin}_p \frac{1}{2}\|p - p^k\|^2 + \sigma F^*(p) - \sigma\langle Ku^k, p\rangle \\ =& \operatorname*{argmin}_p \frac{1}{2}\|p - p^k - \sigma Ku^k\|^2 + \sigma F^*(p) \\ =& \mathrm{prox}_{\sigma F^*}(p^k + \sigma Ku^k) \end{aligned}$$

The primal-dual hybrid gradient algorithm

Let us define

$$\mathrm{PD}(u,p) := G(u) + \langle p, Ku\rangle - F^*(p)$$

and try to alternate implicit accent steps in p with implicit descent steps in u:

$$\begin{aligned} p^{k+1} &= \operatorname{prox}_{\sigma F^*}(p^k + \sigma K u^k) \\ u^{k+1} &= \operatorname{prox}_{\tau PD(\cdot, p^{k+1})}(u^k) \end{aligned}$$

One finds

$$\begin{aligned} u^{k+1} =& \operatorname{prox}_{\tau PD(\cdot, p^{k+1})}(u^k), \\ =& \operatorname*{argmin}_u \frac{1}{2}\|u - u^k\|^2 + G(u) + \langle Ku, p^{k+1}\rangle \\ =& \operatorname*{argmin}_u \frac{1}{2}\|u - u^k + \tau K^* p^{k+1}\|^2 + \tau G(u) \\ =& \operatorname{prox}_{\tau G}(u^k - \tau K^* p^{k+1}) \end{aligned}$$

Primal-dual hybrid gradient method

We found

$$p^{k+1} = \text{prox}_{\sigma F^*}(p^k + \sigma K u^k),$$
$$u^{k+1} = \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}).$$

One should make one (currently unintuitive) modification:

PDHG

We will call the iteration

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K \bar{u}^k), \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k. \end{aligned} \tag{PDHG}$$

the **Primal-Dual Hybrid Gradient Method**. As we will see, it converges if $\tau\sigma < \frac{1}{\|K\|^2}$.

References for PDHG

PDHG is commonly referred to as the Chambolle and Pock algorithm. Nevertheless, several authors contributed to the development. PDHG can be also be derived as a preconditioned version of a classical method (more later).

Here is a (likely imcomplete) list of relevant papers:

- Pock, Cremers, Bischof, Chambolle, A convex relaxation approach for computing minimal partitions.
- Esser, Zhang, Chan, A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science.
- Chambolle, Pock, A first-order primal-dual algorithm for convex problems with applications to imaging.
- Zhang, Burger Osher, A unified primal-dual algorithm framework based on Bregman iteration.

Understanding PDHG

Why does PDHG work?

1. Sanity check: If the algorithm converges, we found a minimizer!

2. Why does PDHG converge? Computation on the board:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \underbrace{\begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix}}_{=:T} \begin{pmatrix} u^{k+1} \\ p^{k+1} \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{1}{\tau} I & -K^T \\ -K & \frac{1}{\sigma} I \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{pmatrix}$$

for the set-valued operator $T : \mathbb{R}^n \times \mathbb{R}^n \to \mathcal{P}(\mathbb{R}^n) \times \mathcal{P}(\mathbb{R}^n)$

$$T(z) = \begin{pmatrix} \{K^T p\} + \partial G(u) \\ \partial F^*(p) - \{Ku\} \end{pmatrix}$$

for $z = (u; p)$.

Fixed point iteration

We found the optimality condition

$$0 \in Tz^{k+1} + M(z^{k+1} - z^k)$$

for a set-valued operator T and a matrix M. Let us define the process of computing the next iterate as the *resolvent*

$$z^{k+1} = (M + T)^{-1}(Mz^k). \tag{CPPA}$$

We already know one example of an iteration of the same form,

$$u^{k+1} = prox_E(u^k) = (I + \tau \partial E)^{-1}(u^k)$$

the proximal point algorithm.

The update (CPPA) is structurally very similar, so we can for the same tools to help us with the convergence analysis. We will call it a *customized proximal point algorithm* (CPPA).

Convergence of the CPPA

Remember what we did for the proximal gradient algorithm?

$\rightarrow$ Show that $prox_E = (I + \tau \partial E)^{-1}$ is firmly nonexpansive, i.e. averaged with $\alpha = 1/2$.

Remember what the crucial inequality was?

$$\langle p_u - p_v, u - v \rangle \geq 0 \qquad \forall u, v, p_u \in \partial E(u), p_v \in \partial E(v)$$

This can be generalized!

Monotone Operator

A set valued operator T is called *monotone* if the inequality

$$\langle p_u - p_v, u - v \rangle \geq 0$$

holds for all u, v, $p_u \in T(u)$ and $p_v \in T(v)$.

Convergence of the CPPA

This has the potential to show convergence of

$$0 \in T(z^{k+1}) + z^{k+1} - z^k, \tag{PPA}$$

provided that the above iteration is well-defined, i.e. the resolvent $(I + T)^{-1}(z)$ is defined for any $z \in \mathbb{R}^n$. This is a technical issue which can be resolved by considering *maximal* monotone operators. In the settings we are considering, this is not an issue.

Convergence proximal point algorithm

Let T be a maximal monotone operator, and let there exist a z such that $0 \in T(z)$. Then the (generalized) proximal point algorithm (PPA) converges to a point $\tilde{z}$ with $0 \in T(\tilde{z})$.

Convergence of the CPPA

But we wrote the PDHG algorithm as

$$0 \in T(z^{k+1}) + Mz^{k+1} - Mz^k, \qquad \text{(CPPA)}$$

i.e. with an additional matrix M.

Idea: For symmetric positive definite matrices, write $M = L^T L$ and rewrite (CPPA) as

$$0 \in L^{-T} T L^{-1}(\zeta^{k+1}) + \zeta^{k+1} - \zeta^k, \qquad \text{(CPPA)}$$

with $\zeta^k = Lz^k$, and

$$L^{-T} T L^{-1}(\zeta) = \{q \in \mathbb{R}^n \mid q = L^{-T} p, \;\; p \in T(L^{-1}\zeta)\}.$$

Lemma

If T is monotone, then $L^{-T} T L^{-1}$ is monotone, too.

