Total variation regularization for X-ray tomography

Vesa Kaarnioja

LUT School of Engineering Science

February 3, 2021

Some helpful resources on the Chambolle–Pock algorithm:

- A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision* **40**:120-145, 2011.
- L. Condat. A generic proximal algorithm for convex optimization – application to total variation minimization. *IEEE Signal Proc. Letters* **21**(8):985–989, 2014.
- E. Y. Sidky, J. H. Jørgensen, and X. Pan. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm. *Phys. Med. Biol.* **57**:3065–3091, 2012.
- Operator Discretization Library. `https://odl.readthedocs.io/math/solvers/nonsmooth/chambolle_pock.html`, 2017.
- PORTAL. `https://portal.readthedocs.io/en/latest/chambollepock.html`, written by P. Paleo, 2015.

Additional resources on total variation regularization for X-ray tomography:

- J. L. Mueller and S. Siltanen. Linear and Nonlinear Inverse Problems with Practical Applications. 2012.
- S. Siltanen. Total variation regularization for X-ray tomography. FIPS Computational Blog, `https://blog.fips.fi/tomography/x-ray/total-variation-regularization-for-x-ray-tomography/`, 2017.

Recall that the discrete measurement model for X-ray tomography can be expressed as

$$y = Ax.$$

This time, we consider solving the inverse problem of recovering x based on noisy measurements y.

We are interested in *anisotropic total variation regularization*

$$\arg\min_{x\geq 0}\Big\{\frac{1}{2}\|y - Ax\|^2 + \lambda\|Dx\|_1\Big\}, \quad \lambda > 0,$$

where $\|x\|_1 = \sum_i |x_i|$, $D = \begin{bmatrix} L_H \\ L_V \end{bmatrix}$ is the discretized (image) gradient operator,

$$\|Dx\|_1 = \sum_j |(Dx)_j| = \sum_j |(L_H x)_j| + \sum_j |(L_V x)_j|,$$

and L_H and L_V denote the horizontal and vertical (image) finite difference matrices, respectively.

Special feature: TV regularization preserves sharp edges.

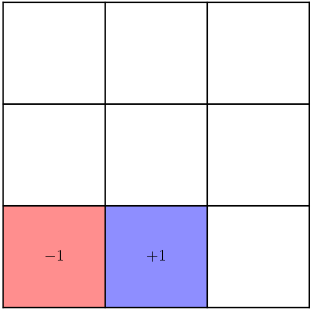
x_{91}	x_{92}	x_{93}	x_{94}	x_{95}	x_{96}	x_{97}	x_{98}	x_{99}	x_{100}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}	x_{89}	x_{90}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}	x_{79}	x_{80}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}	x_{69}	x_{70}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}	x_{59}	x_{60}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}	x_{49}	x_{50}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}	x_{39}	x_{40}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

Recall that the vector x is related to the density matrix $(f_{i,j})$ of the computational domain via

$$x_{(j-1)n+i} = f_{i,j}, \quad i, j \in \{1, \dots, n\}.$$

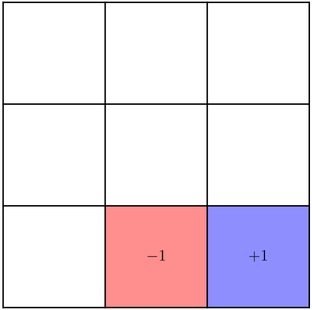
`x = f(:)` and `f = reshape(x,n,n)`

Construction of L_H (periodic boundary conditions)



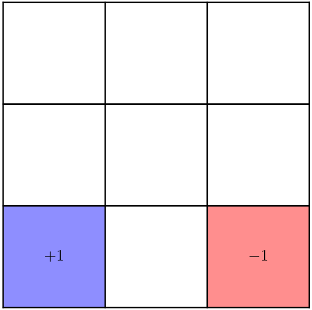
$$\begin{bmatrix} -1 & 1 & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



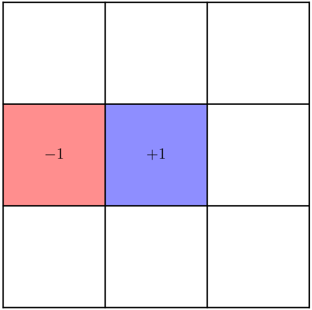
$$
\begin{bmatrix}
-1 & 1 & & & & & & & \\
 & -1 & 1 & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}
$$

Construction of L_H (periodic boundary conditions)



