Solver for TV-TV Minimization

Marija Vella and João F. C. Mota

February 2020

1 Introduction

Let $X^\star \in \mathbb{R}^{M\times N}$ represent an image of which we have m linear observations denoted as b. Representing its column-major vectorization by $x^\star \in \mathbb{R}^n$, where $n := M \times N$, this can be represented by $b = Ax^\star$, where $A \in \mathbb{R}^{m\times n}$. We make the following assumptions:

- The image $X^\star$ has a small 2D total variation, i.e., a small 2D-TV norm, defined as $\|X\|_{\text{TV}} := \|Dx\|_1$, where x is the column-major vectorization of X, $\|\cdot\|_1$ the ℓ_1 -norm , and D is a matrix that works the differences between adjacent pixels in the vertical and horizontal directions. For example, with M = 3 and N = 3, D is

$$\begin{bmatrix}
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1\\
1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix} = \in \mathbb{R}^{2n\times n} = \mathbb{R}^{18\times 9}. \tag{1}$$

  Notice that the first (resp. last) n rows of D capture vertical (resp. horizontal) differences in the image.

- We have access to another image $W \in \mathbb{R}^{M\times N}$ that is similar to $X^\star$ in the 2D-TV norm, i.e., $\|X^\star - W\|_{\text{TV}}$ is expected to be small.

  Our goal is to reconstruct $X^\star$ from the linear measurements b and the side information W. Inspired by [3], we pose the problem as TV-TV minimization:

$$
\begin{array}{ll}
\underset{X}{\text{minimize}} & \|X\|_{\text{TV}} + \beta\|X - W\|_{TV} \\
\text{subject to} & A\,\text{vec}(X) = b,
\end{array} \tag{2}
$$

where vec(·) denotes the vectorization of a matrix. Using a matrix D with proper dimensions [see (1)], problem (2) can be written in vector form as

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & \|Dx\|_1 + \beta\|Dx - w\|_1 \\
\text{subject to} & Ax = b,
\end{array} \tag{3}
$$

where $w := D\ \text{vec(W)} \in \mathbb{R}^n$.

2 Algorithm

2.1 Multiplication by D and D^T

Before we proceed, we explore the structure of the matrix D in (1) to see how matrix-vector multiplications can be efficiently computed. First notice that D can be written as

$$
D = \begin{bmatrix} D_v \\ D_h \end{bmatrix}, \tag{4}
$$

where $D_v, D_h \in \mathbb{R}^{n\times n}$. The matrix D_h is circulant, i.e.,

$$
D_h = \text{circ}\left(\begin{bmatrix} -1 & 0 & \cdots & 0 & 1 & 0 & \cdots \end{bmatrix}^\top\right) := \text{circ}\left(c_h\right), \tag{5}
$$

and is generated by its first column $c_h \in \mathbb{R}^n$ (with -1 in entry 1, and 1 in entry $n - M + 1$). The matrix D_v is also circulant i.e.,

$$
D_v = \text{circ}\left(\begin{bmatrix} -1 & 0 & \cdots & 0 & 0 & 1 & \cdots \end{bmatrix}^\top\right) := \text{circ}\left(c_v\right), \tag{6}
$$

and is generated by its first column $c_v \in \mathbb{R}^n$ (with -1 in entry 1, and 1 in entry n). Being circulant, D_h and D_v are diagonalized by the DFT matrix with appropriate dimensions:

$$
D_h = F_n^H \text{Diag}\left(F_n c_h\right) F_n \tag{7}
$$

$$
D_v = F_n^H \text{Diag}\left(F_n c_v\right) F_n, \tag{8}
$$

where $F_q \in \mathbb{C}^{q\times q}$ is the $q \times q$ DFT matrix and Diag(f) is a diagonal matrix whose diagonal is given by f. The multiplication of a vector $x \in \mathbb{R}^n$ by D_h and by D_h^T can then be computed as

$$
D_h x = \left(F_n^H\, \text{Diag}\left(F_n c_h\right) F_n\right) x = F_n^H \left(\left(F_n c_h\right) \odot \left(F_n x\right)\right) = \text{ifft}_n\left(\text{ifft}_n\left(c_h\right) \odot \text{fft}_n(x)\right) \tag{9}
$$

$$
D_h^\top x = \left(F_n\, \text{Diag}\left(F_n c_h\right) F_n^H\right) x = F_n\left(\left(F_n c_h\right) \odot \left(F_n^H x\right)\right) = \text{fft}_n\left(\text{fft}_n\left(c_h\right) \odot \text{ifft}_n(x)\right). \tag{10}
$$

