# Optimization methods of image inpainting

Name: Shengnan Miao RIN: 661969553

### **Abstract**

#### 1. Problems

mage inpainting [2, 10, 20, 38] is the procesting data in a designated region of a still or lications range from removing objects from suching damaged paintings and photograph produce a revised image in which the it is seamlessly merged into the image in a detectable by a typical viewer. Traditionall been done by professional artists? For photographs or scratches and dust spots in filternove elements (e.g., removal of stamped of from photographs, the infamous "airbrushi enemies [201]. A convent artists area of n



Figure: Left: A contaminated image. Right: Ground truth.

Let  $f: \mathbb{R}^2 \to \mathbb{R}$  be an image contaminated in a domain  $\Omega^c$ . Reconstruct the image based on a the correct intensity of f on  $\Omega$  by solving the following convex optimization

$$\min_{u} R(u), \text{ s.t. } u(x) = f(x), \ \forall x \in \Omega$$

where *R* can be chosen as wavelet tight frame transformation or total variation

# 2. Methodology

#### 2.1 Model

Since we can choose different objective functions as regularization R(u) and they can have different performance, here I list 3 methods.

- 1. Wavelet tight frame transformation: build a sparse vector Wu as R(u) Idea: To smooth the contaminated area of the image, we want to find a sparse vector as Wu under certain overcomplete basis, in this assumption, we need do  $\min_{u} ||Wu||_{0}$ , s.t. Au = b. Since computation of  $\min_{u} ||Wu||_{0}$  is complicated, we can relax the problem to  $\min_{u} ||Wu||_{1}$ , s.t. Au = b, here W is wavelet tight frame transformation, u is the image vector, whose magnitude represents an image. Au = b exhibits the constraint.
- 2. Total variation: 1-norm of Gradient defined as R(u) Idea: To smooth the contaminated area of the image, we can minimize the total variation of the image. Here we define the total variation as the integration of  $\|\nabla u\|_1$  on the whole image.

$$\min_{u} \int_{\Omega + \Omega_{c}} \|\nabla u(x)\|_{1} dx, \text{ s.t. } u(x) = f(x), \ \forall x \in \Omega$$

here u(x) is the image function  $\mathbb{R}^2 \to \mathbb{R}$ , whose value represents an image.

3. Total variation: 2-norm of Gradient defined as R(u) Idea: To smooth the contaminated area of the image, we can minimize the total variation of the image. Here we define the total variation as the integration of  $\|\nabla u\|_2^2$  on the whole image.  $\min_{u} \int_{\Omega + \Omega_c} \|\nabla u(x)\|_2^2 dx$ , s.t. u(x) = f(x),  $\forall x \in \Omega$ 

here u(x) is the image function  $\mathbb{R}^2 \to \mathbb{R}$ , whose value represents an image.

# 2.2 Algorithm

## Method 1: Wavelet tight frame transformation

In this method, I used Augmented Lagrangian method and Alternative directional minimization method  $(ADMM)^{[1][2]}$  for the outer loop, steepest gradient and proximal gradient descent<sup>[1][3]</sup> in the inner loop. The details of derivation of algorithm for method 1 is as following:

First, variable splitting: introduce an auxiliary variable Q = Wx, define a problem:  $\min_{x,Q} \|Q\|_1$  s.t. Ax = b & Wx = Q.

$$L_{\mu}(x,Q;v_{1},v_{2}) = \|Q\|_{1} + \langle v_{1}, b - Ax \rangle + \frac{\mu}{2} \|Ax - b\|_{2}^{2} + + \langle v_{2}, Q - Wx \rangle + \frac{\mu}{2} \|Wx - Q\|_{2}^{2};$$

Outer loop: Use ADMM method:  $\begin{cases} x^{k+1} \leftarrow arg \min_{x} L_{\mu}(x; Q^{k}; v_{1}^{k}, v_{2}^{k}) \\ Q^{k+1} \leftarrow arg \min_{Q} L_{\mu}(x^{k+1}; Q; v_{1}^{k}, v_{2}^{k}) \\ v_{1}^{k+1} = v_{1}^{k} + \rho(b - Ax^{k+1}) \\ v_{2}^{k+1} = v_{2}^{k} + o(O^{k+1} - Wx^{k+1}) \end{cases}$ 

**Inner loop**-- x-sub problem: Apply iterative method to solve Hx = C, where  $H = \mu A^T A + \mu I$ ,  $C = A^T (v_1^k + \mu b) + W^T (v_2^k + \mu Q^k)$ . So we need to do  $\min_x \frac{1}{2} x^T Hx - x^T C$ , then I apply steepest gradient descent method.

