Split Bregman Image Restoration

Yuji Oyamada [1]

[1] HVRL, Keio University

January 20, 2012

I wrote this document when I tried to understand Split Bregman Iteration for image restoration. Practical examples written in Sec. 4, 5, 6 are inspired by Gilles [2011].

1 Linear image model

Suppose a linear measuring system

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{f} \in \mathbb{R}^n$ denotes an observed image, $\boldsymbol{u} \in \mathbb{R}^m$ denotes an unknown original image, $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ represents a measurement matrix, and $\boldsymbol{n} \in \mathbb{R}^n$ denotes image noise.

When $n < m$, the system is under-determined meaning there is no unique answer. Due to the under-determinedness, solving the system requires additional queues, *i.e.*, reqularizers. Our task is to find $\boldsymbol{u}$ by solving the minimization problem as

$$\boldsymbol{u} = \arg \min_{\boldsymbol{u}} \quad J(\boldsymbol{u}) + H(\boldsymbol{u}) \tag{2}$$

where $H(\boldsymbol{u})$ is the data fidelity term and $J(\boldsymbol{u})$ is the regularization term.

2 Sparsity-based image restoration

One well-used prior knowledge is that the original signal $\boldsymbol{u}$ can be represented by a sparse vector $\boldsymbol{d} = \Theta(\boldsymbol{u})$ in a certain domain. When the number of non-zero entries is less than the length of the observation $\boldsymbol{f}$, the system can be regarded as an over-determined system.

Under certain conditions, such problems are known to be solvable by l_1 minimization. Simple unconstrained l_1 minimization is formulated as

$$\begin{aligned} &\boldsymbol{u} = \arg \min_{\boldsymbol{u}} \quad J(\boldsymbol{u}) + H(\boldsymbol{u}) \\ &J(\boldsymbol{u}) = \|\boldsymbol{u}\|_1 \quad \text{or} \quad \|\Theta(\boldsymbol{u})\|_1 \\ &H(\boldsymbol{u}) = \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \end{aligned} \tag{3}$$

for $\mu > 0$.

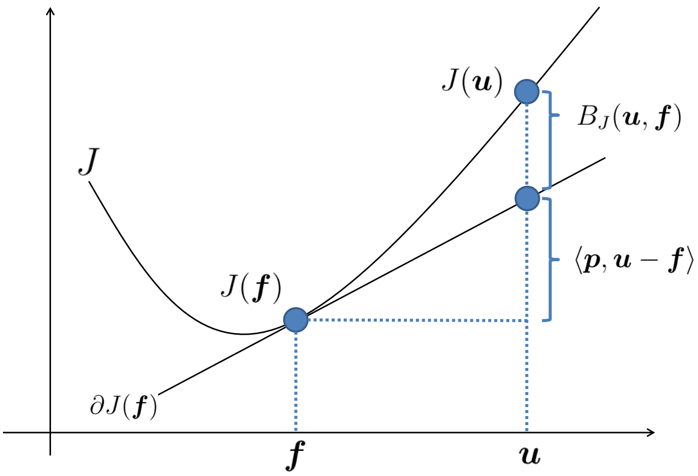

Figure 1: Bregman distance

2.1 Sparse image model

2.2 Sparsity-based image restoration

3 Split Bregman image restoration

3.1 Bregman distance

The Bregman distance is similar to a metric but does not satisfy the triangle inequality nor symmetryBregman [1967]. The Bregman distance is associated with a convex function J between points $\boldsymbol{u}$ and $\boldsymbol{f}$ is defined as

$$B_J(\boldsymbol{u}, \boldsymbol{f}) = J(\boldsymbol{u}) - J(\boldsymbol{f}) - \langle \boldsymbol{p}, \boldsymbol{u} - \boldsymbol{f} \rangle, \tag{4}$$

where $\boldsymbol{p} \in \partial J(\boldsymbol{f})$ is an element in the sub-gradient of J at the point $\boldsymbol{f}$. This can be thought of as the difference between the value of J an point u ($J(\boldsymbol{u})$) and the value of the first-order Taylor expansion of J around point $\boldsymbol{f}$ evaluated at point $\boldsymbol{u}$ ($J(\boldsymbol{f}) + \langle \boldsymbol{p}, \boldsymbol{u} - \boldsymbol{f} \rangle$). Figure 3.1 depicts the Bregman distance.

3.1.1 Properties

The Bregman distance has following properties:

Non-negativity $B_J(\boldsymbol{u}, \boldsymbol{u}^k) \geq 0$ for all $\boldsymbol{u}$, $\boldsymbol{f}$. This is a consequence of the convexity of J.

Convexity $B_J(\boldsymbol{u}, \boldsymbol{f})$ is convex in its first argument, but not necessarily in the second argument.

Linearity If we think of the Bregman distance as an operator on the function J, then it is linear w.r.t. non-negative coefficients. In other words, for J_1, J_2 strictly convex and differentiable, and $\lambda > 0$,

$$B_{J_1 + \lambda J_2}(\boldsymbol{u}, \boldsymbol{f}) = B_{J_1}(\boldsymbol{u}, \boldsymbol{f}) + \lambda B_{J_2}(\boldsymbol{u}, \boldsymbol{f}). \tag{5}$$

Duality The function J has a convex conjugate J^*. The Bregman distance defined w.r.t. J^* has an interesting relationship to $B_J(\boldsymbol{u}, \boldsymbol{f})$ as

$$B_{J^*}(\nabla J(\boldsymbol{u}), \nabla J(\boldsymbol{f})) = B_J(\boldsymbol{f}, \boldsymbol{u}) \tag{6}$$

A key result about Bregman distance is that, given a random vector, the mean vector minimizes the expected Bregman distance from the random vector. This result generalizes the textbook result that the mean of a set minimizes total squared error to elements in the set. This result was proved for the vector case and extended to the case of functions/distributions. This result is important because it further justifies using a mean as a representative of a random set, particularly in Bayesian estimation.