Proof: Exercise.

Convergence conclusions CPPA

Convergence CPPA

Let T be a maximally monotone operator. Let there exist a z such that $0 \in T(z)$, and let the matrix M be symmetric positive definite. Then the customized proximal point algorithm

$$z^{k+1} = (M + T)^{-1}(Mz^k)$$

converges to a $\hat{z}$ with $0 \in T(z)$.

Convergence conclusions PDHG

The primal-dual hybrid gradient method

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K(2u^k - u^{k-1})), \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \end{aligned} \tag{PDHG}$$

can be rewritten (after an index shift) as

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \underbrace{\begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix}}_{=:T} \begin{pmatrix} u^{k+1} \\ p^{k+1} \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{1}{\tau} I & -K^T \\ -K & \frac{1}{\sigma} I \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{pmatrix}.$$

Convergence PDHG

The operator T is maximally monotone. For $\tau\sigma < \frac{1}{\|K\|^2}$ the matrix M in the PDHG algorithm is positive definite. Hence, PDHG converges.

(Assuming F and G to be proper, closed, and convex, assuming there is a $u \in \text{ri}(G)$ such that $Ku \in \text{ri}(F)$, and assuming the existence of a minimizer).

Applications of PDHG

ROF Denoising

$$\min P(u) = \min_u \frac{1}{2}\|u - f\|^2 + \alpha\|Ku\|_{2,1}$$

with K being a discretization of the multichannel gradient operator.

ROF Denoising

We write

$$\min_u P(u) = \min_u \max_p \frac{1}{2}\|u - f\|^2 + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p).$$

The (PDHG) updates are

$$\begin{aligned}
p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K\bar{u}^k)\\
u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}),\\
\bar{u}^{k+1} &= 2u^{k+1} - u^k.
\end{aligned}$$

which in this case amounts to

$$\begin{aligned}
p^{k+1} &= \arg\min_p \frac{1}{2}\|p - (p^k + \sigma K\bar{u}^k)\|^2 + \sigma\iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p),\\
u^{k+1} &= \arg\min_u \frac{1}{2}\|u - (u^k - \tau K^* p^{k+1})\|^2 + \frac{\tau}{2}\|u - f\|^2\\
&= \frac{u^k - \tau K^* p^{k+1} + \tau f}{1+\tau}\\
\bar{u}^{k+1} &= 2u^{k+1} - u^k.
\end{aligned}$$

TV-L^1 Denoising

$$\min P(u) = \min_u \|u - f\|_1 + \alpha \|Ku\|_{2,1}$$

with K being a discretization of the multichannel gradient operator.

TV-L^1 Denoising

We write

$$\min_u P(u) = \min_u \max_p \frac{1}{2}\|u - f\|_1 + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty} \leq \alpha}(p).$$

The (PDHG) updates are

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K \bar{u}^k) \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k. \end{aligned}$$

which in this case amounts to

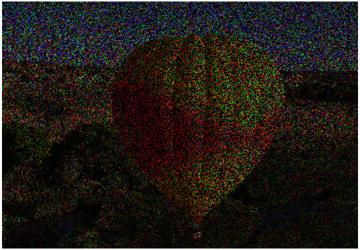
An exercise! :-)

TV-Inpainting

$$\min P(u) = \min_u \iota_{f_{|I}}(u) + \alpha \|Ku\|_{2,1}$$

with K being a discretization of the color gradient operator, and

$$\iota_{f_{|I}}(u) = \begin{cases} 0 & \text{if } u_i = f_i \text{ for all } i \in I, \\ \infty & \text{otherwise.} \end{cases}.$$



TV-Inpainting

We write

$$\min_u P(u) = \min_u \max_p \iota_{f_{|I}}(u) + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p).$$

The (PDHG) updates are

$$\begin{aligned}
p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K\bar{u}^k)\\
u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}),\\
\Rightarrow u_i^{k+1} &= \begin{cases} f_i & \text{if } i \in I,\\ (u^k - \tau K^* p^{k+1})_i & \text{otherwise.}\end{cases}\\
\bar{u}^{k+1} &= 2u^{k+1} - u^k.
\end{aligned}$$

TV-Deblurring

$$\min P(u) = \min_u \frac{1}{2}\|Au - f\|^2 + \alpha\|Ku\|_{2,1}$$

with K being a discretization of the multichannel gradient operator, A being a convolution with a blur kernel.

TV-Deblurring - Option 1

We write

$$\min_u P(u) = \min_u \max_p \frac{1}{2}\|Au - f\|^2 + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p).$$

The (PDHG) updates are

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K\bar{u}^k) \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k. \end{aligned}$$

which in this case amounts to

$$\begin{aligned} p^{k+1} &= \operatorname*{argmin}_p \frac{1}{2}\|p - (p^k + \sigma K\bar{u}^k)\|^2 + \sigma\iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p), \\ u^{k+1} &= \operatorname*{argmin}_u \frac{1}{2}\|u - (u^k - \tau K^* p^{k+1})\|^2 + \frac{\tau}{2}\|Au - f\|^2 \\ &= (I + \tau A^* A)^{-1}(u^k - \tau K^* p^{k+1} + \tau f) \\ \bar{u}^{k+1} &= 2u^{k+1} - u^k. \end{aligned}$$

TV-Deblurring - Option 2

We write

$$
\begin{aligned}
&\min_u P(u) \\
&= \min_u \max_{p,q} \langle Au - f, q\rangle - \frac{1}{2}\|q\|^2 + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p) \\
&= \min_u \max_{p,q} \left\langle \begin{pmatrix} A \\ K \end{pmatrix} u, \begin{pmatrix} q \\ p \end{pmatrix} \right\rangle - \langle f, q\rangle - \frac{1}{2}\|q\|^2 - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p)
\end{aligned}
$$

