Image Inpainting with Total Variation Using Split Bregman Techniques

Zhenya Liu

May 23, 2023

1 Introduction

The Image inpainting problem refers to the task of restoring a missing part from an image decomposed into "known" and "unknown" region.

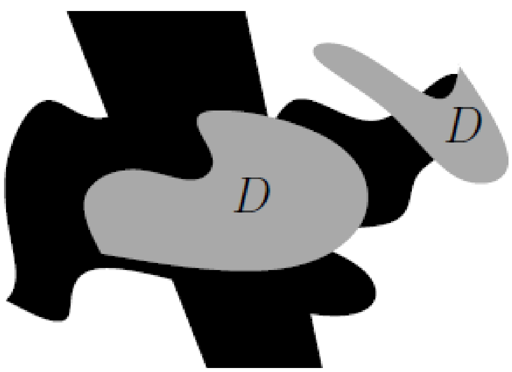


Figure 1: (followed from Papafitsoros et al.(2013), page 113)[6]. This figure is one illustration of image inpainting problems, where we consider the whole image as Ω, and missing parts as D.

In the general notation, we consider the whole image space as Ω, which is usually one rectangle, and model the image function as $u : \Omega \to \mathbb{R}$. In the grayscale image case, $u(x)$ is the intensity of the grey level at the point $x \in \Omega$. The missing part of the image is denoted by $D \subseteq \Omega$.

One of the traditional methods to deal with this problem is via total variation techniques, which attempt to find the expected image solution by minimizing a functional which includes the total variation of the weak derivative of u. Numerically, one popular method to solve this problem is to use Split Bregman algorithm which converts the original optimization problem into easily solved subproblems. In this set of notes, we will go through precise formulation of this algorithm, and show its application to solve the image inpainting problem.

2 Total Variation Method

The total variation method applied in image inpainting is very similar to the case for image denoizing problem. It is intuitive to observe that image denoizing can be treated as one special case of image inpainting by considering noises as small missing area for the whole image. We aim to fill the missing part such that the new image looks smooth generally. More precisely, we want "piecewisely smoothness" because we want to keep edges in the missing area. Thus, we follow the same idea with image denoizing and naturally propose the total variation.

Follow the definition of total vairation introduced by Getreuer, page 149[4], consider the Bounded Variation Space $BV(\Omega)$, the TV seminorm. Function $u \in BV(\Omega)$ if there exists weak derivative Du such that

$$\int_\Omega u(x)\,\mathrm{div}\,\vec{g}(x)dx = -\int_\Omega \langle \vec{g}, Du(x)\rangle \quad \text{for all } \vec{g} \in C_c^1\left(\Omega, \mathbb{R}^2\right)^2 \tag{1}$$

When u is smooth, the total variation seminorm of u is

$$\|u\|_{\mathrm{TV}(\Omega)} = \int_\Omega |Du| = \int_\Omega |\nabla u| \tag{2}$$

The TV method is to find the $u \in BV(\Omega)$ such that

$$u = \underset{u \in BV(\Omega)}{\arg\min} \|u\|_{\mathrm{TV}(\Omega)} + \frac{\lambda}{2}\int_{\Omega\setminus D}(f(x)-u(x))^2 dx, \tag{3}$$

Here f represents the original image, and λ is the positive parameter which controls the effect of denoizing. Notice that here we consider both denoizing and inpainting of original figure. The first term of (3) is called the *regularising term*, which asks for smoothness of solution u, and the second term is called *fidelity term*, which asks u to keep the same value with f outside from impainting domain.

The minimization problem shown in (3) is nearly same as the Rudin, Osher, and Fatemi (ROF) denoising problem except that in the second term, ROF problem takes the integral over whole Ω. Under certain mild condition, the existence of minimizer in $BV(\Omega)$ is robustly shown in the chapter (2.1),(2.2),(3.2) of [1]. We can regard (3) as denoizing problem by assigning a spatially-varying regularization term $\lambda(x)$,

$$u = \underset{u \in BV(\Omega)}{\arg\min} \|u\|_{\mathrm{TV}(\Omega)} + \frac{1}{2}\int_{\Omega}\lambda(x)(f(x)-u(x))^2 dx, \tag{4}$$

here $\lambda(x) = 0$ when $x \in D$, and $\lambda(x) > 0$ when outside of D.

3 Bregman Iteration

To solve the minimization problem like (3), note that the objection function includes a convex but not differentiable term, we will therefore find it useful to apply general convex optimization techniques. For instance, Bregman Iteration is one useful technique for solving convex problem like:

$$\arg\min_u J(u) \quad \text{s.t } H(u) = 0 \tag{5}$$

where J, H are convex functions in one Hilbert Space $\mathcal{H}$ with $\min H(u) = 0$.

We also make use of the *subgradient* of a convex function J at v, defined via $\partial J(v) = \{p : J(u) \geq J(v) + \langle p, u - v \rangle, \forall u\}$. When J is differentiable at v, then the single element of $\partial J(v)$ is $p = \nabla J(v)$.

The specific technique we will consider is an iteration known as the Bregman Iteration, based on the following notion of Bregman distance (this discussion is followed from Getreuer[3]).

Definition 1. *[3] Define the Bregman distance of J between u and v be*

$$D_J^p(u, v) = J(u) - J(v) - \langle p, u - v \rangle, p \in \partial J(v) \tag{6}$$

Geometrically, $D_J^p(u, v)$ is the difference between $J(u)$ and the tangent plane $J(v) + \langle p, u - v \rangle$.

Note that when u is the minimizer of J, then $D_J^p(u, v) \geq 0$ and $D_J^p(u, v) \geq D_J^p(w, v)$ for all w on the line segment $[u, v]$. Now we define *Bregman Iteration* to solve (5).

Definition 2. *[3] Given $\lambda > 0$, the Bregman Iteration for minimization problem* (5) *is*

$$u_{k+1} = \arg\min_u D_J^{p_k}(u, u_k) + \lambda H(u) \ , \ p_k \in \partial J(u_k) \tag{7}$$

or equivalent, expand $D_J^{p_k}(u, u_k)$ and drop constant terms,

$$u_{k+1} = \arg\min_u J(u) - \langle p_k, u - u_k \rangle + \lambda H(u) \ , \ p_k \in \partial J(u_k) \tag{8}$$

Proposition 3.1. *[3]The Bregman distance has following properties:*

1. $D_J^p(v, v) = 0$

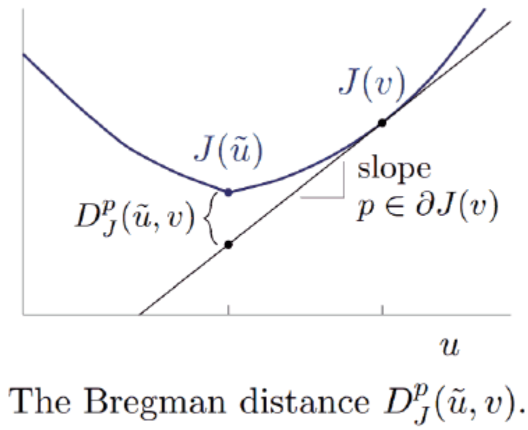


Figure 2: This figure copies from Getreuer[3]. It illustrates the Bregman distance in 1-D geometrically.