3.1.2 Examples

- Squared Euclidean distance $B_J(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|^2$ is the canonical example of a Bregman distance, generated by the convex function $J(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$.
- The squared Mahalanobis distance $B_J(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{y})^{\mathbf{T}} Q(\boldsymbol{x} - \boldsymbol{y})$ which is generated by the convex function $J(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathbf{T}} Q\boldsymbol{x}$. This can be thought of as a generalization of the above squared Euclidean distance.
- The generalized Kullback-Leibler divergence $B_J(\boldsymbol{p}, \boldsymbol{q}) = \sum \boldsymbol{p}(i) \log \frac{\boldsymbol{p}(i)}{\boldsymbol{q}(i)} - \sum \boldsymbol{p}(i) + \sum \boldsymbol{q}(i)$ is generated by the convex function $J(\boldsymbol{p}) = \sum \boldsymbol{p}(i) \log \boldsymbol{p}(i) - \sum \boldsymbol{p}(i)$.
- The Itakura-Saito distance $B_J(\boldsymbol{p}, \boldsymbol{q}) = \sum_i \left(\frac{\boldsymbol{p}(i)}{\boldsymbol{q}(i)} - \log \frac{\boldsymbol{p}(i)}{\boldsymbol{q}(i)} - 1\right)$ is generated by the convex function $J(\boldsymbol{p}) = -\sum \log \boldsymbol{p}(i)$.

3.2 Bregman image restoration

The Bregman image restoration iteratively solves Eq. (3) based on Bregman distance measure as

$$\boldsymbol{u}^{k+1} = \arg\ \min_{\boldsymbol{u}} B_J(\boldsymbol{u}, \boldsymbol{u}^k) + \frac{\lambda}{2} H(\boldsymbol{u}). \tag{7}$$

From Eq. (4), Eq. (7) is formulated as

$$\begin{cases} \boldsymbol{u}^{k+1} = \arg\ \min_{\boldsymbol{u}} J(\boldsymbol{u}) - J(\boldsymbol{u}^k) - \langle \boldsymbol{p}^k, \boldsymbol{u} - \boldsymbol{u}^k \rangle + \frac{\lambda}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \\ \boldsymbol{p}^{k+1} = \boldsymbol{p}^k + \lambda \boldsymbol{A}^{\mathbf{T}}(\boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}^{k+1}). \end{cases} \tag{8}$$

Omitting the constant variables, the solution $\boldsymbol{u}^{k+1}$ is obtained as

$$\begin{aligned} \boldsymbol{u}^{k+1} &= \arg\ \min_{\boldsymbol{u}} J(\boldsymbol{u}) - J(\boldsymbol{u}^k) - \langle \boldsymbol{p}^k, \boldsymbol{u} - \boldsymbol{u}^k \rangle + \frac{\lambda}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \\ &= \arg\ \min_{\boldsymbol{u}} J(\boldsymbol{u}) - \langle \boldsymbol{p}^k, \boldsymbol{u} \rangle + \frac{\lambda}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 - J(\boldsymbol{u}^k) + \langle \boldsymbol{p}^k, \boldsymbol{u}^k \rangle \\ &\to \arg\ \min_{\boldsymbol{u}} J(\boldsymbol{u}) - \langle \boldsymbol{p}^k, \boldsymbol{u} \rangle + \frac{\lambda}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \end{aligned} \tag{9}$$

The difference between Eq. (3) and Eq. (7) is in the use of regularization. Equation Eq. (7) regularizes $\boldsymbol{u}$ by minimizing the Bregman distance of $J(\boldsymbol{u})$ to a previous solution $\boldsymbol{u}^k$ while Eq. (3) regularizes $\boldsymbol{u}$ by directly minimizing $J(\boldsymbol{u})$.

Two key results for the sequence $\{\boldsymbol{u}^k\}$ obtained by Eq. (7). First, $\left\|\boldsymbol{u}^k - \boldsymbol{b}\right\|$ converges to $\mathbf{0}$ monotonically; second, $\boldsymbol{u}^k$ also gets closer to $\boldsymbol{u}$, the unknown noiseless image, monotonically in terms of the Bregman distance, at least while $\left\|\boldsymbol{u}^k - \boldsymbol{b}\right\| \geq \|\boldsymbol{u} - \boldsymbol{b}\|$.

Interestingly, not only for the first iteration $k = 0$ but for all k, the new problem Eq. (7) can be reduced to the original problem Eq. (3) with the input $\boldsymbol{b}^{k+1} := \boldsymbol{b} + (\boldsymbol{b}^k - \boldsymbol{u}^k)$ starting with $\boldsymbol{b}^0 = \boldsymbol{u}^0 = \mathbf{0}$; *i.e.*, the iterations Eq. (7) are equivalent to

$$\boldsymbol{u}^{k+1} \leftarrow \min_{\boldsymbol{u}} \; B_J + \frac{1}{2}\left\|\boldsymbol{u} - \boldsymbol{b}^{k+1}\right\|_2^2, \quad \text{where} \;\; \boldsymbol{b}^{k+1} = \boldsymbol{b} + (\boldsymbol{b}^k - \boldsymbol{u}^k). \tag{10}$$

The algorithm of the Bregman image restoration is given in Alg. 1.

Algorithm 1 Bregman image restoration

Task: Recover the original signal $\boldsymbol{u}$.
Input: Observation signal $\boldsymbol{f}$ and measuring matrix $\boldsymbol{A}$.
Initialization: set
$\boldsymbol{u}^0 = 0$
$\boldsymbol{p}^0 = 0$
while until convergence **do**
$\boldsymbol{u}^{k+1} = \arg\min_{\boldsymbol{u}} J(\boldsymbol{u}) - \langle \boldsymbol{p}^k, \boldsymbol{u}\rangle + \frac{\lambda}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2$
$\boldsymbol{p}^{k+1} = \boldsymbol{p}^k - \boldsymbol{A}^{\mathbf{T}}(\boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}^{k+1})$
end while
Output: Recovered signal $\boldsymbol{u} = \boldsymbol{u}^k$.