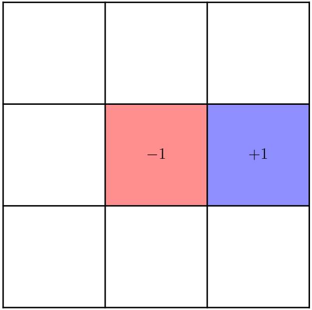
$$\begin{bmatrix} -1 & 1 & \\ & -1 & 1 \\ 1 & & -1 \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



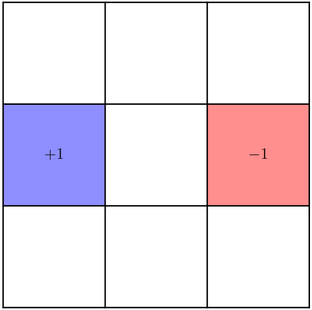
$$\begin{bmatrix} -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ 1 & & -1 & & & & & & \\ & & & -1 & 1 & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



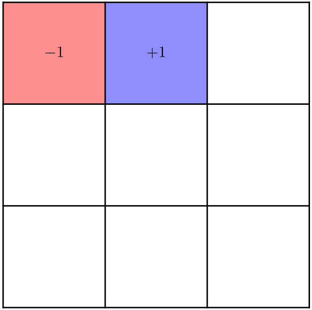
$$\begin{bmatrix}
-1 & 1 & & & & & & & \\
 & -1 & 1 & & & & & & \\
1 & & -1 & & & & & & \\
 & & & -1 & 1 & & & & \\
 & & & & -1 & 1 & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



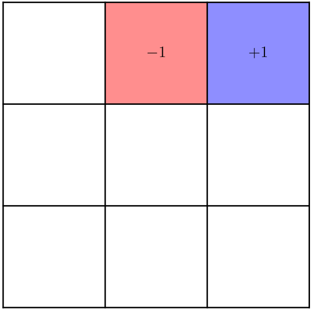
$$\begin{bmatrix} -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ 1 & & -1 & & & & & & \\ & & & -1 & 1 & & & & \\ & & & & -1 & 1 & & & \\ & & & 1 & & -1 & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



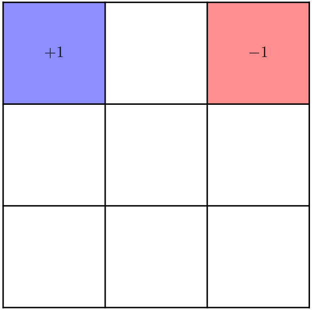
$$\begin{bmatrix}
-1 & 1 & & & & & & & \\
 & -1 & 1 & & & & & & \\
1 & & -1 & & & & & & \\
 & & & -1 & 1 & & & & \\
 & & & & -1 & 1 & & & \\
 & & & 1 & & -1 & & & \\
 & & & & & & -1 & 1 & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



$$\begin{bmatrix} -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ 1 & & -1 & & & & & & \\ & & & -1 & 1 & & & & \\ & & & & -1 & 1 & & & \\ & & & 1 & & -1 & & & \\ & & & & & & -1 & 1 & \\ & & & & & & & -1 & 1 \\ & & & & & & & & \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)



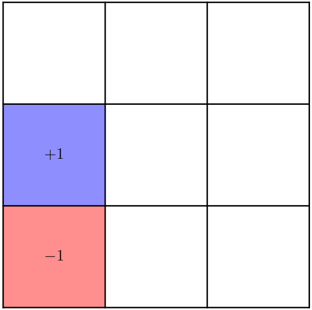
$$\begin{bmatrix} -1 & 1 & & & & & & & \\ & -1 & 1 & & & & & & \\ 1 & & -1 & & & & & & \\ & & & -1 & 1 & & & & \\ & & & & -1 & 1 & & & \\ & & & 1 & & -1 & & & \\ & & & & & & -1 & 1 & \\ & & & & & & & -1 & 1 \\ & & & & & & 1 & & -1 \end{bmatrix}$$

Construction of L_H (periodic boundary conditions)

$$
\begin{bmatrix}
-1 & 1 & & & & & & & \\
 & -1 & 1 & & & & & & \\
1 & & -1 & & & & & & \\
 & & & -1 & 1 & & & & \\
 & & & & -1 & 1 & & & \\
 & & & 1 & & -1 & & & \\
 & & & & & & -1 & 1 & \\
 & & & & & & & -1 & 1 \\
 & & & & & & 1 & & -1
\end{bmatrix}
$$