Now we have

$$
\begin{aligned}
F^*(p,q) &= \langle f, q\rangle + \frac{1}{2}\|q\|^2 + \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p) \\
G(u) &= 0 \\
\tilde{K} &= \begin{pmatrix} A \\ K \end{pmatrix}
\end{aligned}
$$

TV-Deblurring - Option 2

The (PDHG) updates are

$$
\begin{aligned}
q^{k+1} &= \operatorname*{argmin}_q \frac{1}{2}\|q - (q^k + \sigma A\bar{u}^k)\|^2 + \sigma\langle f, q\rangle + \frac{\sigma}{2}\|q\|^2,\\
p^{k+1} &= \operatorname*{argmin}_p \frac{1}{2}\|p - (p^k + \sigma K\bar{u}^k)\|^2 + \sigma\iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p),\\
u^{k+1} &= u^k - \tau K^* p^{k+1} - \tau A^* q^{k+1}\\
\bar{u}^{k+1} &= 2u^{k+1} - u^k.
\end{aligned}
$$

TV-Zooming

$$\min P(u) = \min_u \frac{1}{2}\|Au - f\|^2 + \alpha\|Ku\|_{2,1}$$

with K being a discretization of the multichannel gradient operator, $A = DB$, with B being a convolution with a blur kernel, and D being a downsampling, e.g. a matrix

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

PDHG implementation: Option 2 from the previous example.

TV-Zooming

Input data

Nearest neighbor

TV Zooming

Image Segmentation

$$\min P(u) = \min_u \iota_\Delta(u) + \iota_{\geq 0}(u) + \langle u, f\rangle + \alpha \|Ku\|_{2,1}$$

where $K : \mathbb{R}^{n\times m\times c} \to \mathbb{R}^{nmc\times 2}$ being a discretization of the multichannel gradient operator, and

$$\iota_\Delta(u) = \begin{cases} 0 & \text{if } \sum_k u_{i,j,k} = 1, \ \forall (i,j) \\ \infty & \text{else.} \end{cases}$$

$$\iota_{\geq 0}(u) = \begin{cases} 0 & \text{if } u_{i,j,k} \geq 0, \ \forall (i,j,k) \\ \infty & \text{else.} \end{cases}$$

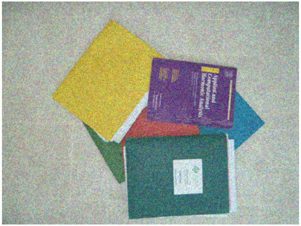
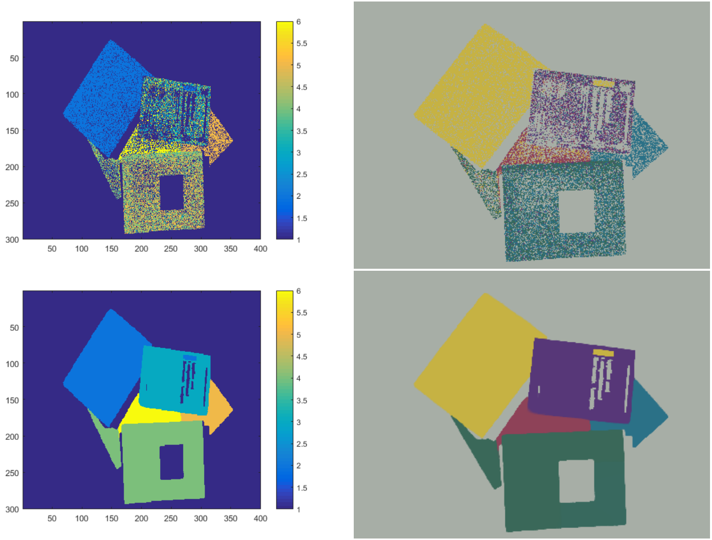


Image Segmentation



Upper row: data term minimization (=kmeans assignment), lower row: variational method

Image Segmentation

Option 1: We solve

$$\min_u \max_p \iota_\Delta(u) + \iota_{\geq 0}(u) + \langle u, f\rangle + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p).$$

$\rightarrow$ Primal proximal operator: Projection onto unit simplex.

Option 2: We solve

$$\min_u \max_{p.q} \langle Su - 1, q\rangle + \iota_{\geq 0}(u) + \langle u, f\rangle + \langle Ku, p\rangle - \iota_{\|\cdot\|_{2,\infty}\leq\alpha}(p).$$

where $(Su)_{i,j} = \sum_k u_{i,j}$.

$\rightarrow$ Very simple proximal operators, but additional variable.

Beyond total variation regularization

We used $J(u) = \|Ku\|_{2,1}$ in most applications.

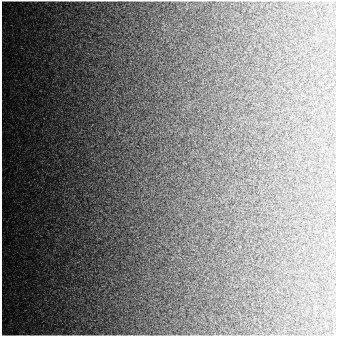
Pros:

- Preserves discontinuities/jumps
- Easy to use and fast to implement
- Good quality on images that are well approximated by piecewise constant images

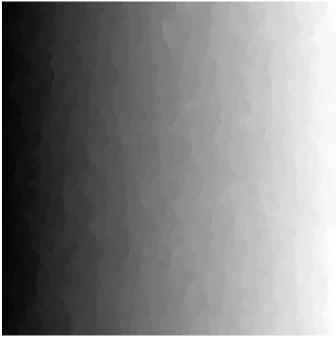
Cons:

- Staircasing
- Not necessarily a realistic image prior

Total generalized variation[1]



Input image



TV denoising

Can one avoid this ”staircasing” effect while preserving the ability to reconstruct jumps?