3.3 Adding back the residual

The iterative procedure Eq. (10) has an intriguing interpretation: Let $\boldsymbol{w}$ represent the noise in $\boldsymbol{b}$, *i.e.*, $\boldsymbol{b} = \boldsymbol{u} + \boldsymbol{w}$, and let μ be large enough. At $k = 0$, $\boldsymbol{b}^k - \boldsymbol{u}^k = \mathbf{0}$, so Eq. (10) decomposes the input noisy image $\boldsymbol{b}$ into $\boldsymbol{u}^1 + \boldsymbol{v}^1$. The residual $\boldsymbol{v}^1 = \boldsymbol{b} - \boldsymbol{u}^1 = (\boldsymbol{u} - \boldsymbol{u}^1) + \boldsymbol{w}$, hence, is the sum of the un-recovered good signal $(\boldsymbol{u} - \boldsymbol{u}^1)$ and the bad noise $\boldsymbol{w}$. **Adding back the residual** algorithm turns out to be both better and non-intuitive. The algorithm adds the residual $\boldsymbol{v}^1$ back to the original input $\boldsymbol{b}$. Thus, the new input of Eq. (10) in the second iteration is

$$\boldsymbol{b} + \boldsymbol{v}^1 = (\boldsymbol{u}^1 + \boldsymbol{v}^1) + \boldsymbol{v}^1 = \boldsymbol{u}^1 + 2(\boldsymbol{u} - \boldsymbol{u}^1) + 2\boldsymbol{w}. \tag{11}$$

Compared to the original input $\boldsymbol{b}$, the new input contains twice as much of both the un-recovered good signal and the bad noise. As a result, the new decomposition $\boldsymbol{u}^2$ is a better approximation of $\boldsymbol{u}$ than $\boldsymbol{u}^1$. The adding back the residual algorithm is given in Alg. 2.

Algorithm 2 Bregman image restoration: Adding back the residual

Task: Recover the original signal $\hat{\boldsymbol{u}}$.
Input: Observation signal $\boldsymbol{f}$ and measuring matrix $\boldsymbol{A}$.
Initialization: set
$\boldsymbol{u}^0 = 0$
$\boldsymbol{f}^0 = 0$
while until convergence **do**
$\boldsymbol{f}^{k+1} = \boldsymbol{f}^k + (\boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}^k)$
$\boldsymbol{u}^{k+1} = \arg\min_{\boldsymbol{u}} J(\boldsymbol{u}) + \frac{\lambda}{2}\left\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}^{k+1}\right\|_2^2$
end while
Output: Recovered signal $\hat{\boldsymbol{u}} = \boldsymbol{u}^k$.

3.4 Split Bregman image restoration

When the optimization function is differentiable as

$$\hat{\boldsymbol{u}} = \arg \min_{\boldsymbol{u}} H(\boldsymbol{u}) = \arg \min_{\boldsymbol{u}} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 , \tag{12}$$

or is solvable by shrinkage algorithm as

$$\hat{\boldsymbol{u}} = \arg \min_{\boldsymbol{u}} J(\boldsymbol{u}) + H(\boldsymbol{u}) = \arg \min_{\boldsymbol{u}} \|\Phi(\boldsymbol{u})\|_1 + \|\boldsymbol{u} - \boldsymbol{f}\|_2^2 , \tag{13}$$

the problems are easy to solve. However, when the optimization function couples l_1 and l_2 terms as

$$\hat{\boldsymbol{u}} = \arg \min_{\boldsymbol{u}} J(\boldsymbol{u}) + H(\boldsymbol{u}) = \arg \min_{\boldsymbol{u}} \|\Phi(\boldsymbol{u})\|_1 + \frac{\mu}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 , \tag{14}$$

the problem is hard to solve.

The Split Bregman Iteration Goldstein and Osher [2009] splits the problem (Eq. (14)), into l_1 and l_2 components as

$$\begin{gathered}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{d}}) = \arg \min_{\boldsymbol{u},\boldsymbol{d}} J(\boldsymbol{d}) + H(\boldsymbol{u}) \\ \text{subject to } \boldsymbol{d} = \Phi(\boldsymbol{u}),\end{gathered} \tag{15}$$

where $J(\boldsymbol{d}) = \|\boldsymbol{d}\|_1$. With a Lagrange multiplier λ, we obtain the following minimization problem:

$$\begin{aligned}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{d}}) &= \arg \min_{\boldsymbol{u},\boldsymbol{d}} J(\boldsymbol{d}) + H_{\boldsymbol{u}}(\boldsymbol{u}) + H_{\boldsymbol{d}}(\boldsymbol{d}) \\ &= \arg \min_{\boldsymbol{u},\boldsymbol{d}} \|\boldsymbol{d}\|_1 + \frac{\mu}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2} \|\boldsymbol{d} - \Phi(\boldsymbol{u})\|_2^2 \\ &= \arg \min_{\boldsymbol{u},\boldsymbol{d}} E(\boldsymbol{u}, \boldsymbol{d}) + \frac{\lambda}{2} \|\boldsymbol{d} - \Phi(\boldsymbol{u})\|_2^2 ,\end{aligned} \tag{16}$$

where

$$E(\boldsymbol{u}, \boldsymbol{d}) = J(\boldsymbol{d}) + H_{\boldsymbol{u}}(\boldsymbol{u}) = \|\boldsymbol{d}\|_1 + \frac{\mu}{2} \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 . \tag{17}$$

To enforce the constraint condition, we now plug this problem into the Bregman formulation as

$$(\boldsymbol{u}^{k+1}, \boldsymbol{d}^{k+1}) = \arg \min_{\boldsymbol{u},\boldsymbol{d}} B(\boldsymbol{u}, \boldsymbol{u}^k, \boldsymbol{d}, \boldsymbol{d}^k) + \frac{\lambda}{2} \|\boldsymbol{d} - \Phi(\boldsymbol{u})\|_2^2 , \tag{18}$$

where

$$B(\boldsymbol{u}, \boldsymbol{u}^k, \boldsymbol{d}, \boldsymbol{d}^k) = E(\boldsymbol{u}, \boldsymbol{d}) - \langle \boldsymbol{p}_u^k, \boldsymbol{u} - \boldsymbol{u}^k \rangle + \langle \boldsymbol{p}_d^k, \boldsymbol{d} - \boldsymbol{d}^k \rangle. \tag{19}$$

By simplifying the equations, we get the following two line algorithm

$$\begin{aligned}&(\boldsymbol{u}^{k+1}, \boldsymbol{d}^{k+1}) = \arg \min_{\boldsymbol{u},\boldsymbol{d}} E(\boldsymbol{u}, \boldsymbol{d}) + \frac{\lambda}{2} \left\|\boldsymbol{d} - \Phi(\boldsymbol{u}) - \boldsymbol{b}^k\right\|_2^2 \\ &\boldsymbol{b}^{k+1} = \boldsymbol{b}^k - (\Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{d}^{k+1}).\end{aligned} \tag{20}$$