2. $D_J^p(u,v) \geq 0$

3. $D_J^p(u,v) + D_J^{\tilde{p}}(v,\tilde{v}) - D_J^{\tilde{p}}(u,\tilde{v}) = \langle p - \tilde{p}, v - u\rangle$

The proof is trivial and directly follows the definition of subgradient and convexity of J.

Proposition 3.2. *[3] The Bregman Iteration (5) decreases $H(u_k)$ monotonically with*

$$H\left(u_{k+1}\right) \leq H\left(u_{k+1}\right) + \frac{1}{\lambda} D_J^{p_k}\left(u_{k+1}, u_k\right) \leq H\left(u_k\right) \tag{9}$$

Proof. Since u_{k+1} minimizes $D_J^{p_k}(u, u_k) + \lambda H(u)$,

$$\begin{aligned} \lambda H\left(u_{k+1}\right) &\leq D_J^{p_k}\left(u_{k+1}, u_k\right) + \lambda H\left(u_{k+1}\right) \\ &\leq D_J^{p_k}\left(u_k, u_k\right) + \lambda H\left(u_k\right) = \lambda H\left(u_k\right) \end{aligned}$$

□

This proposition shows that the general Bregman Iteration can generate u_k such that $H(u_k)$ is monotonically decreasing.

Proposition 3.3. *[3] Suppose H is differentiable, then $p_k - \lambda \nabla H(u_{k+1}) \in \partial J(u_{k+1})$, and the Iteration (7) or (8) has the special case by choosing p_{k+1} such that*

$$\begin{cases} u_{k+1} = \arg\min_u D_J^{p_k}\left(u, u_k\right) + \lambda H(u) \\ p_{k+1} = p_k - \lambda \nabla H\left(u_{k+1}\right). \end{cases} \tag{10}$$

Moreover, let $u^\star$ be the minimizer of H, the iteration (10) implies that

$$H(u_k) \le H(u^\star) + \frac{D_J^{p_0}(u^\star, u_0)}{\lambda k} \tag{11}$$

Proof. Suppose f is convex and smooth and g is convex, and if x is a minimum point of $x \mapsto f(x) + g(x)$, then $0 \in f'(x) + \partial g(x)$. Then for the iteration (8)

$$u_{k+1} = \arg\min_u \; J(u) - \langle p_k, u - u_k \rangle + \lambda H(u)$$

Here J is convex, and $(-\langle p_k, u - u_k\rangle + \lambda H(u))$ is convex and differentiable, since u_{k+1} is the minimizer,

$$\begin{aligned} 0 \in \partial J(u_{k+1}) - p_k + \lambda \nabla H(u_{k+1}) \\ p_k - \lambda \nabla H\left(u_{k+1}\right) \in \partial J\left(u_{k+1}\right) \end{aligned}$$

So we show that (10) is one special case of Bregman Iteration. From Proposition 3.1, the iteration (10) also satisfies

$$\begin{aligned} D_J^{p_k}\left(u^\star, u_k\right) - D_J^{p_{k-1}}\left(u^\star, u_{k-1}\right) &\le D_J^{p_k}\left(u^\star, u_k\right) + D_J^{p_{k-1}}\left(u_k, u_{k-1}\right) - D_J^{p_{k-1}}\left(u^\star, u_{k-1}\right) \\ &= \langle p_k - p_{k-1}, u_k - u^\star \rangle \\ &= \langle \lambda \nabla H\left(u_k\right), u^\star - u_k \rangle \\ &\le \lambda\left(H\left(u^\star\right) - H\left(u_k\right)\right) \end{aligned}$$

The last inequality is achieved by the convexity of H. Now summing k from $k = 0$ to $k = K$, and apply Proposition 3.2, we get

$$\begin{aligned} \sum_{k=1}^{K} D_J^{p_k}\left(u^\star, u_k\right) - D_J^{p_{k-1}}\left(u^\star, u_{k-1}\right) + \lambda\left(H\left(u_k\right) - H\left(u^\star\right)\right) &\le 0 \\ D_J^{p_K}\left(u^\star, u_K\right) - D_J^{p_0}\left(u^\star, u_0\right) + \lambda K\left(H\left(u_K\right) - H\left(u^\star\right)\right) &\le 0 \\ - D_J^{p_0}\left(u^\star, u_0\right) + \lambda K\left(H\left(u_K\right) - H\left(u^\star\right)\right) &\le 0 \\ H(u_k) \le H(u^\star) + \frac{D_J^{p_0}(u^\star, u_0)}{\lambda k} \end{aligned}$$

□

Proposition 3.3 shows that with additional differentiability of H, the Bregman Iteration can generate u_k such that $H(u_k) \to H(u^\star)$. For image inpainting problem, we always use $H(u) = \frac{1}{2}\|Au - f\|_2^2$, and the minimization problem actually becomes

$$\arg\min_u \; J(u) \quad \text{s.t } \frac{1}{2}\|Au - f\|_2^2 = 0 \Leftrightarrow Au = f \tag{12}$$

The Bregman Iteration of (12) is

$$\begin{cases} u_{k+1} = \arg\min_u D_J^{p_k}\left(u, u_k\right) + \frac{\lambda}{2}\|Au - f\|_2^2 \\ p_{k+1} = p_k - \lambda A^T(Au_{k+1} - f). \end{cases} \tag{13}$$

Theorem 3.1. *[3] If A is a linear operator, Iteration (13) given $p_0 = 0$ can be reformulated as a nice form:*

$$\begin{cases} u_{k+1} = \arg\min_u J(u) + \frac{\lambda}{2}\|Au - f_k\|_2^2 \\ f_{k+1} = f_k + (f - Au_{k+1}), \quad f_0 = f \end{cases} \tag{14}$$