Steepest gradient descent:  $\begin{cases} r^k = Hx^k - C \\ \alpha^k = \|r^k\|_2^2/(r^k)^T Hr^k \\ x^{k+1} = x^k - \alpha^k r^k \end{cases}$ 

Q-sub problem:  $Q^{k+1} = arg\min_{\alpha} \|Q\|_1 + \frac{\mu}{2} \|Q - (Wx^{k+1} - \frac{1}{\mu}v_2^k)\|_2^2 = shrink(Wx^{k+1} - \frac{1}{\mu}v_2^k, \frac{1}{\mu}).$ 

#### \*Pseudo code:

**Step 0.** Set  $x^0$ ,  $Q^0$ ,  $v_1^0$ ,  $v_2^0$  as 0 vector. Choose  $\mu$ ,  $\rho$ .

(Outer loop) for L=0 to 100

**Step 1.** While  $||r^k||_{\infty} > tol = 1e^{-6}$  (Inner loop)

$$r^{k,L} = Hx^{k,L} - C.$$

$$\alpha^{k,L} = \|r^{k,L}\|_{2}^{2} / (r^{k,L})^{T} H r^{k,L},$$
  
$$x^{k+1} = x^{k,L} - \alpha^{k,L} r^{k,L}.$$

End

$$x^{1,L+1} = x^{k,L}$$

Step 2. 
$$Q^{L+1} = shrink(Wx^{1,L+1} - \frac{1}{\mu}v_2^L, \frac{1}{\mu}).$$

Step 3. 
$$v_1^{L+1} = v_1^L + \rho(b - Ax^{L+1});$$
  
 $v_2^{L+1} = v_2^L + \rho(Q^{L+1} - Wx^{L+1});$ 

end while

# Method 2: Total variation—1 norm of gradient

In this method, I used Augmented Lagrangian method and Alternative directional minimization method (ADMM) for the outer loop, steepest gradient and proximal gradient in the inner loop. The details of derivation of algorithm for method 2 is as following:

First, variable splitting: introduce an auxiliary variable  $Q = (D_1 - D_2)x$ , define a problem:  $\min_{x,Q} \|Q\|_1$  s.t.  $Ax = b \& (D_1 - D_2)x = Q$ . Where x is a 256× 256 row, 1 column-image vector, Q is a reshaped vector consists x-gradient and y-gradient generated by the Kernel<sup>[4]</sup>  $\begin{bmatrix} 1 & -1 \end{bmatrix}$  and  $\begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$ , convoluted with the image matrix.

$$L_{\mu}(x,Q;v_{1},v_{2}) = \|Q\|_{1} + \langle v_{1}, b - Ax \rangle + \frac{\mu}{2} \|Ax - b\|_{2}^{2} + + \langle v_{2}, Q - (D_{1} - D_{2})x \rangle + \frac{\mu}{2} \|(D_{1} - D_{2})x - Q\|_{2}^{2};$$

$$\begin{aligned} \textbf{Outer loop:} \text{ Use ADMM method:} & \begin{cases} & x^{k+1} \leftarrow arg \min_{x} L_{\mu} \big( x; Q^{k}; v_{1}{}^{k}, v_{2}{}^{k} \big) \\ & Q^{k+1} \leftarrow arg \min_{Q} L_{\mu} \big( x^{k+1}; Q; v_{1}{}^{k}, v_{2}{}^{k} \big) \\ & v_{1}{}^{k+1} = v_{1}{}^{k} + \rho (b - Ax^{k+1}) \\ & v_{2}{}^{k+1} = v_{2}{}^{k} + \rho (Q^{k+1} - (D_{1} - D_{2})x^{k+1}) \end{cases} \end{aligned}$$

Inner loop-- x-sub problem: Apply iterative method to solve Hx=C, where  $H=\mu A^TA+\mu(D_1-D_2)^T(D_1-D_2)$ ,  $C=A^T(v_1^k+\mu b)+(D_1-D_2)^T(v_2^k+\mu Q^k)$ . So we need to do  $\min_x \frac{1}{2}x^THx-x^TC$ , then I apply steepest gradient descent method.