Thus, the update of each variable is written as

$$
\begin{cases}
\boldsymbol{u}^{k+1} & = \arg \min\limits_{\boldsymbol{u}} H_{\boldsymbol{u}}(\boldsymbol{u}) + \frac{\lambda}{2} \left\| \boldsymbol{d}^k - \Phi(\boldsymbol{u}) - \boldsymbol{b}^k \right\|_2^2, \\
 & = \arg \min\limits_{\boldsymbol{u}} \| \boldsymbol{A}\boldsymbol{u} - \boldsymbol{f} \|_2^2 + \frac{\lambda}{2} \left\| \boldsymbol{d}^k - \Phi(\boldsymbol{u}) - \boldsymbol{b}^k \right\|_2^2, \\
\boldsymbol{d}^{k+1} & = \arg \min\limits_{\boldsymbol{d}} J(\boldsymbol{d}) + \frac{\lambda}{2} \left\| \boldsymbol{d} - \Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{b}^k \right\|_2^2, \\
 & = \arg \min\limits_{\boldsymbol{d}} \| \boldsymbol{d} \|_1 + \frac{\lambda}{2} \left\| \boldsymbol{d} - \Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{b}^k \right\|_2^2, \\
\boldsymbol{b}^{k+1} & = \boldsymbol{b}^k + (\Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{d}^{k+1}).
\end{cases}
\tag{21}
$$

Since the update of $\boldsymbol{u}$ is differentiable, we can directly solve it with Gauss-Seidel, Conjugate Gradient, *etc*. The update of $\boldsymbol{d}$ is solvable by shrinkage algorithm and one of $\boldsymbol{b}$ is explicit. The algorithm of the split Bregman image restoration is given in Alg. 3.

Algorithm 3 Split Bregman image restoration

Task: Recover the original signal $\hat{\boldsymbol{u}}$.
Input: Observation signal $\boldsymbol{f}$ and measuring matrix $\boldsymbol{A}$.
  Initialization: Set $\boldsymbol{d}^0 = \boldsymbol{0}$ and $\boldsymbol{b}^0 = \boldsymbol{0}$.
  while until convergence **do**
    $\boldsymbol{u}^{k+1} = \arg \min\limits_{\boldsymbol{u}} \| \boldsymbol{A}\boldsymbol{u} - \boldsymbol{f} \|_2^2 + \frac{\lambda}{2} \left\| \boldsymbol{d}^k - \Phi(\boldsymbol{u}) - \boldsymbol{b}^k \right\|_2^2$
    $\boldsymbol{d}^{k+1} = \arg \min\limits_{\boldsymbol{d}} \| \boldsymbol{d} \|_1 + \frac{\lambda}{2} \left\| \boldsymbol{d} - \Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{b}^k \right\|_2^2$
    $\boldsymbol{b}^{k+1} = \boldsymbol{b}^k + (\Phi(\boldsymbol{u}^{k+1}) - \boldsymbol{d}^{k+1})$
  end while
Output: Recovered signal $\hat{\boldsymbol{u}} = \boldsymbol{u}^k$.

Algorithm 4 Sparse signal recovery by Split Bregman Iteration

Task: Recover the original signal $\boldsymbol{u}$.
Input: Observation signal $\boldsymbol{f}$ and measuring matrix $\boldsymbol{A}$.
  Initialization: Set $\boldsymbol{d}^0 = \mathbf{0}$ and $\boldsymbol{b}^0 = \mathbf{0}$.
  while until convergence **do**
    $\boldsymbol{u}^{k+1} = (\mu \boldsymbol{A}^{\mathbf{T}}\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}(\mu \boldsymbol{A}^{\mathbf{T}}\boldsymbol{f} + \lambda(\boldsymbol{d}^k - \boldsymbol{b}^k))$
    $\boldsymbol{d}^{k+1} = \mathrm{Shrink}(\boldsymbol{u}^{k+1} + \boldsymbol{b}^k, \frac{1}{\lambda})$
    $\boldsymbol{b}^{k+1} = \boldsymbol{b}^k + (\boldsymbol{u}^{k+1} - \boldsymbol{d}^{k+1})$
  end while
Output: Recovered signal $\boldsymbol{u} = \boldsymbol{f}$.

4 Sparse Signal Recovery

The purpose is to recover a sparse signal $\boldsymbol{u}$ from its observation $\boldsymbol{f}$. Here, we assume that $\boldsymbol{u}$ is altered by a known linear operator $\boldsymbol{A}$. By setting

$$\begin{aligned} &\boldsymbol{D} = \boldsymbol{I} \text{ (Identity matrix) },\\ &\Theta(\boldsymbol{d}) = \|\boldsymbol{d}\|_1 ,\\ &\Phi_{\boldsymbol{u}}(\boldsymbol{u}) = \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 , \end{aligned} \tag{22}$$

the variable update of Split Bregman Iteration is written as

$$\begin{cases} \boldsymbol{u}^{k+1} = \arg\ \min\limits_{\boldsymbol{u}} \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}^k - \boldsymbol{u} - \boldsymbol{b}^k\right\|_2^2 ,\\ \boldsymbol{d}^{k+1} = \arg\ \min\limits_{\boldsymbol{d}} \|\boldsymbol{d}\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d} - \boldsymbol{u}^{k+1} - \boldsymbol{b}^k\right\|_2^2 ,\\ \boldsymbol{b}^{k+1} = \boldsymbol{b}^k + (\boldsymbol{u}^{k+1} - \boldsymbol{d}^{k+1}). \end{cases} \tag{23}$$

To solve $\boldsymbol{u}^{k+1}$,

$\boldsymbol{u}^{k+1}$ update: Find $\boldsymbol{u}$ that sets the gradient zero as

$$\begin{aligned} &\partial_{\boldsymbol{u}}\left(\frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}^k - \boldsymbol{u} - \boldsymbol{b}^k\right\|_2^2\right) = 0\\ \to &\mu \boldsymbol{A}^{\mathbf{T}}(\boldsymbol{A}\boldsymbol{u}^{k+1} - \boldsymbol{f}) - \lambda(\boldsymbol{d}^k - \boldsymbol{u}^{k+1} - \boldsymbol{b}^k) = 0\\ \to &\boldsymbol{u}^{k+1} = (\mu \boldsymbol{A}^{\mathbf{T}}\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}(\mu \boldsymbol{A}^{\mathbf{T}}\boldsymbol{f} + \lambda(\boldsymbol{d}^k - \boldsymbol{b}^k)) \end{aligned} \tag{24}$$