Proof. On the $K-th$ iteration

$$\begin{aligned}
\lambda H(u) - \langle p_K, u - u_k\rangle &= \lambda H(u) - \left\langle p_0 - \lambda \sum_{k=1}^{K} \nabla_u H(u_k), u - u_k \right\rangle \\
&= \frac{\lambda}{2}\|Au\|_2^2 + \frac{\lambda}{2}\|f\|_2^2 - \lambda\langle f, Au\rangle + \lambda \left\langle \sum_{k=1}^{K} A^*(Au_k - f), u - u_k \right\rangle \\
&= \frac{\lambda}{2}\|Au\|_2^2 + \frac{\lambda}{2}\|f\|_2^2 - \lambda \left\langle f + \sum_{k=1}^{K}(f - Au_k), Au \right\rangle + \lambda \left\langle \sum_{k=1}^{K} A^*(Au_k - f), -u_k \right\rangle \\
&= \frac{\lambda}{2}\|Au\|_2^2 + \frac{\lambda}{2}\|f\|_2^2 - \lambda\langle f_K, Au\rangle + \lambda \left\langle \sum_{k=1}^{K} A^*(Au_k - f), -u_k \right\rangle \\
&= \frac{\lambda}{2}\|Au\|_2^2 + \frac{\lambda}{2}\|f_K\|_2^2 - \lambda\langle f_K, Au\rangle + \frac{\lambda}{2}\|f\|_2^2 - \frac{\lambda}{2}\|f_K\|_2^2 + \lambda \left\langle \sum_{k=1}^{K} A^*(Au_k - f), -u_k \right\rangle \\
&= \frac{\lambda}{2}\|Au - f_K\|_2^2 + C
\end{aligned}$$

Thus, for each iteration of u_{k+1} in (13),

$$\begin{aligned}
u_{k+1} &= \arg\min_u D_J^{p_k}(u, u_k) + \frac{\lambda}{2}\|Au - f\|_2^2 \\
&= \arg\min_u J(u) - \langle p_k, u - u_k\rangle + \frac{\lambda}{2}\|Au - f_k\|_2^2 \\
&= \arg\min_u J(u) + \frac{\lambda}{2}\|Au - f_K\|_2^2 + C \\
&= \arg\min_u J(u) + \frac{\lambda}{2}\|Au - f_K\|_2^2
\end{aligned}$$

Here we show that for each iteration, (14) generates u_{k+1} which is one valid solution for (13). So the iteration (13) and (14) are equivalent in the perspective that for each iteration they can generate u_{k+1} from the same objective function. □

Theorem 3.1 shows that the Bregman Iteration for (12) can be considered as (14). Suppose $u^\star$ is the minimizer of $H(u) = \frac{1}{2}\|Au - f\|_2^2$, and $H(u^\star) = 0$, since H is convex and differentiable, from previous propositions $H(u_k) \to H(u^\star)$. Furthermore, we can conclude that $J(u_k) \to J(\tilde{u})$, where $\tilde{u}$ is the solution of (12).

Goldstein[5] states the following result to show that we can get one valid solution based on iterations introduced by Theorem 3.1.

Proposition 3.4. *([5],page 6) From the iteration stated in* (14)*, suppose that at some iterate,*u_{k+1} *satisfies* $Au_{k+1} = f$*, the* u_{k+1} *is a valid solution of the original constrained problem* (12)*.*

Proof. At that iterate, $Au_{k+1} = f$, and

$$u_{k+1} = \arg\min_u J(u) + \frac{\lambda}{2} \|Au - f_k\|_2^2$$

Let $\tilde{u}$ be the true solution of problem (12), then $A\tilde{u} = Au_{k+1} = f$, which means

$$\|Au_{k+1} - f_k\|_2^2 = \|A\tilde{u} - f_k\|_2^2$$

Since u_{k+1} is the minimizer of this iterate,

$$J(u_{k+1}) + \frac{\lambda}{2} \|Au_{k+1} - f_k\|_2^2 \leq J(\tilde{u}) + \frac{\lambda}{2} \|A\tilde{u} - f_k\|_2^2$$

We get $J(u_{k+1}) \leq J(\tilde{u})$, and so $J(u_{k+1}) = J(\tilde{u})$. u_{k+1} is the solution of constrained optimization problem (12). □

Note that Proposition 3.4 argues in the case that $H(u_k) = 0$ in finite iterations, and we can use the similar proof to show that in general case $J(u_k) \to J(\tilde{u})$ given that $H(u_k) \to H(\tilde{u})$.

Now we conclude the Bregman Iteration algorithm:

Algorithm 1 Bregman Iteration

Input: J, H are convex, $H(u) = Au - f$, A is linear, $\lambda > 0$.
Output: $\tilde{u} = \arg\min J(u)$ s.t $H(u) = 0$

```
  f_0 ← f
  u_0 ← arg min J(u) + (λ/2)||Au − f_0||_2^2
  k ← 0
  while J(u_k) not converge do
      u_{k+1} ← arg min J(u) + (λ/2)||Au − f_k||_2^2
      f_{k+1} ← f_k + (f_0 − Au_{k+1})
      k ← k + 1
  end while
  return u_k
```

By [5], this algorithm has several advantages over traditional penalty methods worked with constrained optimization problem. The first is that if J contains $L1$ regularization term (widely used in image denoise and image inpaint), the algorithm converges very quickly. Second, notice that λ is a constant in whole iterations compared to increasing sequences of λ used in penalty methods, we can choose λ to minimize the conditional number of A, so that the subproblem in each iteration can converge efficiently by applying classical iterative methods for unconstrained optimization problem.

4 Split Bregman

Now consider the $L1$ regularized problem which is closer to problem (4):

$$\min_u |\Phi(u)| + E(u) \tag{15}$$

where $|\cdot|$ denotes $L1$ norm, both $|\Phi(u)|$ and $E(u)$ are convex functions, and Φ is linear. To apply Bregman Iteration, we convert (15) to one equivalent constrained optimization problem:

$$\arg\min_{u,d} |d| + E(u) \quad \text{s.t } d = \Phi(u) \tag{16}$$

Define $J(u,d) = |d| + E(u)$, and $H(u,d) = \frac{1}{2}\|d - \Phi(u)\|_2^2$.