[1] *Total Generlaized Variation*, Bredis, Kunisch, Pock, SIAM Imag. Sci., 2010

Total generalized variation

Regularization

$$\begin{aligned} TGV(u) &= \min_{Ku=z_1+z_2} \|z_1\|_{2,1} + \beta\|\tilde{K}z_2\|_{2,1} \\ &= \min_{z} \|Ku - z\|_{2,1} + \beta\|\tilde{K}z\|_{2,1} \end{aligned}$$

with $\tilde{K} : \mathbb{R}^{nm\times 2c} \to \mathbb{R}^{nm\times 4c}$ being a discretization of the derivative of a vector field.

Idea: Optimally divide into a penalty of the first derivative (TV) and the Hessian.

TGV denoising

Consider the TGV denoising problem

$$\min_{u,z} \frac{1}{2}\|u-f\|_2^2 + \alpha\left(\|Ku - z\|_{2,1} + \beta\|\tilde{K}z\|_{2,1}\right)$$

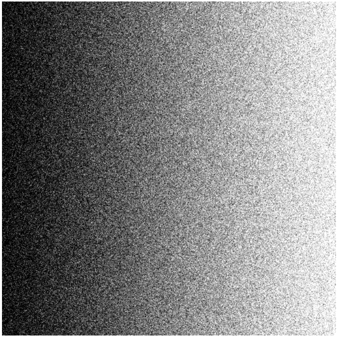
How can we minimize such a function (with PDHG)?

$$\min_{u,z} \frac{1}{2}\|u-f\|_2^2 + \alpha\left\|\begin{pmatrix} Ku - z \\ \beta\tilde{K}z \end{pmatrix}\right\|_{2,1}$$

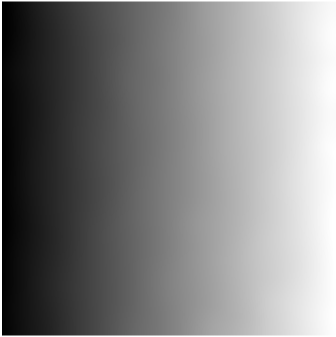
$$= \min_{u,z} \frac{1}{2}\|u-f\|_2^2 + \alpha\left\|\underbrace{\begin{pmatrix} K & -I \\ 0 & \tilde{K} \end{pmatrix}}_{=:D}\underbrace{\begin{pmatrix} u \\ z \end{pmatrix}}_{=:v}\right\|_{2,1}$$

$$= G(v) + F(Dv)$$

TGV denoising



Input image



TGV denoising

Final remark for applications

If you are too lazy to compute the proximity operator of F^*

$$
\begin{aligned}
\tilde{p} &= \mathrm{prox}_{\sigma F^*}(z) \\
&= \arg\min_p \frac{1}{2}\|p - z\|^2 + \sigma F^*(p) \\
\Rightarrow 0 &= \tilde{p} - z + \sigma \tilde{u}, \quad \tilde{u} \in \partial F^*(\tilde{p}) \\
\Rightarrow 0 &= \tilde{u} - z/\sigma + \frac{1}{\sigma}\tilde{p}, \quad \tilde{p} \in \partial F(\tilde{u}) \\
\Rightarrow \tilde{u} &= \mathrm{prox}_{\frac{1}{\sigma}F}(z/\sigma) \\
\Rightarrow \tilde{p} &= z - \sigma\, \mathrm{prox}_{\frac{1}{\sigma}F}(z/\sigma)
\end{aligned}
$$

Moreau's identity

If you know prox_F you also know prox_{F^*},

$$
\mathrm{prox}_{\sigma F^*}(z) = z - \sigma\, \mathrm{prox}_{\frac{1}{\sigma}F}(z/\sigma).
$$

Modifications of PDHG

Convergence rate

We have seen: PDHG works very well on problems of the form

$$\min G(u) + F(Ku),$$

for which F and G are simple.

We get a convergence rate of

$$\min_{j \in \{0,\dots,k\}} \|(I + L^{-T} T L^{-1})(\xi^k) - \xi^k\|^2 \leq C \frac{\|\xi^0 - \xi^0\|}{k+1}$$

for $\xi^k = L(u^k, p^k)$, L being the matrix square-root of M, and C being a constant.

According to Chambolle and Pock we also get ergodic convergence of the partial primal-dual gap with rate $\mathcal{O}(1/k)$.

What if our problem is "more friendly"? E.g. what if G or F or both are strongly convex?

Either G or F^* is strongly convex

$$
\begin{aligned}
p^{k+1} &= \text{prox}_{\sigma_k F^*}(p^k + \sigma_k K \bar{u}^k), \\
u^{k+1} &= \text{prox}_{\tau_k G}(u^k - \tau_k K^* p^{k+1}), \\
\theta_k &= \frac{1}{\sqrt{1 + 2\gamma\tau_k}}, \\
\tau_{k+1} &= \theta_k \tau_k, \quad \sigma_{k+1} = \sigma_k / \theta_k \\
\bar{u}^{k+1} &= u^{k+1} + \theta_k (u^{k+1} - u^k).
\end{aligned}
\tag{PDHG2}
$$

for $\tau_0 \sigma_0 \leq \|K\|^2$, and G being γ-strongly convex.

Theorem about (PDHG2), strongly convex G, Chambolle, Pock '10

For any $\epsilon > 0$ there exists an N_0 such that for any $N \geq N_0$:

$$
\|\tilde{u} - u^N\|^2 \leq \frac{1+\epsilon}{\gamma^2 N^2} \left(\frac{\|\tilde{u} - u^0\|^2}{\tau_0^2} + \|K\|^2 \|\tilde{p} - p^0\|^2 \right)
$$

One strongly convex function

Discussion of the convergence orders:

- Didn't the gradient methods obtain linear convergence on strongly convex energies?
- Yes, but we additionally needed a part of the energy to be L-smooth!
- Note that L-smoothness of the primal corresponds to $1/L$-strong convexity of the convex conjugate!
- What can we do if we additionally assume F to be L-smooth, i.e., assume F^* to be strongly convex?