Steepest gradient descent: 
$$\begin{cases} r^k = Hx^k - C \\ \alpha^k = \left\| r^k \right\|_2^2 / (r^k)^T H r^k \\ x^{k+1} = x^k - \alpha^k r^k \end{cases}$$

Q-sub problem: 
$$Q^{k+1} = arg \min_{\alpha} \|Q\|_1 + \frac{\mu}{2} \|Q - ((D_1 - D_2)x^{k+1} - \frac{1}{\mu}v_2^k)\|_2^2 = shrink((D_1 - D_2)x^{k+1} - \frac{1}{\mu}v_2^k, \frac{1}{\mu}).$$

#### \*Pseudo code:

**Step 0.** Set  $x^0$ ,  $Q^0$ ,  $v_1^0$ ,  $v_2^0$  as 0 vector. Choose  $\mu$ ,  $\rho$ .

(Outer loop) for L=0 to 20

**Step 1.** While 
$$||r^k||_{\infty} > tol = 1e^{-6}$$
 (Inner loop)

$$r^{k,L} = Hx^{k,L} - C.$$

$$\alpha^{k,L} = \|r^{k,L}\|_{2}^{2}/(r^{k,L})^{T}Hr^{k,L},$$

$$x^{k+1} = x^{k,L} - \alpha^{k,L} r^{k,L}.$$

End

$$\chi^{1,L+1} = \chi^{k,L}$$

Step 2. 
$$Q^{L+1} = shrink((D_1 - D_2)x^{1,L+1} - \frac{1}{\mu}v_2^L, \frac{1}{\mu}).$$

Step 3. 
$$v_1^{L+1} = v_1^L + \rho(b - Ax^{L+1});$$

$$v_2^{L+1} = v_2^L + \rho(Q^{L+1} - (D_1 - D_2)x^{L+1});$$

end while

#### Method 3: Total variation—2 norm of gradient

In this method, I used Augmented Lagrangian method and Alternative directional minimization method (ADMM) for the outer loop and steepest gradient in the inner loop. The details of derivation of algorithm for method 3 is as following:

First, variable splitting: introduce an auxiliary variable  $Q = (D_1 - D_2)x$ , define a problem:  $\min_{x,Q} \|Q\|_2^2$  s.t.  $Ax = b \& (D_1 - D_2)x = Q$ . Where x is a 256× 256 row, 1 column-image vector, Q is a reshaped vector consists x-gradient and y-gradient generated by the Kernel  $\begin{bmatrix} 1 & -1 \end{bmatrix}$  and  $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ , convoluted with the image matrix.

$$L_{\mu}(x,Q;v_{1},v_{2}) = \|Q\|_{2}^{2} + \langle v_{1}, b - Ax \rangle + \frac{\mu}{2} \|Ax - b\|_{2}^{2} + + \langle v_{2}, Q - (D_{1} - D_{2})x \rangle + \frac{\mu}{2} \|(D_{1} - D_{2})x - Q\|_{2}^{2};$$

 $\begin{aligned} \textbf{Outer loop:} \text{ Use ADMM method:} & \begin{cases} & x^{k+1} \leftarrow arg \min_{x} L_{\mu} \big( x; Q^{k}; v_{1}{}^{k}, v_{2}{}^{k} \big) \\ & Q^{k+1} \leftarrow arg \min_{Q} L_{\mu} \big( x^{k+1}; Q; v_{1}{}^{k}, v_{2}{}^{k} \big) \\ & v_{1}{}^{k+1} = v_{1}{}^{k} + \rho (b - Ax^{k+1}) \\ & v_{2}{}^{k+1} = v_{2}{}^{k} + \rho (Q^{k+1} - (D_{1} - D_{2})x^{k+1}) \end{cases} \end{aligned}$ 

Inner loop-- x-sub problem: Apply iterative method to solve Hx=C, where  $H=\mu A^TA+\mu(D_1-D_2)^T(D_1-D_2)$ ,  $C=A^T({v_1}^k+\mu b)+(D_1-D_2)^T({v_2}^k+\mu Q^k)$ . So we need to do  $\min_x \frac{1}{2}x^THx-x^TC$ , then I apply steepest gradient descent method.

Steepest gradient descent: 
$$\begin{cases} r^k = Hx^k - C \\ \alpha^k = \left\| r^k \right\|_2^2 / (r^k)^T H r^k \\ x^{k+1} = x^k - \alpha^k r^k \end{cases}$$

Q-sub problem: 
$$Q^{k+1} = arg\min_{\alpha} \|Q\|_{2}^{2} + \frac{\mu}{2} \|Q - ((D_{1} - D_{2})x^{k+1} - \frac{1}{\mu}v_{2}^{k})\|_{2}^{2} = \frac{2}{2+\mu} (\mu(D_{1} - D_{2})x^{k+1} - v_{2}^{k})$$