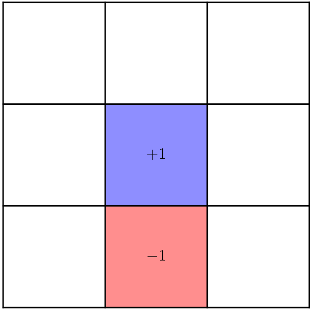
```
n = 3;
block = spdiags([1,-1,1].*ones(n,3),[1-n,0,1],n,n);
%block = -eye(n)+diag(ones(n-1,1),1); block(end,1) = 1;
LH = [];
for ii = 1:n
  LH = blkdiag(LH,block); % form the 9x9 block matrix
end
```

Construction of L_V (periodic boundary conditions)



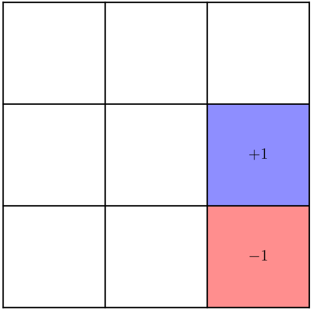
$$\begin{bmatrix} -1 & & & 1 & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_V (periodic boundary conditions)



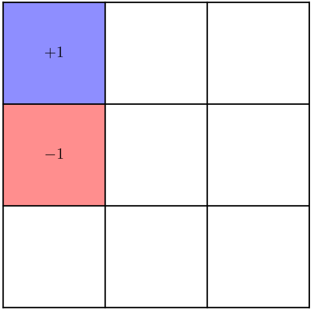
$$
\begin{bmatrix}
-1 & & & 1 & & & & & \\
 & -1 & & & 1 & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}
$$

Construction of L_V (periodic boundary conditions)



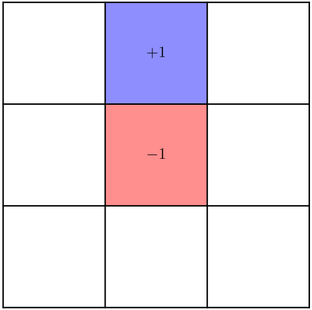
$$\begin{bmatrix} -1 & & & 1 & & & & & \\ & -1 & & & 1 & & & & \\ & & -1 & & & 1 & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_V (periodic boundary conditions)



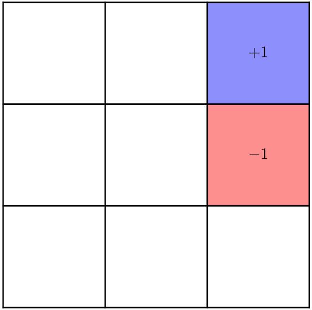
$$\begin{bmatrix} -1 & & & 1 & & & & & \\ & -1 & & & 1 & & & & \\ & & -1 & & & 1 & & & \\ & & & -1 & & & 1 & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \end{bmatrix}$$

Construction of L_V (periodic boundary conditions)



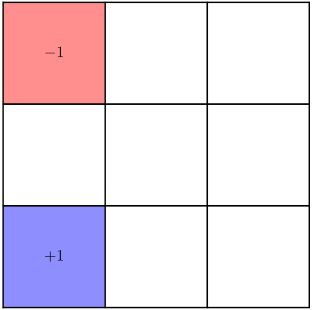
$$\begin{bmatrix}
-1 & & & 1 & & & & & \\
 & -1 & & & 1 & & & & \\
 & & -1 & & & 1 & & & \\
 & & & -1 & & & 1 & & \\
 & & & & -1 & & & 1 & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}$$

Construction of L_V (periodic boundary conditions)



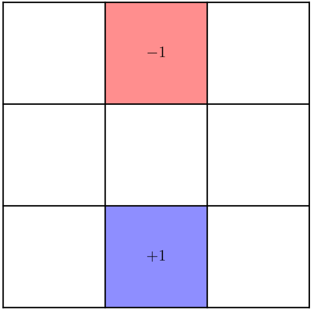
$$
\begin{bmatrix}
-1 & & & 1 & & & & & \\
 & -1 & & & 1 & & & & \\
 & & -1 & & & 1 & & & \\
 & & & -1 & & & 1 & & \\
 & & & & -1 & & & 1 & \\
 & & & & & -1 & & & 1 \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix}
$$

Construction of L_V (periodic boundary conditions)