Proposition 4.1. *([5],page 7) When J, H are convex, and H is differentiable for u, d, by applying the Bregman Iteration* (13)*, we can get*

$$\begin{aligned}
\left(u^{k+1}, d^{k+1}\right) &= \arg\min_{u,d} D_J^{\mathbf{p}}\left((u,d),(u^k,d^k)\right) + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2 \\
&= \arg\min_{u,d} J(u,d) - \left\langle p_u^k, u - u^k \right\rangle - \left\langle p_d^k, d - d^k \right\rangle + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2 \\
p_u^{k+1} &= p_u^k - \lambda(\nabla\Phi)^T\left(\Phi(u^{k+1}) - d^{k+1}\right) \\
p_d^{k+1} &= p_d^k - \lambda\left(d^{k+1} - \Phi(u^{k+1})\right)
\end{aligned}$$

Proof. Denote $u \in \mathcal{H}_1, d \in \mathcal{H}_2$, then $J(u,d)$ and $H(u,d)$ are functionals defined from the Cartesian Hilbert Space $\mathcal{H}_1 \times \mathcal{H}_2$ to $\mathbb{R}$, where $\mathcal{H}_1 \times \mathcal{H}_2$ equips with inner product $\langle (u_1,d_1),(u_2,d_2)\rangle_{\mathcal{H}_1\times\mathcal{H}_2} = \langle u_1,u_2\rangle_{\mathcal{H}_1} + \langle d_1,d_2\rangle_{\mathcal{H}_2}$. The subgradient of $J(u,d)$ can be written as $\mathbf{p} = (p_u, p_d) \in \partial J$. Write $F(d) = |d|$, since $J(u,d) = E(u) + F(d)$, which can be divided into two single variable (separable), it is not hard to find that $p_u \in \partial E(u)$ and $p_d \in \partial F(d)$. Then the Bregman Iteration (13) applied on the question (16) is like

$$\begin{cases}
(u^{k+1}, d^{k+1}) = \arg\min\limits_{(u,d)} D_J^{\mathbf{p}}\left((u,d),(u^k,d^k)\right) + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2 \\
(p_u^{k+1}, p_d^{k+1}) = (p_u^k, p_d^k) - \lambda\nabla H(u^{k+1}, d^{k+1}).
\end{cases} \tag{17}$$

The Bregman Distance in this space becomes

$$\begin{aligned}
D_J^{\mathbf{p}}\left((u,d),(u^k,d^k)\right) &= J(u,d) - J(u^k,d^k) - \left\langle (p_u^k, p_d^k), (u - u^k, d - d^k) \right\rangle \\
&= J(u,d) - J(u^k,d^k) - \left\langle p_u^k, u - u^k \right\rangle - \left\langle p_d^k, d - d^k \right\rangle
\end{aligned}$$

Also,

$$\nabla H(u^{k+1}, d^{k+1}) = \left((\nabla\Phi)^T(\Phi(u^{k+1}) - d^{k+1}), d^{k+1} - \Phi(u^{k+1})\right)$$

Substitute these two results into (17), we get the expect iteration algorithm. $\square$

By Theorem 3.1, iterations in Proposition 4.1 can be written in elegant form:

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

To implement this algorithm, we follow "split" procedure for the problem

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

We can split $L1$ and $L2$ components from the objection function, and iteratively minimize u and d separately. This method refers to "Split Bregman". Specifically, (18) splits into 2 subproblems which can be efficient solved ([5],page 7):

$$u^{k+1} = \arg\min_{u} E(u) + \frac{\lambda}{2}\left\|d^k - \Phi(u) - b^k\right\|_2^2 \tag{19}$$

$$d^{k+1} = \arg\min_{d} |d| + \frac{\lambda}{2}\left\|d - \Phi\left(u^{k+1}\right) - b^k\right\|_2^2 \tag{20}$$

To solve u in (19), assume that E is differentiable, then the subproblem can be solved by any calculus-based unconstrained optimization methods such as *Gradient Descent* and *Newton's method*. If E is in particular form such that we can find the linear closed form of the minimizer, then it is efficient to apply *Gauss-Seidel* to iteratively solve the linear system.

To solve d in (20), to minimize the $L1$ proximal operator, we have *shrinkage* operator to minimize objection function with $L1$ term.d can be explicitly calculated by ([5],page 8)

$$d_j^{k+1} = shrink\left(\Phi(u)_j + b_j^k, 1/\lambda\right) \tag{21}$$

where $shrink : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$,

$$shrink(x, \gamma) = \frac{x}{|x|} * \max(|x| - \gamma, 0) \tag{22}$$

Here give the general Split Bregman algorithm (see next page):

Algorithm 2 Split Bregman for (18)

$u^0, d^0, b^0 \leftarrow 0$
$k \leftarrow 0$
while u^k, d^k not converge **do**
  $u_{k+1} \leftarrow \arg\min_u E(u) + \frac{\lambda}{2}\left\|d^k - \Phi(u) - b^k\right\|_2^2$
  $d^{k+1} = \min_d |d| + \frac{\lambda}{2}\left\|d - \Phi\left(u^{k+1}\right) - b^k\right\|_2^2$
  $b^{k+1} \leftarrow b^k + \left(\Phi\left(u^{k+1}\right) - d^{k+1}\right)$
  $k \leftarrow k+1$
end while
return u_k

In practice, this algorithm runs efficiently because it depends on efficiencies of two subproblems. Even if subproblem u is solved inexactly, the algorithm can still converge. This algorithm is also robust to numerical imprecision. First, every fix point $(u^\star, d^\star)$ is the minimizer of the original constrained problem (16) by Proposition 3.4. Moreover, given fix points $u^\star$ and $b^\star$, then d becomes $d^\star$ by $b^* = b^* + \Phi u^* - d^*$, which implies $d^* = \Phi u^*$. Apply Proposition 3.4 again, we show that $(u^\star, d^\star)$ is a solution to (16).

v

5 Split Bregman v.s. ADMM Algorithm

Another popular strategy to solve constrained conven optimization problems is Alternating Direction Method of Multipliers(ADMM) algorithm.
Follow Boyed[2]'s description of ADMM, when solving the general constrained optimization problem like

$$\min_{x,z} f(x) + g(z) \text{ subject to } Ax + Bz = c$$

We define augmented Lagrangian, for a parameter $\lambda > 0$,

$$L_\lambda(x, z, b) = f(x) + g(z) + b^T(Ax + Bz - c) + \frac{\lambda}{2}\|Ax + Bz - c\|_2^2$$

for $k = 1, 2, 3, \ldots$

$$\begin{aligned}
x^{(k)} &= \underset{x}{\operatorname{argmin}}\ L_\lambda\left(x, z^{(k-1)}, b^{(k-1)}\right) \\
z^{(k)} &= \underset{z}{\operatorname{argmin}}\ L_\lambda\left(x^{(k)}, z, b^{(k-1)}\right) \\
b^{(k)} &= b^{(k-1)} + \lambda\left(Ax^{(k)} + Bz^{(k)} - c\right)
\end{aligned}$$

where f, g are convex.