## \*Pseudo code:

**Step 0.** Set  $x^0$ ,  $Q^0$ ,  $v_1^0$ ,  $v_2^0$  as 0 vector. Choose  $\mu$ ,  $\rho$ .

(Outer loop) for L=0 to 5

**Step 1.** While 
$$||r^k||_{\infty} > tol = 1e^{-6}$$
 (Inner loop)

$$r^{k,L} = Hx^{k,L} - C.$$

$$\alpha^{k,L} = \|r^{k,L}\|_{2}^{2} / (r^{k,L})^{T} H r^{k,L},$$

$$x^{k+1} = x^{k,L} - \alpha^{k,L} r^{k,L}.$$

End

$$x^{1,L+1} = x^{k,L}$$

Step 2. 
$$Q^{L+1} = \frac{2}{2+\mu} (\mu (D_1 - D_2) x^{L+1} - v_2^L).$$

Step 3. 
$$v_1^{L+1} = v_1^L + \rho(b - Ax^{L+1});$$
 
$$v_2^{L+1} = v_2^L + \rho(Q^{L+1} - (D_1 - D_2)x^{L+1});$$

end while

## 3 Results

Now use our model and algorithm to do the inpainting, these are the original contaminated images. The performance of each method is as followings.

This book is about crosses optimination, a special class of the problems, which includes least-expanse and linear price is well known that least-expanse and linear programmin complete theory, wise in a variety of applications, and a very efficiently. The basic point of the book is that the larger class of convex optimination problems.

While the mathematics of convex optimization has to century, several related remain devolutements have attend topic. The first is the rangelition that interior-point in 1980s to solve linear programming problems, can be used then problems as well. These new mathems allow us to of convex optimization problems, such as semidefinite prome programs, almost as easily as hear programs.

The exceed development is the decreasy that convex (beyond least-equates and linear programs) are more power previously throught. Since 1990 many explications are such as extension of manifestime, and activates electronic creat design, data training and financial optimization and observation electronic training has a found breaterial optimization and blood optimization, where it

The examil development is the discovery that cause of the projects, which includes least-squares and linear programming complete theory, arise in a variety of applications, and overy efficiently. The besic point of this book is that the larger class of convex optimisation involutions.

While the mathematics of convex optimisation has been related recognition that interior-point not upon. The first is the recognition that interior-point not 1960s to solve linear programming problems, can be used time problems as well. There are muchods allow us to of convex optimisation problems, such as remidefinite process programs, almost an easily as history programs.

The recond development in the discovery that convex (beyond least-squares and linear programs) are more programs are accounted to thought. Since 1990 many applications are south as actionatic control systems, estimation and manifestions and action to, electronic circuit design, data statistics, and factors. Conventor, electronic circuit design, data

Figure 2: Left:Barbara,

Right: Cameraman.

# Method 1: Wavelet tight frame transformation





The computation time for these figures are 128.504828s and 108.504828s.

Method 2: Total variation—1 norm of gradient





The computation time for these figures are 78.504828s and 70.365029s.

Method 3: Total variation—2 norm of gradient





The computation time for these figures are 30.866046s and 24.085837s.

# 4 Observation and Conclusions

From the results of three methods, we can see that:

1. For the method 1, which uses wavelet tight frame transformation to build a sparse vector Wu as R(u):

Advantages: it will not lose the sharpness of the image.

Disadvantages: the contamination was not totally removed; the iteration number is large and thus computation time is long comparing to other methods.

- To some degree, the performance of method 1 depends on the choice of initialization and iteration number.
- 2. For the method 2, which defines total variation as 1-norm of Gradient and use it as R(u): Advantages: the contamination was removed cleanly; the image is smoother and the computation is fast.
  - Disadvantages: it loses some sharpness of the image.
  - The performance of method 2 is not sensitively dependent on the choice of initialization and iteration number.
- 3. For the method 3, which defines total variation as 1-norm of Gradient and use its square as R(u):
  - Advantages: the contamination was removed cleanly; the loss of sharpness is not obvious, and the computation is very efficient.
  - The performance of method 3 is stable and not sensitively dependent on the choice of initialization and iteration number.

Conclusion: we can do good inpainting image if we choose suitable regularization R(u) as objective functions and use the knowledge of convex optimization to design the algorithm.

#### References

- [1] Mokhtar S. Bazaraa, Hanif D. Sherali, C. M. Shetty-- Nonlinear Programming: Theory and Algorithm
- [2] Lecture note 6 from "Introduction of optimization" by Prof. Rongjie Lai.
- [3] Lecture note 3 from "Introduction of optimization" by Prof. Rongjie Lai.
- [4] Wikipedia: Kernel (image processing)

https://en.wikipedia.org/wiki/Kernel (image processing)

## Appendix

(see attached codes)