Similarly, D_v and D_v^T can be computed as

$$
D_v x = \left(F_n^H\, \text{Diag}\left(F_n c_v\right) F_n\right) x = F_n^H \left(\left(F_n c_v\right) \odot \left(F_n x\right)\right) = \text{ifft}_n\left(\text{ifft}_n\left(c_v\right) \odot \text{fft}_n(x)\right) \tag{11}
$$

$$
D_v^\top x = \left(F_n\, \text{Diag}\left(F_n c_v\right) F_n^H\right) x = F_n\left(\left(F_n c_v\right) \odot \left(F_n^H x\right)\right) = \text{fft}_n\left(\text{fft}_n\left(c_v\right) \odot \text{ifft}_n(x)\right), \tag{12}
$$

where $\odot$ denotes the Hadamard product. Therefore, multiplying $x \in \mathbb{R}^n$ by D and D^T can be computed as

$$Dx = \begin{bmatrix} D_v x \\ D_h x \end{bmatrix} \tag{13}$$

$$D^\top x = D_v^\top x + D_h^\top x. \tag{14}$$

2.2 Reformulation for ADMM

To solve (3) we use ADMM [1] by introducing two auxiliary variables $\boldsymbol{Dv = u}$ and $\boldsymbol{x = v}$ as suggested in [2]. We can thus rewrite (3) as

$$\begin{array}{ll} \underset{(u,x),v}{\text{minimize}} & \|u\|_1 + \beta\|u - w\|_1 \\ \text{subject to} & Ax = b \\ & Dv = u \\ & v = x\,. \end{array} \tag{15}$$

This is equivalent to:

$$\underset{(u,x),v}{\text{minimize}} \quad f(u,x) + g(v), \tag{16}$$

where $f(u,x) = \|u\|_1 + \beta\|u - w\|_1 + i_{\{x:Ax=b\}}(x)$ and $g(v) = i_{\{v:Dv=u\}}(v) + i_{\{v:x=v\}}(v)$. Eq.(16) is also equivalent to:

$$\begin{array}{ll} \underset{(u,x),v}{\text{minimize}} & f(u,x) + g(v) \\ \text{subject to} & Dv = u \\ & v = x\,, \end{array}$$

where $f(u,x) = \|u\|_1 + \beta\|u - w\|_1 + i_{\{x:Ax=b\}}(x)$ and $g(v) = 0$. Note that indicator functions can be replaced with constraints. This confirms that the problem is in the correct format to apply ADMM. The rest of the working is based on the notation of Eq.(15).

2.3 Augmented Lagrangian and ADMM iterations

The augmented Lagrangian of problem (15) is

$$L_\rho(u,x,v;\lambda,\mu) = \|u\|_1 + \beta\|u - w\|_1 + i_{\{x:Ax=b\}}(x) + \lambda^T(u - Dv) + \mu^T(x - v) + \frac{\rho}{2}\|u - Dv\|_2^2 + \frac{\rho}{2}\|x - v\|_2^2,$$

while the ADMM iterations are:

$$
\begin{aligned}
u^{k+1} =\ & L_\rho(u, x^k, v^k; \lambda^k, \mu^k) \\
=\ & \arg\min_u \ \ \|u\|_1 + \beta\|u - w\|_1 + (\lambda^k - \rho D v^k)^T u + \frac{\rho}{2}\|u\|_2^2 && (17) \\
x^{k+1} =\ & L_\rho(u^{k+1}, x, v^k; \lambda^k, \mu^k) \\
=\ & \arg\min_x \ \ {\mu^k}^T x + \frac{\rho}{2}\|x - v^k\|_2^2 + i_{Ax=b}(x) \\
=\ & \begin{array}{ll} \arg\min\limits_x & {\mu^k}^T x + \frac{\rho}{2}\|v^k - x\|_2^2 \\ \text{s.t.} & b = Ax \end{array} && (18) \\
v^{k+1} =\ & L_\rho(u^{k+1}, x^{k+1}, v; \lambda^k, \mu^k) \\
=\ & \arg\min_v \ \ -{\lambda^k}^T D v - {\mu^k}^T v + \frac{\rho}{2}\|x^{k+1} - v\|_2^2 + \frac{\rho}{2}\|u^{k+1} - Dv\|_2^2 && (19) \\
\lambda^{k+1} =\ & \lambda^k + \rho(u^{k+1} - D v^{k+1}) && (20) \\
\mu^{k+1} =\ & \mu^k + \rho(x^{k+1} - v^{k+1}). && (21)
\end{aligned}
$$

2.3.1 Problem in (u, x)

Considering the minimization of the Lagrangian with respect to (u, x) we end up with two problems that are independent of each other and can be worked out in parallel.

1) ***Problem in u:*** To find the solution of (17), let $s := \lambda^k - \rho D v^k$. Then (17) reads as:

$$
u^{k+1} = \arg\min_u \ \ \|u\|_1 + \beta\|u - w\|_1 + s^T u + \frac{\rho}{2}\|u\|_2^2,
$$

whose *ith* component is given by:

$$
u_i^{k+1} = \arg\min_{u_i} = |u_i| + \beta|u_i - w_i| + s_i u_i + \frac{\rho}{2} u_i^2. \tag{22}
$$

To find a closed-form solution of (22), we need to consider the following cases:

- $w_i > 0$

$-u_i < 0$: The optimality condition in this case are:

$$
0 = -1 - \beta + s_i + \rho u_i \iff u_i = \frac{1}{\rho}(\beta + 1 - s_i),
$$

which holds when $s_i > \beta + 1$.

$-0 < u_i < w_i$:

$$
0 = 1 - \beta + s_i + \rho u_i \iff u_i = \frac{1}{\rho}(\beta - 1 - s_i),
$$

which holds when $-\rho w_i + \beta - 1 < s_i < \beta - 1$.

$-u_i > w_i$:

$$
0 = 1 + \beta + s_i + \rho u_i \iff u_i = \frac{1}{\rho}(-\beta - 1 - s_i),
$$

which holds when $s_i < -\rho w_i - \beta - 1$.

We then have, for $w_i > 0$:

$$u_i^{k+1} = \begin{cases} \frac{1}{\rho}\left(-\beta - 1 - s_i\right) & , s_i < -\rho w_i - \beta - 1 \\ w_i & , -\rho w_i - \beta - 1 \leq s_i \leq -\rho w_i + \beta - 1 \\ \frac{1}{\rho}\left(\beta - 1 - s_i\right) & , -\rho w_i + \beta - 1 < s_i < \beta - 1 \\ 0 & , \beta - 1 \leq s_i \leq \beta + 1 \\ \frac{1}{\rho}\left(\beta + 1 - s_i\right) & , s_i > \beta + 1. \end{cases}$$

• $w_i < 0$:

$-\, u_i < w_i$: the optimality conditions are

$$0 = -1 - \beta + s_i + \rho u_i \quad \iff \quad u_i = \frac{1}{\rho}\left(\beta + 1 - s_i\right),$$

which holds when $s_i > -\rho w_i + \beta + 1$.

$-\, w_i < u_i < 0$:

$$0 = -1 + \beta + s_i + \rho u_i \quad \iff \quad u_i = \frac{1}{\rho}\left(-\beta + 1 - s_i\right),$$

which holds when $-\beta + 1 < s_i < -\rho w_i - \beta + 1$.

$-\, u_i > 0$:

$$0 = 1 + \beta + s_i + \rho u_i \quad \iff \quad u_i = \frac{1}{\rho}\left(-\beta - 1 - s_i\right),$$

which holds when $s_i < -\beta - 1$.