$\boldsymbol{d}^{k+1}$ update: Since the problem is l_1 minimization problem, shrinkage algorithm is applicable as

$$\boldsymbol{d}^{k+1} = \mathrm{Shrink}(\boldsymbol{u}^{k+1} + \boldsymbol{b}^k, \frac{1}{\lambda}), \tag{25}$$

where the component-wise shrinkage operator is applied on each component i of the vector:

$$\mathrm{Shrink}(\boldsymbol{u}_i, \delta) = \mathrm{sign}(\boldsymbol{u}_i)\max(0, |\boldsymbol{u}_i| - \delta). \tag{26}$$

The algorithm of the Split Bregman Iteration is given in Alg. 4.

5 ROF denoising

5.1 Anisotropic case

The purpose is to recover an unknown image $\boldsymbol{u}$ from its noisy observation $\boldsymbol{f}$. Here, the relationship of the images are $\boldsymbol{f} = \boldsymbol{u} + \boldsymbol{e}$. The anisotropic denoising Rudin-Osher-Fatemi model is formulated as

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \|\nabla_x \boldsymbol{u}\|_1 + \|\nabla_y \boldsymbol{u}\|_1 + \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{f}\|_2^2. \tag{27}$$

By setting

$$\begin{aligned} &\boldsymbol{d} = \boldsymbol{D}\boldsymbol{u} = \boldsymbol{d}_x + \boldsymbol{d}_y = \nabla_x \boldsymbol{u} + \nabla_y \boldsymbol{u}, \\ &\Theta(\boldsymbol{d}) = \|\boldsymbol{d}_x\|_1 + \|\boldsymbol{d}_y\|_1, \\ &\Phi_{\boldsymbol{u}}(\boldsymbol{u}) = \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{f}\|_2^2, \end{aligned} \tag{28}$$

the variable update of Split Bregman Iteration is written as

$$\begin{cases} \boldsymbol{u}^{k+1} = \arg\min_{\boldsymbol{u}} \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u} - \boldsymbol{b}_x^k\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u} - \boldsymbol{b}_y^k\right\|_2^2, \\ \boldsymbol{d}_x^{k+1} = \arg\min_{\boldsymbol{d}_x} \|\boldsymbol{d}_x\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x - \nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{b}_x^k\right\|_2^2, \\ \boldsymbol{d}_y^{k+1} = \arg\min_{\boldsymbol{d}_y} \|\boldsymbol{d}_y\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y - \nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{b}_y^k\right\|_2^2, \\ \boldsymbol{b}_x^{k+1} = \boldsymbol{b}_x^k + (\nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{d}_x^{k+1}), \\ \boldsymbol{b}_y^{k+1} = \boldsymbol{b}_y^k + (\nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{d}_y^{k+1}). \end{cases} \tag{29}$$

To solve $\boldsymbol{u}^{k+1}$,

$\boldsymbol{u}^{k+1}$ update: Find $\boldsymbol{u}$ that sets the gradient zero as

$$\begin{aligned} &\partial_{\boldsymbol{u}}\left(\frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u} - \boldsymbol{b}_x^k\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u} - \boldsymbol{b}_y^k\right\|_2^2\right) = 0 \\ &\rightarrow \mu(\boldsymbol{u}^{k+1} - \boldsymbol{f}) - \lambda\nabla_x^{\mathbf{T}}(\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{b}_x^k) - \lambda\nabla_y^{\mathbf{T}}(\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{b}_y^k) = 0 \\ &\rightarrow \mu\boldsymbol{u}^{k+1} - \lambda\nabla_x^{\mathbf{T}}\nabla_x \boldsymbol{u}^{k+1} - \lambda\nabla_y^{\mathbf{T}}\nabla_y \boldsymbol{u}^{k+1} = \mu\boldsymbol{f} - \lambda\nabla_x(\boldsymbol{d}_x^k - \boldsymbol{b}_x^k) - \lambda\nabla_y(\boldsymbol{d}_y^k - \boldsymbol{b}_y^k) \\ &\rightarrow \mu\boldsymbol{u}^{k+1} - \lambda\Delta\boldsymbol{u}^{k+1} = \mu\boldsymbol{f} - \lambda\left(\nabla_x(\boldsymbol{d}_x^k - \boldsymbol{b}_x^k) + \nabla_y(\boldsymbol{d}_y^k - \boldsymbol{b}_y^k)\right) \\ &\rightarrow (\mu\boldsymbol{I} - \lambda\Delta)\boldsymbol{u}^{k+1} = \mu\boldsymbol{f} - \lambda \cdot \operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right) \\ &\rightarrow (\mu \cdot \mathcal{F}(\boldsymbol{I}) - \lambda \cdot \mathcal{F}(\Delta))\mathcal{F}(\boldsymbol{u}^{k+1}) = \mu \cdot \mathcal{F}(\boldsymbol{f}) - \lambda \cdot \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right)) \\ &\rightarrow \mathcal{F}(\boldsymbol{u}^{k+1}) = (\mu \cdot \mathcal{F}(\boldsymbol{I}) - \lambda \cdot \mathcal{F}(\Delta))^{-1}\left(\mu \cdot \mathcal{F}(\boldsymbol{f}) - \lambda \cdot \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right))\right) \\ &\rightarrow \boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left((\mu \cdot \mathcal{F}(\boldsymbol{I}) - \lambda \cdot \mathcal{F}(\Delta))^{-1}\left(\mu \cdot \mathcal{F}(\boldsymbol{f}) - \lambda \cdot \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right))\right)\right) \end{aligned} \tag{30}$$