Reconsider (16) which we solved by Split Bregman algorithm,

$$\min_{u,d} |d| + E(u) \;\text{ s.t } d = \Phi(u)$$

Apply ADMM algorithm, we first define the augmented Lagrangian

$$L_\lambda(u,d,b) = E(u) + |d| + b^T(d - \Phi(u)) + \frac{\lambda}{2}\|d - \Phi(u)\|_2^2$$

Then the iteration becomes

$$\begin{cases} u^{k+1} = \underset{u}{\operatorname{argmin}}\; E(u) + |d^k| + (b^k)^T(d^k - \Phi(u)) + \frac{\lambda}{2}\|d^k - \Phi(u)\|_2^2 \\ d^{k+1} = \underset{d}{\operatorname{argmin}}\; E(u^{k+1}) + |d| + (b^k)^T(d - \Phi(u^{k+1})) + \frac{\lambda}{2}\|d - \Phi(u^{k+1})\|_2^2 \\ b^{k+1} = b^k + \lambda\left(d^{k+1} - \Phi(u^{k+1})\right) \end{cases}$$

Compare it with Split Bregman of problem (16)

$$\begin{cases} u^{k+1} = \arg\min_u E(u) + \frac{\lambda}{2}\left\|d^k - \Phi(u) - b^k\right\|_2^2 \\ d^{k+1} = \arg\min_d |d| + \frac{\lambda}{2}\left\|d - \Phi\left(u^{k+1}\right) - b^k\right\|_2^2 \\ b^{k+1} = b^k + \left(\Phi\left(u^{k+1}\right) - d^{k+1}\right) \end{cases}$$

Easy to check that these two algorithms are actually equivalent: for u and d subproblems, expanding $L2$ entries and drop all irrelevant terms, then we can gain exactly same objection functions. For b update, note that in both algorithms, $\Phi(u^k) \to d^k$, which guarantees that u and d converge to a valid solution of (16).

These two methods are similar in the sense that both solve a constrained convex optimization problems by iterative unconstrained problems which can be divided into simpler subproblems. The difference shows on the implemented principle: the Split Bregman algorithm is based on Bregman distance, but ADMM algorithm essentially solves the dual problem of one augmented Lagrangian version of the original problem.

6 Numerical Application I

Switch back to the Image Inpaint problem, where we use TV method to solve

$$u = \arg\min_{u \in BV(\Omega)} \|u\|_{\mathrm{TV}(\Omega)} + \frac{1}{2}\int_\Omega \lambda(x)(f(x) - u(x))^2 dx, \tag{23}$$

Follow the method designed by Gretreuer[4] (page $149-151$), we will discretize this problem and numerically solve it with Split Bregman algorithm. Consider Ω as $N\times N$ image, $i.e$ $\mathbb{R}^{N\times N}$. Define the discrete derivative of u as ∇u, and $\nabla u_{ij}=(D_xu(i,j),D_yu(i,j))$ with forward difference operator

$$D_xu(i,j)=\begin{cases}u(i,j+1)-u(i,j) & \text{if } 1\le i\le N, 1\le j<N\\ u(i,1)-u(i,j) & \text{if } 1\le i\le N, j=N\end{cases}$$
$$D_yu(i,j)=\begin{cases}u(i+1,j)-u(i,j) & \text{if } 1\le i<N, 1\le j\le N\\ u(1,j)-u(i,j) & \text{if } i=N, 1\le j\le N\end{cases}$$

And define discrete divergence through $\operatorname{div}:=-\nabla^*=-D_x^*-D_y^*$ and discrete Laplacian by $\Delta:=\operatorname{div}\nabla$:

$$\begin{aligned}\operatorname{div}v(i,j)&=v^x_{i,j}-v^x_{i-1,j}+v^y_{i,j}-v^y_{i,j-1}\\ \Delta u(i,j)&=-4u_{i,j}+u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\end{aligned}$$

TV is approximated by summing over all $|\nabla u_{ij}|$ over pixels in Ω,

$$\|u\|_{\mathrm{TV}(\Omega)}\approx\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|\nabla u_{i,j}|$$

After discretion, the problem (23) becomes

$$\begin{aligned}&\arg\min_{d,u}\sum_{i,j}|d_{i,j}|+\frac{1}{2}\sum_{i,j}\lambda_{i,j}\left(f_{i,j}-u_{i,j}\right)^2\\ &\text{s.t } d=\nabla u\end{aligned}\tag{24}$$

Now apply Bregman Iteration stated in (18), we need to solve

$$\arg\min_{d,u}\sum_{i,j}|d_{i,j}|+\frac{1}{2}\sum_{i,j}\lambda_{i,j}\left(f_{i,j}-u_{i,j}\right)^2+\frac{\gamma}{2}\sum_{i,j}|d_{i,j}-\nabla u_{i,j}-b_{i,j}|^2\tag{25}$$

here we neglect the superscript k to represent the iterated step, and b is updated in the way $b^{k+1}=b^k+\nabla u^{k+1}-d^{k+1}$.
Use Split Bregman, (25) is divided into **subproblem** u and **subproblem** d:

Subproblem u
With d fixed, the u subproblem is

$$\arg\min_{u}\ \frac{1}{2}\sum_{i,j}\lambda_{i,j}\left(f_{i,j}-u_{i,j}\right)^2+\frac{\gamma}{2}\sum_{i,j}|d_{i,j}-\nabla u_{i,j}-b_{i,j}|^2$$

u can be solved by a closed form

$$\frac{1}{\gamma}\lambda u - \Delta u = \frac{1}{\gamma}\lambda f - \operatorname{div}(d - b) \tag{26}$$

Given our discretized definition of Δ and div, (26) is actually one linear system which has variables of all u_{ij}.

Subproblem d

With u fixed, the d subproblem is

$$\arg\min_{d} \sum_{i,j} |d_{i,j}| + \frac{\gamma}{2} \sum_{i,j} |d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2 \tag{27}$$

Recall *shrink* operator in (21) and (22), (27) is solved as

$$d_{i,j} = \frac{\nabla u_{i,j} + b_{i,j}}{|\nabla u_{i,j} + b_{i,j}|} \max\{|\nabla u_{i,j} + b_{i,j}| - 1/\gamma, 0\} \text{ for all } i, j \tag{28}$$

Then we present the algorithm which solves (24).

Algorithm 3 Split Bregman for (24)

$u, d, b \leftarrow 0$
$k \leftarrow 0$
while $\|u_{new} - u_{old}\|_2 > Tol$ **do**
  $u \leftarrow$ *solve* (26)
  $d \leftarrow$ *solve* (28)
  update b
  $k \leftarrow k + 1$
end while
return u

In practice, γ can be selected as any positive constant, and it can be specifically chosen to keep a good convergence rate for both subproblems during each iteration.

Example

We list some examples stated in [4] to show the effect of algorithm 3, given that $\gamma = 5, Tol = \|f\|_2/10^5$.