Two strongly convex functions

Consider

$$\begin{aligned} p^{k+1} &= \text{prox}_{\sigma F^*}(p^k + \sigma K\bar{u}^k), \\ u^{k+1} &= \text{prox}_{\tau G}(u^k - \tau K^* p^{k+1}), \\ \bar{u}^{k+1} &= u^{k+1} + \theta(u^{k+1} - u^k). \end{aligned} \qquad \text{(PDHG3)}$$

Chambolle, Pock '10

For $\mu \leq 2\sqrt{\gamma\delta}/\|K\|$, $\tau = \mu/(2\gamma)$, $\sigma = \mu/(2\delta)$, $\theta \in [1/(1+\mu), 1]$, G being γ-strongly convex and F^* being δ-strongly convex, there exists $\omega < 1$, such that the iterates of (PDHG3) meet

$$\gamma\|u^N - \tilde{u}\|^2 + (1-\omega)\delta\|p^N - \tilde{p}\|^2 \leq \omega^N(\gamma\|u^0 - \tilde{u}\|^2 + \delta\|p^0 - \tilde{p}\|^2).$$

$\rightarrow$ Linear convergence!

Generic form

Remember the optimality conditions of the saddle point formulation

$$\min_u \max_p \; G(u) + \langle Ku, p\rangle - F^*(p)$$

were

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{p} \end{pmatrix}.$$

We could not compute $(\hat{u}, \hat{p})$ directly. Therefore,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix} \begin{pmatrix} u^{k+1} \\ p^{K+1} \end{pmatrix} + \underbrace{\begin{pmatrix} M_1 & M_3 \\ M_4 & M_2 \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{K+1} - p^k \end{pmatrix}$$

such that

- M is symmetric, i.e. $M_3 = (M_4)^T$,
- sequential updates are possible, i.e. $M_3 = -K^T$, or $M_4 = K$.

Diagonal M_1 and M_2

Sticking to $M_3 = -K^T$ leads to

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix} \begin{pmatrix} u^{k+1} \\ p^{K+1} \end{pmatrix} + \underbrace{\begin{pmatrix} M_1 & -K^T \\ -K & M_2 \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{K+1} - p^k \end{pmatrix}.$$

Only remaining requirement: M should be positive definite!

In PDHG we chose $M_1 = \frac{1}{\tau} I$, $M_2 = \frac{1}{\sigma} I$ for simplicity.

In many cases, e.g., for separable F^* and G, the updates remain easy to compute if M_1 and M_2 are diagonal.

Pock, Chambolle 2011

Let $\alpha \in [0, 2]$, $M_1 = \text{diag}(m_j^1)$ and $M_2 = \text{diag}(m_i^2)$ with

$$m_j^1 > \sum_i |K_{i,j}|^{2-\alpha}, \qquad m_i^2 > \sum_j |K_{i,j}|^{\alpha}.$$

Then M is positive definite.

Some remarks

Regarding the choice of M_1 and M_2:

- It does not influence the convergence rate.
- It is an active field of research to understand its influence on constants in the convergence rates.
- It can make a huge difference in practice!!
- Typically, the practical convergence speed improves the more information about K is included in M_1, M_2.

The latter motivates yet a different and vastly popular algorithm, the **alternating direction method of multipliers (ADMM)**.

ADMM

Let us consider

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix} \begin{pmatrix} u^{k+1} \\ p^{k+1} \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{1}{\lambda} I & -K^T \\ -K & \lambda K K^T \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{pmatrix}.$$

The resulting M is only positive semi-definite.

Either add ϵI, or exploit fixed point iterations of averaged operators in a different way to still show some kind of convergence.
For updating p, we need to solve

$$p^{k+1} = \arg\min_p F^*(p) + \frac{\lambda}{2} \left\| K^T p - K^T p^k - \frac{1}{\lambda} K(2u^{k+1} - u^k) \right\|^2,$$

such that we need a special structure of K or F^* to still be able to solve this subproblem.

Example: Graph projection splitting

For any generic problem of the form we are currently considering we can write

$$
\begin{aligned}
&\min_u H(u) + R(Du) \\
=\ &\min_{u,v,d} H(v) + R(d), \qquad \text{s.t.} \begin{pmatrix} I & -I & 0 \\ D & 0 & -I \end{pmatrix} \begin{pmatrix} u \\ v \\ d \end{pmatrix} = 0 \\
=\ &\min_{u,v,d} \max_p H(v) + R(d) + \left\langle \begin{pmatrix} I & -I & 0 \\ D & 0 & -I \end{pmatrix} \begin{pmatrix} u \\ v \\ d \end{pmatrix}, p \right\rangle
\end{aligned}
$$

Now we can identify $F^* = 0$ and the solution of the subproblem in p becomes a linear system!

Remarks on ADMM

ADMM is often derived from a different perspective. In this perspective, the above ADMM is the classical algorithm applied to the dual formulation of the problem. The primal version is

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial G & K^T \\ -K & \partial F^* \end{pmatrix} \begin{pmatrix} u^{k+1} \\ p^{K+1} \end{pmatrix} + \underbrace{\begin{pmatrix} \lambda K^T K & K^T \\ K & \frac{1}{\lambda} I \end{pmatrix}}_{=:M} \begin{pmatrix} u^{k+1} - u^k \\ p^{K+1} - p^k \end{pmatrix}.$$

and requires G to be sufficiently simple in order to solve the update equations, i.e.

$$p^{k+1} = \text{prox}_{\lambda F^*}(p^k + \lambda K u^k)$$

$$u^{k+1} = \arg\min_u \frac{\lambda}{2} \left\| Ku - Ku^k + \frac{1}{\lambda}(2p^{k+1} - p^k) \right\|^2 + G(u)$$

Some final remarks

Detailed convergence rate analysis of ADMM is still an active field of research. s Whether or not ADMM is faster than PDHG and its variants largely depends on how efficient the non-prox step can be computed.
It often even depends on the architecture you are computing on. Tendency:

- PDHG is better parallelizable $\rightarrow$ GPU
- ADMM makes more progress per iteration $\rightarrow$ CPU

Various (heuristic) suggestions for how to accelerate/approximate ADMM exist, e.g. a few iterations of preconditioned conjugate gradient (*pcg* in matlab).