Thus,

$$\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right), \tag{31}$$

where

$$\begin{aligned} \boldsymbol{W} &= (\mu \cdot \mathcal{F}(\boldsymbol{I}) - \lambda \cdot \mathcal{F}(\Delta))^{-1} \\ \boldsymbol{F}^k &= \mu \cdot \mathcal{F}(\boldsymbol{f}) - \lambda \cdot \mathcal{F}\left(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right)\right) \\ &= \mu \cdot \mathcal{F}(\boldsymbol{f}) - \lambda \cdot \mathcal{F}\left(\nabla_x\left(\boldsymbol{d}_x^k - \boldsymbol{b}_x^k\right) + \nabla_y\left(\boldsymbol{d}_y^k - \boldsymbol{b}_y^k\right)\right) \end{aligned} \tag{32}$$

$\boldsymbol{d}^{k+1}$ update: Since the problem is l_1 minimization problem, shrinkage algorithm is applicable as

$$\begin{aligned} \boldsymbol{d}_x^{k+1} &= \text{Shrink}\left(\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k, \frac{1}{\lambda}\right), \\ \boldsymbol{d}_y^{k+1} &= \text{Shrink}\left(\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k, \frac{1}{\lambda}\right). \end{aligned} \tag{33}$$

The algorithm of the Split Bregman Iteration is given in Alg. 5.

Algorithm 5 Anisotropic ROF denoising by Split Bregman Iteration

Task: Recover the noise-free image $\boldsymbol{u}$.
Input: Observed noisy image $\boldsymbol{f}$.
  Initialization: Set $\boldsymbol{d}_x^0 = \boldsymbol{0}$, $\boldsymbol{b}_x^0 = \boldsymbol{0}$, $\boldsymbol{d}_y^0 = \boldsymbol{0}$, and $\boldsymbol{b}_y^0 = \boldsymbol{0}$.
  while until convergence **do**
    $\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right)$
    $\boldsymbol{d}_x^{k+1} = \text{Shrink}\left(\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k, \frac{1}{\lambda}\right)$
    $\boldsymbol{d}_y^{k+1} = \text{Shrink}\left(\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k, \frac{1}{\lambda}\right)$
    $\boldsymbol{b}_x^{k+1} = \boldsymbol{b}_x^k + (\nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{d}_x^{k+1})$
    $\boldsymbol{b}_y^{k+1} = \boldsymbol{b}_y^k + (\nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{d}_y^{k+1})$
  end while
Output: Denoised image $\boldsymbol{u} = \boldsymbol{f}$.

5.2 Isotropic case

Similar to the anisotropic case, we consider the isotropic total variation norm as

$$\boldsymbol{u} = \arg\ \min_{\boldsymbol{u}} \sqrt{\left|\nabla_x \boldsymbol{u}\right|^2 + \left|\nabla_y \boldsymbol{u}\right|^2} + \frac{\mu}{2}\left\|\boldsymbol{u} - \boldsymbol{f}\right\|_2^2. \tag{34}$$

Similar to the anisotropic case, we denote $\boldsymbol{d}_x = \nabla_x \boldsymbol{u}$ and $\boldsymbol{d}_y = \nabla_y \boldsymbol{u}$, then the only difference with the anisotropic case is concerning the minimization w.r.t. $\boldsymbol{d}_x$ and $\boldsymbol{d}_y$. Let $\boldsymbol{s}^k$ be

$$\boldsymbol{s}^{k+1} = \sqrt{\left|\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k\right|^2 + \left|\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k\right|^2}, \tag{35}$$

then $\boldsymbol{d}_x$ and $\boldsymbol{d}_y$ are updated by

$$\begin{aligned} \boldsymbol{d}_x^{k+1} &= \max\left(\boldsymbol{s}^{k+1} - \frac{1}{\lambda}, 0\right)\frac{\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k}{\boldsymbol{s}^{k+1}} \\ \boldsymbol{d}_y^{k+1} &= \max\left(\boldsymbol{s}^{k+1} - \frac{1}{\lambda}, 0\right)\frac{\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k}{\boldsymbol{s}^{k+1}}. \end{aligned} \tag{36}$$

The detail of the algorithm is described in Alg. 6.

Algorithm 6 Isotropic ROF denoising by Split Bregman Iteration

Task: Recover the noise-free image $\boldsymbol{u}$.

Input: Observed noisy image $\boldsymbol{f}$.

Initialization: Set $\boldsymbol{d}_x^0 = \mathbf{0}$, $\boldsymbol{b}_x^0 = \mathbf{0}$, $\boldsymbol{d}_y^0 = \mathbf{0}$, and $\boldsymbol{b}_y^0 = \mathbf{0}$.

while until convergence **do**

$\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right)$

$\boldsymbol{s}^{k+1} = \sqrt{\left|\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k\right|^2 + \left|\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k\right|^2}$

$\boldsymbol{d}_x^{k+1} = \max\left(\boldsymbol{s}^{k+1} - \frac{1}{\lambda}, 0\right) \frac{\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k}{\boldsymbol{s}^{k+1}}$

$\boldsymbol{d}_y^{k+1} = \max\left(\boldsymbol{s}^{k+1} - \frac{1}{\lambda}, 0\right) \frac{\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k}{\boldsymbol{s}^{k+1}}$

$\boldsymbol{b}_x^{k+1} = \boldsymbol{b}_x^k + \left(\nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{d}_x^{k+1}\right)$

$\boldsymbol{b}_y^{k+1} = \boldsymbol{b}_y^k + \left(\nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{d}_y^{k+1}\right)$

end while

Output: Denoised image $\boldsymbol{u} = \boldsymbol{f}$.