Text Removal

TV inpainting always has excellent performance in text removal (see Figure 3). For the text removal problem, the inpainting domain D is the area taken by "text". Since this domain is locally "thin", using TV method, it is filled with smooth colors such that people will not find much irregularity from perceptive cognition.

Figure 3: This figure copies from Getreuer(2012), page 152. Text Removal Example I ($\lambda = 10^4$)

Figure 4 shows one important feature of TV method that minimizing TV seminorm allows piecewise smoothness, which means it can reconstruct edges in the original image.

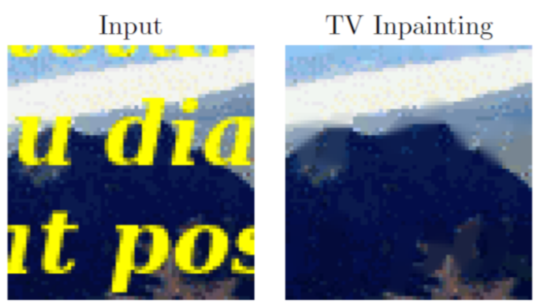


Figure 4: This figure copies from Getreuer(2012), page 152. Details of Figure 3. We can observe that TV method reconstructs the edge.

Generally, besides text removal, TV inpainting can always have great results when dealing with images which have "thin" and "small" impainting domains, such as wires, scratches, and noises.

Object Removal

TV inpainting also applies on removing objects (see Figure 5). With prior analysis, the result is good when the inpaint domain D is thin. However, when D is larger and thicker, the result will perform poorly in general.

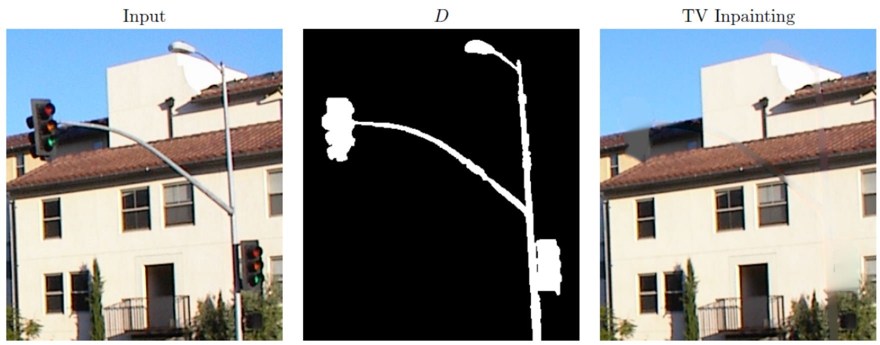


Figure 5: This figure copies from Getreuer(2012), page 155[4]. The figure shows the performance of TV method when removing a traffic light. We can observe it looks good in thin area, but poor in thick and large area.

We can observe that this "poor" portion is caused by the "piecewise constant" color by TV inpainting. When the missing part is large and thick, people will find much irregularity when it is filled with constant colors. To relieve this problem, we need to allow smoother propagation of color on the image, like the effect of "blurring". Thus, we introduce another TV inpainting method which combines both first and second order total variation.

7 Numerical Application II: Second Order Total Variation Method

Compared with TV method (3)

$$u = \underset{u \in BV(\Omega)}{\arg\min} \|u\|_{\mathrm{TV}(\Omega)} + \frac{\lambda}{2} \int_{\Omega \setminus D} (f(x) - u(x))^2 dx,$$

Papafitsoros et al.[6] proposes a new method by adding one regularising term $\mathrm{TV}^2(u) = \mathrm{TV}(\nabla u)$, which is the total variation of the gradient. We call this TV^2 method, and the minimization problem is

$$u_r = \underset{u \in \mathrm{BH}(\Omega)}{\arg\min} \int_{\Omega \setminus D} (u - u_0)^2 \, dx + \alpha \mathrm{TV}(u) + \beta \mathrm{TV}^2(u) \tag{29}$$

where α, β are positive weighted parameters which control inpainting effects. Analogous to $BV(\Omega)$, the appropriate Banach space for minimizing (29) is the space of bounded Hessian function $BH(\Omega)$. For twice differentiable u, $\mathrm{TV}^2(u)$ is the $L1$ norm of the Hessian, *i.e* $\mathrm{TV}^2(u) = \int_\Omega |\nabla^2 u|$. The motivation of using this method is that the combination of TV and TV^2 can fill the inpainting domain with blurred effect. Furthermore, it has potential to connect for large gaps where traditional TV method can not achieve. We will show the numerical implementation of (29) ([6], page 116-118).

Initially, we discretize D_x, D_y, D_{xx}, D_{yy}, and D_{xy} using forward difference and backward difference, where results from backward are used to calculate divergence and solution of subproblems.
Consider $\Omega = \mathbb{R}^{n\times m}$. For discrete $\nabla u = (D_x u, D_y u)$ using forward difference,

$$D_x u(i,j) = \begin{cases} u(i,j+1) - u(i,j) & \text{if } 1 \le i \le n, 1 \le j < m \\ u(i,1) - u(i,j) & \text{if } 1 \le i \le n, j = m \end{cases}$$

$$D_y u(i,j) = \begin{cases} u(i+1,j) - u(i,j) & \text{if } 1 \le i < n, 1 \le j \le m \\ u(1,j) - u(i,j) & \text{if } i = n, 1 \le j \le m \end{cases}$$

The partial derivative using back difference denotes as

$$\overleftarrow{D_x} u(i,j) = \begin{cases} u(i,j) - u(i,m) & \text{if } 1 \le i \le n, j = 1 \\ u(i,j) - u(i,j-1) & \text{if } 1 \le i \le n, 1 < j \le m \end{cases}$$

$$\overleftarrow{D_y} u(i,j) = \begin{cases} u(i,j) - u(n,j) & \text{if } i = 1, 1 \le j \le m \\ u(i,j) - u(i-1,j) & \text{if } 1 < i \le n, 1 \le j \le m \end{cases}$$

Given $p = (p_1, p_2) \in (\mathbb{R}^{n\times m})^2$

$$(\operatorname{div} p)(i,j) = \overleftarrow{D_x} p_1(i,j) + \overleftarrow{D_y} p_2(i,j)$$