Stopping customized proximal point algorithms

Generic form:

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \underbrace{\begin{bmatrix} M_1 & -K^T \\ -K & M_2 \end{bmatrix}}_{=:M} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}$$

such that the matrix M is positive (semi-)definite.

Natural considerations:

- How close is $-K^T p^{k+1}$ to being an element of $\partial G(u^{k+1})$?
- How close is Ku^{k+1} to being an element of $\partial F^*(p^{k+1})$?

We define the ***primal*** *and* ***dual*** *residuals*:

$$\begin{aligned} r_p^{k+1} &= M_2(p^{k+1} - p^k) - K(u^{k+1} - u^k) \\ r_d^{k+1} &= M_1(u^{k+1} - u^k) - K^T(p^{k+1} - p^k) \end{aligned}$$

Primal and dual residuals

Based on the ***primal*** *and* ***dual*** *residuals*:

$$r_p^{k+1} = M_2(p^{k+1} - p^k) - K(u^{k+1} - u^k)$$
$$r_d^{k+1} = M_1(u^{k+1} - u^k) - K^T(p^{k+1} - p^k)$$

we could consider our algorithm to be convergent if $\|r_d^{k+1}\|^2 + \|r_p^{k+1}\|^2 \to 0$, because this implies

$$\text{dist}(-K^T p^{k+1}, \partial G(u^{k+1})) \to 0,$$
$$\text{dist}(K u^{k+1}, \partial F^*(p^{k+1})) \to 0.$$

Note that this notion of convergences does not imply convergence of u^k and p^k yet!

Nevertheless, we know PDHG and ADMM do converge, and $\|r_d^{k+1}\|$ and $\|r_p^{k+1}\|$ are good measures for convergence!

Upper bounds on the residuals

How should we use $\|r_d^{k+1}\|$ and $\|r_p^{k+1}\|$ to formalize a stopping criterion?

- Simple option: Iterator until $\|r_d^{k+1}\| \leq \epsilon$ and $\|r_p^{k+1}\| \leq \epsilon$.
- Could be unfair, if $u^k \in \mathbb{R}^n$ and $p^k \in \mathbb{R}^m$ and e.g. $n >> m$. Use $\|r_d^{k+1}\| \leq \sqrt{n}\,\epsilon$ and $\|r_p^{k+1}\| \leq \sqrt{m}\,\epsilon$.
- Could be unfair for different scales! Introduce absolute and relative error criteria:

$$\begin{aligned}\|r_d^{k+1}\| &\leq \sqrt{n}\,\epsilon^{abs} + \text{dual scale factor} \cdot \epsilon^{rel}\\ \|r_p^{k+1}\| &\leq \sqrt{m}\,\epsilon^{abs} + \text{primal scale factor} \cdot \epsilon^{rel}\end{aligned}$$

But what are reasonable scale factors?

Scaling the primal residuum

The primal residual

$$r_p^{k+1} = M_2(p^{k+1} - p^k) - K(u^{k+1} - u^k)$$

measures how far Ku^{k+1} is away from a particular element in $\partial F^*(p^{k+1})$, and therefore scales with the magnitude of elements in $\partial F^*(p^{k+1})$.

More precisely:

$$\begin{aligned}
&0 \in \partial F^*(p^{k+1}) - Ku^{k+1} + r_p^{k+1} \\
\Rightarrow &0 \in \partial F^*(p^{k+1}) - K^T(2u^{k+1} - u^k) + M_2(p^{k+1} - p^k). \\
\Rightarrow &\underbrace{M_2(p^k - p^{k+1}) + K^T(2u^{k+1} - u^k)}_{=:z^{k+1}} \in \partial F^*(p^{k+1})
\end{aligned}$$

Thus, we can use

$$\|r_p^{k+1}\| \leq \sqrt{m}\,\epsilon^{abs} + \|z^{k+1}\| \cdot \epsilon^{rel}$$

to be scale-independent.

Scaling the dual residuum

The dual residual

$$r_d^{k+1} = M_1(u^{k+1} - u^k) - K^T(p^{k+1} - p^k)$$

measures how far $-K^T p^{k+1}$ is away from a particular element in $\partial G(u^{k+1})$, and therefore scales with the magnitude of elements in $\partial G(u^{k+1})$.

More precisely:

$$\begin{aligned} & 0 \in \partial G(u^{k+1}) + K^T p^{k+1} + r_d^{k+1}. \\ \Rightarrow\ & 0 \in \partial G(u^{k+1}) + K^T p^k + M_1(u^{k+1} - u^k) \\ \Rightarrow\ & \underbrace{M_1(u^k - u^{k+1}) - K^T p^k}_{=: v^{k+1}} \in \partial G(u^{k+1}) \end{aligned}$$

Thus, we can use

$$\|r_d^{k+1}\| \leq \sqrt{n}\, \epsilon^{abs} + \|v^{k+1}\| \cdot \epsilon^{rel}$$

to be scale-independent.