We then have that for $w_i < 0$

$$u_i^{k+1} = \begin{cases} \frac{1}{\rho}\left(-\beta - 1 - s_i\right) & , s_i < -\beta - 1 \\ 0 & , -\beta - 1 \leq s_i \leq -\beta + 1 \\ \frac{1}{\rho}\left(-\beta + 1 - s_i\right) & , -\beta + 1 < s_i < -\rho w_i - \beta + 1 \\ w_i & , -\rho w_i - \beta + 1 \leq s_i \leq -\rho w_i + \beta + 1 \\ \frac{1}{\rho}\left(\beta + 1 - s_i\right) & , s_i > -\rho w_i + \beta + 1. \end{cases}$$

2) ***Problem in x:*** Problem (18) is equivalent to

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{\rho}{2}\|x\|_2^2 - \rho {v^k}^T x + {\mu^k}^T x \\ \text{subject to} & b = Ax. \end{array}$$

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x - \frac{1}{\rho}(-\mu^k + \rho v^k)\|_2^2 \\ \text{subject to} & b = Ax. \end{array}$$

Defining p $:= \frac{1}{\rho}(-\mu^k + \rho v^k)$ this is equivalent to projecting a point onto $\{x : Ax = b\}$:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}\|x - p\|_2^2 \\ \text{subject to} & b = Ax, \end{array}$$

which has the closed-form solution

$$x^{k+1} = p - A^T(AA^T)^{-1}(Ap - b). \tag{23}$$

Since A is give in implicit form i.e., we just have access to operations Ax and A^Ty. We consider 3 cases of the linear operator A: bicubic interpolation, box averaging and subsampling.

Bicubic interpolation

Most super-resolution (SR) CNNs use bicubic interpolation to obtain the LR images where Ax represents the LR image obtained from bicubic interpolation equivalent to the Matlab function *imresize*. In this case, we solve (23) via the conjugate gradient (CG) method. Solving the following for c,

$$(AA^T)c = Ap - b,$$

we find $(AA^T)^{-1}(Ap - b)$. Once we know c, x is found by solving

$$x^{k+1} = p - A^Tc. \tag{24}$$

Box averaging and subsampling

In the case of subsampling A is an identity sub-matrix thus, $AA^T = I_m$. Thus, we just need to solve the following equation

$$x^{k+1} = p - A^T(I_m)^{-1}(Ap - b). \tag{25}$$

Since $(I_m)^{-1} = (I_m)$ and any matrix multiplied by the identity remains unchanged, (25) can be written as

$$x^{k+1} = p - A^T(Ap - b). \tag{26}$$

On the other hand, for box averaging, $AA^T = (I_m) * (s.f.)^2$ where s.f. is the scaling factor. Thus, (25) can be written as

$$x^{k+1} = p - (\frac{1}{s.f.^2})A^T(Ap - b). \tag{27}$$

This formulation executes much faster than the case of bicubic interpolation. Note that any other A can vary from the scenarios considered.

2.3.2 Problem in v

$$v^{k+1} = \arg\min_v \quad -\lambda^{k^T}Dv - \mu^{k^T}v + \tfrac{\rho}{2}\|u^{k+1} - Dv\|_2^2 + \tfrac{\rho}{2}\|x^{k+1} - v\|_2^2. \tag{28}$$

Using the first-order optimiality condition on (28) we obtain:

$$-D^T\lambda^k - \mu^k + \rho D^T(Dv - u^{k+1}) - \rho(x^{k+1} - v) = 0$$

$$\rho D^TDv + \rho v = \rho x^{k+1} + D^T\lambda^k + \mu^k + \rho D^Tu^{k+1}$$

$$(D^TD + I)v = D^T(u^{k+1} + \frac{1}{\rho}\lambda^k) + \frac{1}{\rho}\mu^k + x^{k+1}. \tag{29}$$