6 Non-Blind Deconvolution

6.1 Total Variation

The purpose is to recover an unknown image $\boldsymbol{u}$ from its blurry observation $\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u}$ where $\boldsymbol{K}$ represents blur information. Non-blind TV deconvolution is formulated as

$$\boldsymbol{u} = \arg\ \min_{\boldsymbol{u}} \|\nabla_x \boldsymbol{u}\|_1 + \|\nabla_y \boldsymbol{u}\|_1 + \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2 . \tag{37}$$

By setting

$$\begin{aligned}
&\boldsymbol{d} = \boldsymbol{D}\boldsymbol{u} = \boldsymbol{d}_x + \boldsymbol{d}_y = \nabla_x \boldsymbol{u} + \nabla_y \boldsymbol{u}, \\
&\Theta(\boldsymbol{d}) = \|\boldsymbol{d}_x\|_1 + \|\boldsymbol{d}_y\|_1 , \\
&\Phi_{\boldsymbol{u}}(\boldsymbol{u}) = \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2 ,
\end{aligned} \tag{38}$$

the variable update of Split Bregman Iteration is written as

$$\begin{cases}
\boldsymbol{u}^{k+1} = \arg\ \min\limits_{\boldsymbol{u}} \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u} - \boldsymbol{b}_x^k\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u} - \boldsymbol{b}_y^k\right\|_2^2 , \\
\boldsymbol{d}_x^{k+1} = \arg\ \min\limits_{\boldsymbol{d}_x} \|\boldsymbol{d}_x\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x - \nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{b}_x^k\right\|_2^2 , \\
\boldsymbol{d}_y^{k+1} = \arg\ \min\limits_{\boldsymbol{d}_y} \|\boldsymbol{d}_y\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y - \nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{b}_y^k\right\|_2^2 , \\
\boldsymbol{b}_x^{k+1} = \boldsymbol{b}_x^k + (\nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{d}_x^{k+1}), \\
\boldsymbol{b}_y^{k+1} = \boldsymbol{b}_y^k + (\nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{d}_y^{k+1}).
\end{cases} \tag{39}$$

To solve $\boldsymbol{u}^{k+1}$,

$\boldsymbol{u}^{k+1}$ update: Find $\boldsymbol{u}$ that sets the gradient zero as

$$\begin{aligned}
&\partial_{\boldsymbol{u}}\left(\frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u} - \boldsymbol{b}_x^k\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u} - \boldsymbol{b}_y^k\right\|_2^2\right) = 0 \\
\to&\mu\boldsymbol{K}^{\mathbf{T}}(\boldsymbol{K}\boldsymbol{u}^{k+1} - \boldsymbol{f}) - \lambda\nabla_x^{\mathbf{T}}(\boldsymbol{d}_x^k - \nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{b}_x^k) - \lambda\nabla_y^{\mathbf{T}}(\boldsymbol{d}_y^k - \nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{b}_y^k) = 0 \\
\to&\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{K}\boldsymbol{u}^{k+1} - \lambda\Delta\boldsymbol{u}^{k+1} = \mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{f} - \lambda\left(\nabla_x(\boldsymbol{d}_x^k - \boldsymbol{b}_x^k) + \nabla_y(\boldsymbol{d}_y^k - \boldsymbol{b}_y^k)\right) \\
\to&(\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{K} - \lambda\Delta)\boldsymbol{u}^{k+1} = \mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{f} - \lambda\cdot\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right) \\
\to&\left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K})^2 - \Re(\Delta)\right)\mathcal{F}(\boldsymbol{u}^{k+1}) = \frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f}) - \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right)) \\
\to&\mathcal{F}(\boldsymbol{u}^{k+1}) = \left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K})^2 - \Re(\Delta)\right)^{-1}\left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f}) - \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right))\right) \\
\to&\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K})^2 - \Re(\Delta)\right)^{-1}\left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f}) - \mathcal{F}(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right))\right)\right)
\end{aligned} \tag{40}$$

Thus,

$$\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right), \tag{41}$$

where

$$\begin{aligned}
\boldsymbol{W} &= \left(\frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K})^2 - \Re(\Delta)\right)^{-1} \\
\boldsymbol{F}^k &= \frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f}) - \mathcal{F}\left(\operatorname{div}\left(\boldsymbol{d}^k - \boldsymbol{b}^k\right)\right) \\
&= \frac{\mu}{\lambda}\cdot\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f}) - \mathcal{F}\left(\nabla_x\left(\boldsymbol{d}_x^k - \boldsymbol{b}_x^k\right) + \nabla_y\left(\boldsymbol{d}_y^k - \boldsymbol{b}_y^k\right)\right)
\end{aligned} \tag{42}$$

$\boldsymbol{d}^{k+1}$ update: Since the problem is l_1 minimization problem, shrinkage algorithm is applicable as

$$
\begin{aligned}
\boldsymbol{d}_x^{k+1} &= \text{Shrink}\left(\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k, \frac{1}{\lambda}\right), \\
\boldsymbol{d}_y^{k+1} &= \text{Shrink}\left(\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k, \frac{1}{\lambda}\right).
\end{aligned} \tag{43}
$$

The algorithm of the Split Bregman Iteration is given in Alg. 7.

Algorithm 7 Anisotropic TV non-blind deconvolution by Split Bregman Iteration

Task: Recover the blur-free image $\boldsymbol{u}$.
Input: Observed blurred image $\boldsymbol{f}$.
 Initialization: Set $\boldsymbol{d}_x^0 = \boldsymbol{0}$, $\boldsymbol{b}_x^0 = \boldsymbol{0}$, $\boldsymbol{d}_y^0 = \boldsymbol{0}$, and $\boldsymbol{b}_y^0 = \boldsymbol{0}$.
 while until convergence **do**
  $\boldsymbol{u}^{k+1} = \mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right)$
  $\boldsymbol{d}_x^{k+1} = \text{Shrink}\left(\nabla_x \boldsymbol{u}^{k+1} + \boldsymbol{b}_x^k, \frac{1}{\lambda}\right)$
  $\boldsymbol{d}_y^{k+1} = \text{Shrink}\left(\nabla_y \boldsymbol{u}^{k+1} + \boldsymbol{b}_y^k, \frac{1}{\lambda}\right)$
  $\boldsymbol{b}_x^{k+1} = \boldsymbol{b}_x^k + (\nabla_x \boldsymbol{u}^{k+1} - \boldsymbol{d}_x^{k+1})$
  $\boldsymbol{b}_y^{k+1} = \boldsymbol{b}_y^k + (\nabla_y \boldsymbol{u}^{k+1} - \boldsymbol{d}_y^{k+1})$
 end while
Output: Deblurred image $\boldsymbol{u} = \boldsymbol{f}$.