A scaled absolute and relative stopping criterion

In summary, a good stopping criterion is

$$\|r_p^{k+1}\| \leq \sqrt{m}\,\epsilon^{abs} + \|z^{k+1}\| \cdot \epsilon^{rel},$$
$$\|r_d^{k+1}\| \leq \sqrt{n}\,\epsilon^{abs} + \|v^{k+1}\| \cdot \epsilon^{rel}.$$

Interesting observation in our previous considerations: ADMM/PDHG actually generates iterates $(u^{k+1}, p^{k+1}, v^{k+1}, z^{k+1})$ with

$$v^{k+1} \in \partial G(u^{k+1}), \qquad z^{k+1} \in \partial F^*(p^{k+1}).$$

The goal of all algorithms is to achieve convergence

$$\|\underbrace{z^{k+1} - Ku^{k+1}}_{=r_p^{k+1}}\| \to 0 \ \text{ and } \|\underbrace{v^{k+1} + K^T p^{k+1}}_{=r_d^{k+1}}\| \to 0!$$

Adaptive stepsizes

r_p^{k+1} and r_d^{k+1} determine the convergence of the algorithm.

Can we also use r_d and r_p to accelerate the algorithm?

Adaptive stepsizes:

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau^k} M_1 & -K^T \\ -K & \frac{1}{\sigma^k} M_2 \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}$$

Base the choices of τ^k and σ^k on the residuals r_p^k and r_d^k, where

$$r_p^{k+1} = \frac{1}{\sigma^k} M_2 (p^{k+1} - p^k) - K(u^{k+1} - u^k),$$
$$r_d^{k+1} = \frac{1}{\tau^k} M_1 (u^{k+1} - u^k) - K^T (p^{k+1} - p^k)?$$

Customized proximal point algorithms

Decreasing residual balancing: Let $(M_1, -K^T; -K, M_2)$ be positive definite. Pick τ^0 and σ^0 with $\tau^0\sigma^0 < 1$. Further choose $\mu > 1$, $\alpha^0 < 1$, $\beta < 1$ and adapt as follows

- If $\|r_p^k\| > \mu\|r_d^k\|$, do
$$\tau^{k+1} = (1-\alpha^k)\tau^k, \quad \sigma^{k+1} = \frac{1}{1-\alpha^k}\sigma^k, \quad \alpha^{k+1} = \alpha^k \cdot \beta.$$
- If $\|r_d^k\| > \mu\|r_p^k\|$, do
$$\tau^{k+1} = \frac{1}{1-\alpha^k}\tau^k, \quad \sigma^{k+1} = (1-\alpha^k)\sigma^k, \quad \alpha^{k+1} = \alpha^k \cdot \beta.$$
- Keep $\tau^{k+1} = \tau^k$, $\sigma^{k+1} = \sigma^k$, and $\alpha^{k+1} = \alpha^k$ otherwise.

Goldstein et al., *Adaptive Primal-Dual Hybrid Gradient Methods for Saddle-Point Problems*: The resulting scheme still converges.

PDHG with backtracking

Start with any step sizes τ^0, σ^0, and constants $\gamma, \beta \in]0,1[$. Do

$$
\begin{aligned}
u^{k+1} &= \mathrm{prox}_{\tau^k G}(u^k - \tau^k K^T p^k) \\
p^{k+1} &= \mathrm{prox}_{\sigma^k F}(p^k + \sigma^k K(2u^{k+1} - u^k))
\end{aligned}
$$

Compute

$$
b^k = \frac{2\tau^k \sigma^k \langle p^{k+1} - p^k, K(u^{k+1} - u^k)\rangle}{\gamma \sigma^k \|u^{k+1} - u^k\|^2 + \gamma \tau^k \|p^{k+1} - p^k\|^2}
$$

If $b^k \leq 1$ keep iterating, if $b^k > 1$ update

$$
\tau^{k+1} = \beta \tau^k / b^k, \quad \sigma^{k+1} = \beta \sigma^k / b^k
$$

Key insight to prove convergence: $b^k > 1$ can only happen finitely many times.

Summary

For proper, closed, convex functions G and $F \circ K$ (with $\mathrm{ri}(\mathrm{dom}(G)) \cap \mathrm{ri}(\mathrm{dom}(F \circ K)) \neq \emptyset$) we can write

$$\min_u G(u) + F(Ku) = \min_u \max_p G(u) + \langle Ku, p \rangle - F^*(p).$$

with the optimality condition

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix}.$$

Typically, $(\hat{u}, \hat{p})$ cannot be computed directly, but one devises iterative methods on this saddle point problem.

Their idea is to decouple the update inclusions in u and p!

Their convergence can be shown via fixed-point iterations of averaged operators.

Saddle point methods

Most prominently, we discussed

- **PDHG, overrelaxation on primal**

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} I & -K^T \\ -K & \frac{1}{\sigma} I \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}.$$

- **PDHG, overrelaxation on dual**

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} I & K^T \\ K & \frac{1}{\sigma} I \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}.$$

- **Primal ADMM**

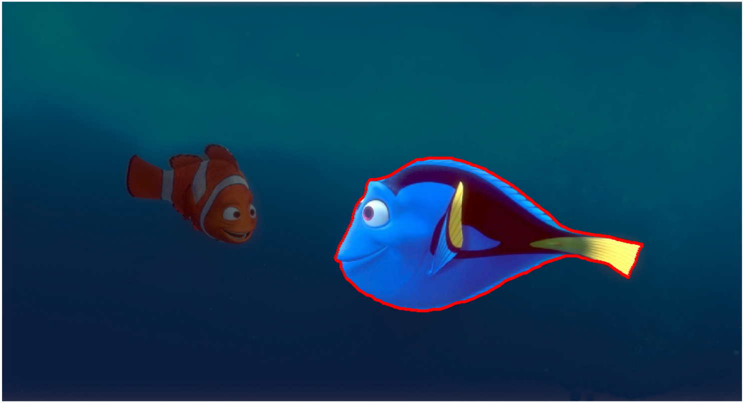
$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \begin{bmatrix} \lambda K^T K & K^T \\ K & \frac{1}{\lambda} I \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}.$$

- **Corresponding dual ADMM**

$$0 \in \begin{bmatrix} \partial G & K^T \\ -K & \partial F^* \end{bmatrix} \begin{bmatrix} u^{k+1} \\ p^{k+1} \end{bmatrix} + \begin{bmatrix} \frac{1}{\lambda} I & -K^T \\ -K & \lambda K K^T \end{bmatrix} \begin{bmatrix} u^{k+1} - u^k \\ p^{k+1} - p^k \end{bmatrix}.$$

Single view 3D reconstruction

Let Ω be the image domain, $S \subset \Omega$ an object.