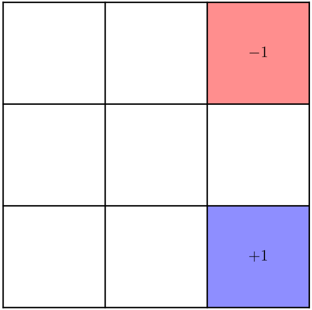
$$\begin{bmatrix}
-1 & & & 1 & & & & & \\
& -1 & & & 1 & & & & \\
& & -1 & & & 1 & & & \\
& & & -1 & & & 1 & & \\
& & & & -1 & & & 1 & \\
& & & & & -1 & & & 1 \\
1 & & & & & & -1 & & \\
& & & & & & & & \\
& & & & & & & &
\end{bmatrix}$$

Construction of L_V (periodic boundary conditions)



$$
\begin{bmatrix}
-1 & & & 1 & & & & & \\
 & -1 & & & 1 & & & & \\
 & & -1 & & & 1 & & & \\
 & & & -1 & & & 1 & & \\
 & & & & -1 & & & 1 & \\
 & & & & & -1 & & & 1 \\
1 & & & & & & -1 & & \\
 & 1 & & & & & & -1 & \\
 & & & & & & & &
\end{bmatrix}
$$

Construction of L_V (periodic boundary conditions)



$$
\begin{bmatrix}
-1 & & & 1 & & & & & \\
 & -1 & & & 1 & & & & \\
 & & -1 & & & 1 & & & \\
 & & & -1 & & & 1 & & \\
 & & & & -1 & & & 1 & \\
 & & & & & -1 & & & 1 \\
1 & & & & & & -1 & & \\
 & 1 & & & & & & -1 & \\
 & & 1 & & & & & & -1
\end{bmatrix}
$$

Construction of L_V (periodic boundary conditions)

$$\begin{bmatrix} -1 & & & 1 & & & & & \\ & -1 & & & 1 & & & & \\ & & -1 & & & 1 & & & \\ & & & -1 & & & 1 & & \\ & & & & -1 & & & 1 & \\ & & & & & -1 & & & 1 \\ 1 & & & & & & -1 & & \\ & 1 & & & & & & -1 & \\ & & 1 & & & & & & -1 \end{bmatrix}$$

```
n = 3;
LV = spdiags([1,-1,1].*ones(n^2,3),[-n^2+n,0,n],n^2,n^2);
%LV = -eye(n^2)+diag(ones(n^2-n,1),n)+diag(ones(n,1),n-n^2);
```

Let $F\colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ and $G\colon \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ be convex lower semicontinuous functions and K is a linear operator. Let F^* denote the *Fenchel conjugate* of F. Consider the abstract problem

$$\min_x \max_\eta \{\langle Kx, \eta\rangle + G(x) - F^*(\eta)\}.$$

The general form of the Chambolle–Pock algorithm can be written as

$$\begin{aligned}
\eta_{k+1} &= \mathrm{prox}_{\sigma F^*}(\eta_k + \sigma K \widetilde{x}_k), && \text{(update dual variable)}\\
x_{k+1} &= \mathrm{prox}_{\tau G}(x_k - \tau K^{\mathrm{T}} \eta_{k+1}), && \text{(update primal variable)}\\
\widetilde{x}_{k+1} &= x_{k+1} + \theta(x_{k+1} - x_k), && \text{(extrapolation)}
\end{aligned}$$

where $\tau > 0$ is the primal step size, $\sigma > 0$ is the dual step size, $\theta > 0$ is an extrapolation parameter, and the *proximal operator* of a function f is defined as

$$\mathrm{prox}_f(\eta) := \arg\min_x \left\{ f(x) + \tfrac{1}{2}\|x - \eta\|^2 \right\}.$$

If $\sigma\tau \leq 1/L^2$, $L = \|K\|_2$ (operator norm), and $\theta = 1$, then the algorithm can be shown to converge at linear rate $\mathcal{O}(k^{-1})$ [Chambolle and Pock 2010].

Let us recast the TV regularization problem

$$\min_{x\geq 0}\left\{\tfrac{1}{2}\|y-Ax\|^2+\lambda\|Dx\|_1\right\},\quad \lambda>0, \tag{1}$$

in the above framework.