6.2 Tight Frame

Let $\boldsymbol{D}$ and $\boldsymbol{D}^{\mathbf{T}}$ be a frame decomposition and frame reconstruction operators respectively. Here, we assume tight frame satisfying $\boldsymbol{D}^{\mathbf{T}}\boldsymbol{D} = \boldsymbol{I}$. The corresponding model is

$$
\boldsymbol{u} = \arg\ \min_{\boldsymbol{u}} \|\boldsymbol{D}\boldsymbol{u}\|_1 + \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2. \tag{44}
$$

By setting

$$
\begin{aligned}
&\boldsymbol{d} = \boldsymbol{D}\boldsymbol{u}, \\
&\Theta(\boldsymbol{d}) = \|\boldsymbol{d}\|_1, \\
&\Phi_{\boldsymbol{u}}(\boldsymbol{u}) = \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2,
\end{aligned} \tag{45}
$$

the variable update of Split Bregman Iteration is written as

$$
\begin{cases}
\boldsymbol{u}^{k+1} = \arg\ \min_{\boldsymbol{u}} \frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u} - \boldsymbol{f}\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{d}^k - \boldsymbol{D}\boldsymbol{u} - \boldsymbol{b}^k\right\|_2^2, \\
\boldsymbol{d}^{k+1} = \arg\ \min_{\boldsymbol{d}} \|\boldsymbol{d}\|_1 + \frac{\lambda}{2}\left\|\boldsymbol{d} - \boldsymbol{D}\boldsymbol{u}^{k+1} - \boldsymbol{b}^k\right\|_2^2, \\
\boldsymbol{b}^{k+1} = \boldsymbol{b}^k + (\boldsymbol{D}\boldsymbol{u}^{k+1} - \boldsymbol{d}^{k+1}).
\end{cases} \tag{46}
$$

To solve $\boldsymbol{u}^{k+1}$,

$\boldsymbol{u}^{k+1}$ update: Find $\boldsymbol{u}$ that sets the gradient zero as

$$
\begin{aligned}
&\partial_{\boldsymbol{u}}\left(\frac{\mu}{2}\|\boldsymbol{K}\boldsymbol{u}-\boldsymbol{f}\|_2^2+\frac{\lambda}{2}\left\|\boldsymbol{d}^k-\boldsymbol{D}\boldsymbol{u}-\boldsymbol{b}^k\right\|_2^2\right)=0\\
\rightarrow&\mu\boldsymbol{K}^{\mathbf{T}}(\boldsymbol{K}\boldsymbol{u}^{k+1}-\boldsymbol{f})-\lambda\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{D}\boldsymbol{u}^{k+1}-\boldsymbol{b}^k)=0\\
\rightarrow&\left(\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{K}+\lambda\boldsymbol{I}\right)\boldsymbol{u}^{k+1}-\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{f}-\lambda\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)=0\\
\rightarrow&\left(\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{K}+\lambda\boldsymbol{I}\right)\boldsymbol{u}^{k+1}=\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{f}+\lambda\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)\\
\rightarrow&\boldsymbol{u}^{k+1}=\left(\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{K}+\lambda\boldsymbol{I}\right)^{-1}\left(\mu\boldsymbol{K}^{\mathbf{T}}\boldsymbol{f}+\lambda\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)\right)\\
\rightarrow&\mathcal{F}(\boldsymbol{u}^{k+1})=\left(\mu\left|\mathcal{F}(\boldsymbol{K})\right|^2+\lambda\right)^{-1}\left(\mu\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f})+\lambda\cdot\mathcal{F}\left(\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)\right)\right)\\
\rightarrow&\boldsymbol{u}^{k+1}=\mathcal{F}^{-1}\left(\left(\mu\left|\mathcal{F}(\boldsymbol{K})\right|^2+\lambda\right)^{-1}\left(\mu\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f})+\lambda\cdot\mathcal{F}\left(\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)\right)\right)\right)
\end{aligned}
\tag{47}
$$

Thus,

$$
\boldsymbol{u}^{k+1}=\mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right), \tag{48}
$$

where

$$
\begin{aligned}
\boldsymbol{W}&=\left(\mu\cdot\left|\mathcal{F}(\boldsymbol{K})\right|^2-\lambda\right)^{-1}\\
\boldsymbol{F}^k&=\mu\mathcal{F}(\boldsymbol{K}^{\mathbf{T}})\mathcal{F}(\boldsymbol{f})+\lambda\cdot\mathcal{F}\left(\boldsymbol{D}^{\mathbf{T}}(\boldsymbol{d}^k-\boldsymbol{b}^k)\right).
\end{aligned}
\tag{49}
$$

$\boldsymbol{d}^{k+1}$ update: Since the problem is l_1 minimization problem, shrinkage algorithm is applicable as

$$
\boldsymbol{d}^{k+1}=\mathrm{Shrink}\left(\boldsymbol{D}\boldsymbol{u}^{k+1}+\boldsymbol{b}^k,\frac{1}{\lambda}\right). \tag{50}
$$

The algorithm of the Split Bregman Iteration is given in Alg. 8.

Algorithm 8 Tight frame non-blind deconvolution by Split Bregman Iteration

Task: Recover the blur-free image $\boldsymbol{u}$.
Input: Observed blurred image $\boldsymbol{f}$.
  Initialization: Set $\boldsymbol{d}_x^0=\mathbf{0}$, $\boldsymbol{b}_x^0=\mathbf{0}$, $\boldsymbol{d}_y^0=\mathbf{0}$, and $\boldsymbol{b}_y^0=\mathbf{0}$.
  while until convergence **do**
    $\boldsymbol{u}^{k+1}=\mathcal{F}^{-1}\left(\boldsymbol{W}\boldsymbol{F}^k\right)$
    $\boldsymbol{d}^{k+1}=\mathrm{Shrink}\left(\boldsymbol{D}\boldsymbol{u}^{k+1}+\boldsymbol{b}^k,\frac{1}{\lambda}\right)$
    $\boldsymbol{b}^{k+1}=\boldsymbol{b}^k+(\boldsymbol{D}\boldsymbol{u}^{k+1}-\boldsymbol{d}^{k+1})$
  end while
Output: Deblurred image $\boldsymbol{u}=\boldsymbol{f}$.

Bibliography

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967. 2

Jérôme Gilles. The bregman cookbook. `http://www.math.ucla.edu/~jegilles/BregmanCookbook.html`, 2011. 1

Tom Goldstein and Stanley Osher. The split Bregman method for l_1-regularized problems. *SIAM Journal on*

Imaging Sciences, 2:323–343, 2009. ISSN 1936-4954. doi: 10.1137/080725891. URL `http://dl.acm.org/citation.cfm?id=1658384.1658386`. 5