Then define second partial derivative by compositions of first order operators:

$$D_{xx} u(i,j) = \begin{cases} u(i,m) - 2u(i,j) + u(i,j+1) & \text{if } 1 \le i \le n, j = 1, \\ u(i,j-1) - 2u(i,j) + u(i,j+1) & \text{if } 1 \le i \le n, 1 < j < m, \\ u(i,j-1) - 2u(i,j) + u(i,1) & \text{if } 1 \le i \le n, j = m, \end{cases}$$

$$D_{yy} u(i,j) = \begin{cases} u(n,j) - 2u(i,j) + u(i+1,j) & \text{if } i = 1, 1 \le j \le m, \\ u(i-1,j) - 2u(i,j) + u(i+1,j) & \text{if } 1 < i < n, 1 \le j \le m, \\ u(i-1,j) - 2u(i,j) + u(1,i) & \text{if } i = n, 1 \le j \le m, \end{cases}$$

$$D_{xy} u(i,j) = \begin{cases} u(i,j) - u(i+1,j) - u(i,j+1) + u(i+1,j+1) & \text{if } 1 \le i < n, 1 \le j < m, \\ u(i,j) - u(1,j) - u(i,j+1) + u(1,j+1) & \text{if } i = n, 1 \le j < m, \\ u(i,j) - u(i+1,j) - u(i,1) + u(i+1,1) & \text{if } 1 \le i < n, j = m, \\ u(i,j) - u(1,j) - u(i,1) + u(1,1) & \text{if } i = n, j = m. \end{cases}$$

Easy to check $D_{xy} = D_{yx}$. Similarly, define the second partial derivative with compositions of first order operators generated by backward difference, and conclude that

$$\overleftarrow{D_{xx}} = D_{xx}, \quad \overleftarrow{D_{yy}} = D_{yy}$$

$$\overleftarrow{D_{xy}}u(i,j) = \begin{cases} u(i,j) - u(i,m) - u(n,i) + u(n,m) & \text{if } i=1, j=1 \\ u(i,j) - u(i,j-1) - u(n,j) + u(n,j-1) & \text{if } i=1, 1<j\leq m \\ u(i,j) - u(i-1,j) - u(i,m) + u(i-1,m) & \text{if } 1<i\leq mj=1 \\ u(i,j) - u(i,j-1) - u(i-1,j) + u(i-1,j-1) & \text{if } 1<i\leq n, 1<j\leq m \end{cases}$$

Finally define the second order discrete divergence: for a $q = (q_{11}, q_{22}, q_{12}, q_{21}) \in (\mathbb{R}^{n\times m})^4$, define

$$\left(\operatorname{div}^2 q\right)(i,j) = \overleftarrow{D_{xx}}q_{11}(i,j) + \overleftarrow{D_{yy}}q_{22}(i,j) + \overleftarrow{D_{xy}}q_{12}(i,j) + \overleftarrow{D_{yx}}q_{21}(i,j)$$

After discretization, the opitimization problem (29) becomes

$$\begin{aligned} &\underset{u,\tilde{u},v,w}{\arg\min} \left\|\mathcal{X}_{\Omega\setminus D}\left(u-u_0\right)\right\|_2^2 + \alpha\|v\|_1 + \beta\|w\|_1 \\ &\qquad\qquad\qquad\qquad \text{s.t } u=\tilde{u}, v=\nabla\tilde{u}, w=\nabla^2\tilde{u} \end{aligned} \tag{30}$$

where $v = (v_x, v_y)$, $w = (w_{xx}, w_{yy}, w_{xy}, w_{yx})$. We introduce one auxiliary variable $\tilde{u}$ for allowing using FFT (Fast Fourier Transform) in a faster implementation.
Apply the Bregman Iteration to (30), we get

$$\begin{aligned} \left[u^{k+1}, \tilde{u}^{k+1}, v^{k+1}, w^{k+1}\right] =& \underset{u,\tilde{u},v,w}{\operatorname{argmin}} \left\|\mathcal{X}_{\Omega\setminus D}\left(u-u_0\right)\right\|_2^2 + \alpha\|v\|_1 + \beta\|w\|_1 \\ &+ \frac{\lambda_0}{2}\left\|b_0^k + \tilde{u} - u\right\|_2^2 + \frac{\lambda_1}{2}\left\|b_1^k + \nabla\tilde{u} - v\right\|_2^2 \\ &+ \frac{\lambda_2}{2}\left\|b_2^k + \nabla^2\tilde{u} - w\right\|_2^2 \\ b_0^{k+1} =& b_0^k + \tilde{u}^{k+1} - u^{k+1} \\ b_1^{k+1} =& b_1^k + \nabla\tilde{u}^{k+1} - v^{k+1} \\ b_2^{k+1} =& b_2^k + \nabla^2\tilde{u}^{k+1} - w^{k+1} \end{aligned} \tag{31}$$

with the initialization $u^1 = u_0, \tilde{u}^1 = u_0, v^1 = \nabla u_0, w^1 = \nabla^2 u_0, b_0^1 = \mathbf{1}, b_1^1 =$

$\mathbf{1}, b_2^1 = \mathbf{1}$. Then using Split Bregman, we convert (31) into 4 subproblems:

$$\begin{aligned}
\textbf{Subproblem 1}: \quad & u^{k+1} = \underset{u}{\operatorname{argmin}} \left\|\mathcal{X}_{\Omega\setminus D}\left(u-u_0\right)\right\|_2^2 + \frac{\lambda_0}{2}\left\|b_0^k + \tilde{u}^k - u\right\|_2^2 \\
\textbf{Subproblem 2}: \quad & \tilde{u}^{k+1} = \underset{\tilde{u}}{\operatorname{argmin}} \frac{\lambda_0}{2}\left\|b_0^k + \tilde{u} - u^{k+1}\right\|_2^2 + \frac{\lambda_1}{2}\left\|b_1^k + \nabla\tilde{u} - v^k\right\|_2^2 \\
& \qquad\qquad + \frac{\lambda_2}{2}\left\|b_2^k + \nabla^2\tilde{u} - w^k\right\|_2^2 \\
\textbf{Subproblem 3}: \quad & v^{k+1} = \underset{v}{\operatorname{argmin}} \alpha\|v\|_1 + \frac{\lambda_1}{2}\left\|b_1^k + \nabla\tilde{u}^{k+1} - v\right\|_2^2 \\
\textbf{Subproblem 4}: \quad & w^{k+1} = \underset{w}{\operatorname{argmin}} \beta\|w\|_1 + \frac{\lambda_2}{2}\left\|b_2^k + \nabla^2\tilde{u}^{k+1} - w\right\|_2^2,
\end{aligned}$$

with the update of b terms

$$\begin{aligned}
b_0^{k+1} &= b_0^k + \tilde{u}^{k+1} - u^{k+1} \\
b_1^{k+1} &= b_1^k + \nabla\tilde{u}^{k+1} - v^{k+1} \\
b_2^{k+1} &= b_2^k + \nabla^2\tilde{u}^{k+1} - w^{k+1}
\end{aligned} \tag{32}$$

We conclude above iterations as algorithm

Algorithm 4 Split Bregman for (30)

$u^1 \leftarrow u_0, \tilde{u}^1 \leftarrow u_0, v^1 \leftarrow \nabla u_0, w^1 \leftarrow \nabla^2 u_0$
$b_0^1 \leftarrow \mathbf{1}, b_1^1 \leftarrow \mathbf{1}, b_2^1 \leftarrow \mathbf{1}$
$k \leftarrow 1$
while $\|u^k - u^{k-1}\|_2 > Tol$ **do**
  $u^k \leftarrow$ *solve* **subproblem 1**
  $\tilde{u}^k \leftarrow$ *solve* **subproblem 2**
  $v^k \leftarrow$ *solve* **subproblem 3**
  $w^k \leftarrow$ *solve* **subproblem 4**
  update b_0^k, b_1^k, b_2^k
  $k \leftarrow k+1$
end while
return u^k

This algorithm will return u which is a valid solution of discretized optimization problem (29). Details of solutions for all subproblems are referred to [6], page 120-121.