From: Finding Nemo, `https://ohmy.disney.com/movies/2015/12/20/dory-finding-nemo-hero/`

Goal: Estimate a 3D model

First version: Single view 2.5D reconstruction

Oswald, Töppe, Cremers CVPR 2012: Find a height map that has minimal surface for fixed volume and respects the contour.

Mathematically for height map $u : S \to \mathbb{R}$

- $\int_S u(x)\, dx = V$, where V is a user given volume
- Constrain $u_{|\partial S} = 0$
- Minimize $\int_S \sqrt{1 + |\nabla u(x)|^2}\, dx$ (surface area)

Discrete form

$$\min_u \quad \sum_i \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u),$$

for a suitable gradient operator D (respecting $u_{|\partial S} = 0$),

$$\Sigma_V = \{u \in \mathbb{R}^{|S|} \mid \sum_i u_i = V\}.$$

Single view 2.5D reconstruction

How can we minimize

$$E(u) = \sum_i \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u) \ ?$$

One option: Gradient projection.

- Descent on the term that does not have an easy prox:

$$u^{k+1/2} = u^k - \tau D^* v^k, \qquad v_{i,:} = \frac{(Du^k)_{i,:}}{\sqrt{1 + |(Du^k)_{i,:}|^2}}$$

  for suitable τ, with $D : \mathbb{R}^n \to \mathbb{R}^{n \times 2}$.

- Project onto constraint set:

$$\text{proj}_{\Sigma_V}(v) = \underset{u}{\text{argmin}} \frac{1}{2}\|u - v\|_2^2 + \delta_{\Sigma_V}(u)$$

  Board: How does the projection look like?

Single view 2.5D reconstruction

$$\operatorname*{argmin}_u \frac{1}{2}\|u - v\|_2^2 + \delta_{\Sigma_V}(u) = \operatorname*{argmin}_u \frac{1}{2}\|u - v\|_2^2 + \delta_{\cdot - V}(\langle \mathbf{1}, u\rangle)$$

Optimality condition

$$0 = \hat{u} - v + \mathbf{1}p, \qquad p \in \partial\delta_{\cdot - V}(\langle \mathbf{1}, \hat{u}\rangle)$$

$$\sum_i \hat{u}_i = V$$

Take inner product of the above equation with $\mathbf{1}$:

$$0 = V - \sum_i v_i + np,$$

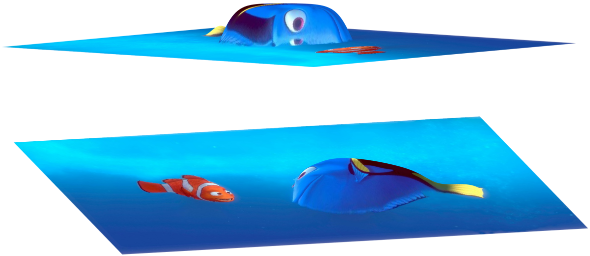
$$\Rightarrow p = \frac{1}{n}\left(V - \sum_i v_i\right),$$

which yields

$$\hat{u} = v - \mathbf{1}\frac{1}{n}\left(V - \sum_i v_i\right) = v - \text{mean}(v)\mathbf{1} + \mathbf{1}\frac{V}{n}$$

Single view 2.5D reconstruction

It works! :-)



Oringinal image from: Finding Nemo,

https://ohmy.disney.com/movies/2015/12/20/dory-finding-nemo-hero/

Single view 2.5D reconstruction

What about our primal-dual/splitting methods?

$$\min_{u} \quad \sum_{i} \sqrt{1 + |(Du)_i|^2} + \delta_{\Sigma_V}(u),$$

Natural reformulation:

$$\min_{u,d} \quad \sum_{i} \sqrt{1 + |d_i|^2} + \delta_{\Sigma_V}(u), \quad Du = d.$$

But is $F(d) = \sum_i \sqrt{1 + |d_i|^2}$ simple?

- Somewhat yes, as it reduces to a 1D problem.
- Somewhat no, as there is no (easy) closed form solution.

Reformulation that makes the prox operator really easy?

Single view 2.5D reconstruction

Let's start with

$$\min_{u,d} \quad \sum_i \sqrt{1+|d_i|^2} + \delta_{\Sigma_V}(u), \quad Du = d.$$

Note that

$$\sqrt{1+|d_i|^2} = \left|(d_i, 1)^T\right|$$

Idea: Introduce variable e with constraint $e_i = 1$ for all i!

$$\min_{u,d,e} \quad \underbrace{\sum_i \underbrace{\sqrt{e_i^2+|d_i|^2}}_{=|(d_i,e_i)^T|}}_{=\|(d,e)\|_{2,1}} + \delta_{\Sigma_V}(u), \quad Du = d, e = \mathbf{1}$$

Single view 2.5D reconstruction

$$\min_{u,d,e} \quad \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u), \quad Du = d, e = \mathbf{1}$$

Now the proximity operators of the two functions are simple!

$$\min_{u,d,e} \max_{p,q} \|(d,e)\|_{2,1} + \delta_{\Sigma_V}(u) + \left\langle \begin{pmatrix} p \\ q \end{pmatrix}, \begin{pmatrix} -D & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} u \\ d \\ e \end{pmatrix} \right\rangle - \langle q, \mathbf{1} \rangle$$

Option 1: Use (PDHG) now!

$\rightarrow$ *Board!*

Option 2: First save some variables, then apply (PDHG)!

$\rightarrow$ *Board!*

Single view 2.5D reconstruction

Single view 2.5D reconstruction

It still works! :-)

Single view 2.5D reconstruction

Single view 2.5D reconstruction