To solve (29), we first compute the right hand side followed by the inverse of $(D^T D + I)$ where

$$D = \begin{bmatrix} D_v \\ D_h \end{bmatrix}. \tag{30}$$

Let $g = D^T(u^{k+1} + \frac{1}{\rho}\lambda^k) + \frac{1}{\rho}\mu^k - x^{k+1}$ then we need to solve:

$$(D^T D + I)v = g.$$

Recall that matrix D is circulant and diagonalizable by the FFT,

$$D_h = F_n^H \text{Diag}\,(F_n c_h)\, F_n \tag{31}$$

$$D_v = F_n^H \text{Diag}\,(F_n c_v)\, F_n, \tag{32}$$

where $F_n^H = F_n^{-1}$ represents the complex conjugate/inverse of F_n.

In this case, we can find a closed form solution. Since the D matrix is made up of two circulant matrices, it will be circulant itself. This inverse can be efficiently computed as follows:

$$D^T D + I_n = \begin{bmatrix} D_v^T D_h^T \end{bmatrix} \begin{bmatrix} D_v \\ D_h \end{bmatrix} + I_n \tag{33}$$

$$D^T D + I_n = D_v^T D_v + D_h^T D_h + I_n. \tag{34}$$

By substituting the definitions of D_v and D_h we obtain:

$$D^T D + I_n = \left(F_n^H \text{Diag}\,(F_n c_h)\, F_n\right) \left(F_n^H \text{Diag}\,(F_n c_h)\, F_n\right) + \left(F_n^H \text{Diag}\,(F_n c_v)\, F_n\right) \left(F_n^H \text{Diag}\,(F_n c_v)\, F_n\right) + I_n,$$

since F_n is a unitary matrix, $F_n^T F_n = F_n F_n^T = I_n$, thus

$$D^T D + I_n = \left(F_n^H \text{Diag}\,(F_n c_h)\, \text{Diag}\,(F_n c_h)\, F_n\right) + \left(F_n^H \text{Diag}\,(F_n c_v)\, \text{Diag}\,(F_n c_v)\, F_n\right) + I_n$$

$$D^T D + I_n = F_n^H \left(\text{Diag}(F_n c_h)\text{Diag}(F_n c_h) + \text{Diag}\,(F_n c_v)\, \text{Diag}\,(F_n c_v)\right) F_n + F_n^H F_n,$$

and the inverse is

$$(D^T D + I_n)^{-1} = F_n^H \left(\text{Diag}(F_n c_h)\text{Diag}(F_n c_h) + \text{Diag}\,(F_n c_v)\, \text{Diag}\,(F_n c_v) + I_n\right)^{-1} F_n.$$

Let $h = \text{Diag}(F_n c_h)\text{Diag}(F_n c_h) + \text{Diag}\,(F_n c_v)\, \text{Diag}\,(F_n c_v) + I_n$. Then, v can be found by solving

$$v^{k+1} = \text{ifft}(h \odot \text{fft}(g)).$$

2.3.3 Primal Residual

The primal residual is:

$$r^{k+1} = \begin{bmatrix} u^{k+1} \\ x^{k+1} \end{bmatrix} + \begin{bmatrix} -Dv^{k+1} \\ -v^{k+1} \end{bmatrix}. \tag{35}$$

2.3.4 Dual Residual

The dual residual is:

$$s^{k+1} = -\begin{bmatrix} D(v^{k+1} + v^k) \\ v^{k+1} + v^k \end{bmatrix}. \tag{36}$$

2.3.5 Update of Lagrange Multipliers

$$\lambda^{k+1} = \lambda^k + \rho(u^{k+1} - Dv^{k+1}) \tag{37}$$

$$\mu^{k+1} = \mu^k + \rho(x^{k+1} - v^{k+1}). \tag{38}$$

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[2] L. Liu, S. H. Chan, and T. Q. Nguyen. Depth reconstruction from sparse samples: Representation, algorithm, and sampling. *IEEE Transactions on Image Processing*, 24(6):1983–1996, 2015-06.

[3] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues. Compressed sensing with prior information: Strategies, geometry, and bounds. *IEEE Transactions on Information Theory*, 63(7):4472–4496, 2017.