Example

Pure TV v.s. Pure TV^2

When we set $\alpha = 0$, (29) becomes pure TV^2 inpainting, and when set $\beta = 0$, (29) becomes pure TV inpainting, so we first compare their performances for the same graph.

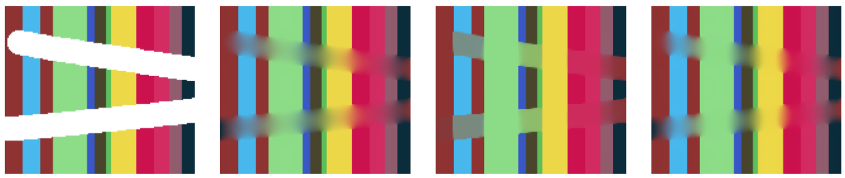
By Figure (6), we can observe in this text removal example, two methods look nearly same in global. However, by zooming out we can find that details of TV^2 method is more realistic than detail of TV method because compared with piecewise constant color reconstructed by TV, TV^2 provides a more natural propagation of colors, which looks like "blurred" effect on original image.

(a) Original image with text (b) Pure TV inpainting, $\alpha = 0.001$ (c) Pure TV^2 inpainting, $\beta = 0.001$

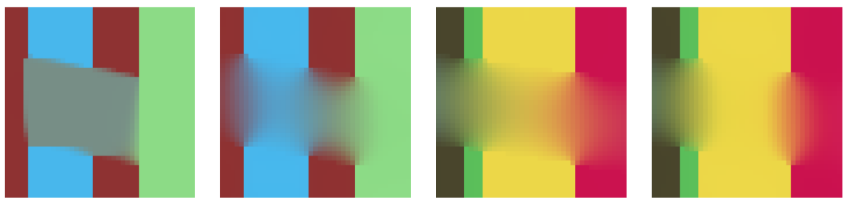
(d) Detail of original image (e) Detail of TV inpainting (f) Detail of TV^2 inpainting

Figure 6: (followed from Papafitsoros et al.(2013), page 126[4]) This example shows the performance of removing large font text from an image. Subjectively, we believe that the pure TV^2 result $(c), (f)$ looks more realistic since the piecewise constant TV reconstruction $(b), (e)$ is not desirable for natural images.

Now see another example of coloured stripes inpainting.



(a) Original image. (b) Harmonic inpainting, $\alpha = 0.001$ (c) Pure TV inpainting, $\alpha = 0.001$ (d) Pure TV^2 inpainting, $\beta = 0.001$



(e) Detail of TV inpainting (f) Detail of TV^2 inpainting (g) Detail of harmonic inpainting (h) Detail of TV^2 inpainting

Figure 7: (followed from Papafitsoros et al.(2013), page 126[4]) Coloured stripes inpainting. Observe the difference between TV and TV^2 inpainting in the light blue stripe at the left, figures (e) and (f), where in the TV^2 case the blue colour is propagated inside the inpainting domain.

Based on original image (a), comparing (c) and (d), pure TV^2 inpainting has smoother filling than TV's inpainting globally, and both methods seem to preserve edges well. Compare (e) and (f), we can find that TV inpainting failed to reconstruct the edge between blue and red strip because red strip is too thin. However, pure TV^2 inpainting propagates the blue,red, and green color inside the inpaint domain, and the blur effect of different colors shows the existence of edge to some degree.

Large Gap Connection

One inevitable problem for traditional TV method is that it cannot connect a large gap. This phenomenon can be explained intuitively: given a large gap, filling the gap will result in a large local value of total variation. By testing with different parameters of α and β (see Figure 8), we can see that TV^2 inpainting can has ability to connect the large gap.

(a) Inpainting domain (b) $\alpha = 0$, $\beta = 0.01$ (c) $\alpha = 0.008$, $\beta = 0.002$ (d) $\alpha = 0.006$, $\beta = 0.004$

(e) $\alpha = 0.005$, $\beta = 0.005$ (f) $\alpha = 0.004$, $\beta = 0.006$ (g) $\alpha = 0.002$, $\beta = 0.008$ (h) $\alpha = 0$, $\beta = 0.01$

Figure 8: (followed from Papafitsoros et al.(2013), page 127) This figure shows the experiments on effects of TV^2 inpainting for large gap connection. From (a) to (h), α is decreasing and β is increasing.

Remark: (b) has typo in the original figure, and it should be with parameters $\alpha = 0.01, \beta = 0$.

Observe that the connectedness is achieved when β/α is large, see (g) and (h). When $\beta/\alpha \leq 1$, no connectedness is obtained.

8 Conclusion

In this project, we introduce the image inpainting problem and general idea to solve it by applying total variation method. We propose one convex optimization technique called Bregman Iteration and analyze theories behind it, then we show the Split Bregman algorithm to solve the type of optimization problem where we meet in TV method. In the remaining part, we introduce two numerical ways on TV method and TV^2 method, with their algorithms, examples, evaluations.

References

[1] Aubert, G. & Kornprobst, P. Mathematical Problems in Image Processing. (2002,1)

[2] Boyd, S. Alternating Direction Method of Multipliers.

[3] Getreuer, P. Notes on Bregman Iteration.

[4] Getreuer, P. Total Variation inpainting using Split Bregman. *Image Processing On Line*. **2** pp. 147-157 (2012), https://doi.org/10.5201/ipol.2012.g-tvi

[5] Goldstein, T. & Osher, S. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal On Imaging Sciences*. **2**, 323-343 (2009)

[6] Papafitsoros, K., Schoenlieb, C. & Sengul, B. Combined First and Second Order Total Variation inpainting using Split Bregman. *Image Processing On Line*. **3** pp. 112-136 (2013), https://doi.org/10.5201/ipol.2013.40