- Note that

$$\tfrac{1}{2}\|Ax-y\|^2=\max_q\left\{\langle Ax-y,q\rangle-\tfrac{1}{2}\|q\|^2\right\},$$

  since $0=\nabla_q(\langle Ax-y,q\rangle-\frac{1}{2}\|q\|^2)=Ax-y-q$ iff $q=Ax-y$.
- Since $\|x\|_1=\sum_i|x_i|=\langle|x|,1\rangle=\langle x,\mathrm{sign}(x)\rangle$,

$$\lambda\|Dx\|_1=\max_{\|z\|_\infty\leq 1}\langle Dx,\lambda z\rangle=\max_{\|z\|_\infty\leq\lambda}\langle Dx,z\rangle=\max_z\left\{\langle Dx,z\rangle-\iota_\lambda(z)\right\},$$

  where $\iota_\lambda(z)=0$ if $\|z\|_\infty\leq\lambda$ and $\iota_\lambda(z)=+\infty$ otherwise.

Then (1) is equivalent to

$$\min_x\max_{q,z}\left\{\langle Ax-y,q\rangle+\langle Dx,z\rangle-\tfrac{1}{2}\|q\|^2-\iota_\lambda(z)+\iota_+(x)\right\},$$

where $\iota_+(x)=0$ if $x\geq 0$ and $\iota_+(x)=+\infty$ otherwise.

It is easy to see that

$$\min_{x} \max_{q,z} \{ \langle Ax - y, q \rangle + \langle Dx, z \rangle - \tfrac{1}{2}\|q\|^2 - \iota_\lambda(z) + \iota_+(x) \}$$

is tantamount to

$$\min_{x} \max_{q,z} \left\{ \left\langle Kx, \begin{bmatrix} q \\ z \end{bmatrix} \right\rangle + G(x) - F^*(q, z) \right\},$$

where

$$\begin{aligned} G(x) &= \iota_+(x), \\ F^*(q, z) &= \langle y, q \rangle + \tfrac{1}{2}\|q\|^2 + \iota_\lambda(z), \\ K &= \begin{bmatrix} A \\ D \end{bmatrix}. \end{aligned}$$

The proximal mapping corresponding to G is simply the projection onto $\{x \geq 0 \mid x \in \mathbb{R}^n\}$:

$$\operatorname{prox}_{\tau G}(x) = (\max(x_i, 0))_i = \texttt{max(x, 0)}.$$

On the other hand,

$$\operatorname{prox}_{\sigma F^*}(q, z) = \left(\frac{q - \sigma y}{1 + \sigma}, \frac{\lambda z}{\max\big(\lambda, |z|\big)} \right). \qquad \text{(N.B. } \eta = (q, z))$$

Noting that $K^{\mathrm{T}} = [A^{\mathrm{T}}, D^{\mathrm{T}}]$, the Chambolle–Pock algorithm takes the form

$$\begin{cases} \eta_{k+1} = \operatorname{prox}_{\sigma F^*}(\eta_k + \sigma K \widetilde{x}_k) \\ x_{k+1} = \operatorname{prox}_{\tau G}(x_k - \tau K^{\mathrm{T}} \eta_{k+1}) \\ \widetilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k) \end{cases}$$

$$\Leftrightarrow \quad \begin{cases} q_{k+1} = \frac{q_k + \sigma A \widetilde{x}_k - \sigma y}{1 + \sigma} \\ z_{k+1} = \frac{\lambda(z_k + \sigma D \widetilde{x}_k)}{\max(\lambda, |z_k + \sigma D \widetilde{x}_k|)} & \text{(elementwise division)} \\ x_{k+1} = \max(x_k - \tau A^{\mathrm{T}} q_{k+1} - \tau D^{\mathrm{T}} z_{k+1}, 0) & \text{(elementwise } \max) \\ \widetilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k). \end{cases}$$

Pseudocode for the Chambolle–Pock algorithm

Given: projection matrix A, data y, regularization parameter λ.

1. Form the difference matrices `L_H` and `L_V`. Set `D = [L_H;L_V];`
2. `L = svds([A;D],1); % alt. use the power method`
3. `tau = 1/L, sigma = 1/L, theta = 1;`
4. `x = zeros(size(A,2),1), q = zeros(size(A,1),1);`
5. `z = zeros(2*size(A,2),1), xhat = x;`

   Repeat

   6. `q = (q+sigma*(A*xhat-y))/(1+sigma);`
   7. `z = lambda * (z+sigma*D*xhat)./max(lambda,abs(z+sigma*D*xhat));`
   8. `xold = x;`
   9. `x = max(x-tau*A'*q-tau*D'*z,0);`
   10. `xhat = x+theta*(x-xold);`

   until convergence.