Fast L_0 Gradient Optimization

Fanyu Geng

Given an input image I with pixel dimension d (d being 1 or 3), we compute an output image S by L_0 minimization.

$$\min_S \sum_p [\frac{\omega}{2}(S_p - I_p)^2 + \|\nabla S_p\|_{2,0}] \tag{1}$$

where $\nabla S_p = (S_x, S_y) \in \mathbb{R}^{2\times d}$, and

$$\|X\|_{2,0} = \begin{cases} 0 & \text{all coordinate of } X \text{ is } 0 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Here is some signal of this passage: in an input image I, m, n is I's heigt and width. P is the set of all position of pixel in I, in other words, $P = \{(i,j)|i,j \in \mathbb{Z}^+, 1 \le i \le m, 1 \le j \le n\}$. For a set A of $m \times n$ elemets, A^u, A^d is the subset of A without first and last row, A^l, A^r is the subset of A without first and last column, $\widetilde{A}$ is seubset without both last row and last column. For example:

$$\begin{aligned}
P^u &= \{(i,j)|i,j \in \mathbb{Z}^+, 1 \le i \le m-1, 1 \le j \le n\} \\
P^d &= \{(i,j)|i,j \in \mathbb{Z}^+, 2 \le i \le m, 1 \le j \le n\} \\
P^l &= \{(i,j)|i,j \in \mathbb{Z}^+, 1 \le i \le m, 1 \le j \le n-1\} \\
P^r &= \{(i,j)|i,j \in \mathbb{Z}^+, 1 \le i \le m, 2 \le j \le n\} \\
\widetilde{P} &= \{(i,j)|i,j \in \mathbb{Z}^+, 1 \le i \le m-1, 1 \le j \le n-1\}
\end{aligned}$$

Our method is minimization of

$$\min_S \sum_p \{\frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \|\nabla S_p\|_{2,0}\} \tag{3}$$

Then for each pixel $p \in P$, we introduce auxiliary variables $Z_{p,x}$ and $Z_{p,y}$. Our aim is let $S_{p,x} = Z_{p,x}$,$S_{p,y} = Z_{p,y}$,and $S_{p,x}, S_{p,y}, Z_{p,x}, Z_{p,y}$ satisfies that gradient field curl equal 0.

We can rewrite the optimization problem as

$$\begin{gathered}
\min_{S_x,S_y,Z_x,Z_y} \sum_{p\in\widetilde{P}} \{\frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \|\nabla Z_p\|_{2,0}\} \\
s.t. S_x = Z_x, S_y = Z_y, S_x^d - S_y^r = Z_x^u - Z_y^l
\end{gathered} \tag{4}$$

Then we can rewrite the optimization problem as

$$\begin{aligned}
\min_{S_x,S_y,Z_x,Z_y} & \sum_{p\in\widetilde{P}} \{\frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \|\nabla Z_p\|_{2,0}\} \\
& + \sum_{p\in\widetilde{P}} \frac{\mu}{2}(\|\nabla S_p - \nabla Z_p\|^2 + \|S_{x,p+(0,1)} - S_{y,p+(1,0)} - Z_{x,p} + Z_{y,p}\|)
\end{aligned} \tag{5}$$

We assume the last term as $AS - BZ$. $S = (S_x, S_y)^T$, $Z = (Z_x, Z_y)^T$, $A, B \in \mathbb{R}^{(2mn-m-n)\times(3mn-2m-2n+1)}$. Such that $AS = (S_x, S_y, S_x^d - S_y^r)^T, BZ = (Z_x, Z_y, Z_x^u - Z_y^l)^T$. We solve this problem using the ADMM algorithm. First we derive its augmented Lagrangian function as

$$L(S, Z, \lambda; \mu) = \\ \sum_{p\in\widetilde{P}} \{\frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \|\nabla Z_p\|_{2,0}\} + \lambda^T(AS - BZ) + \frac{\mu}{2}\|AS - BZ\|_2^2 \tag{6}$$

Divide λ as $\lambda = (\lambda_x, \lambda_y, \lambda_t)^T$.

We search for a saddle point of L by alternating between the following steps:

1. Fix X, λ, Update Z subproblem 1.:

$$\min_Z \sum_{p\in\widetilde{P}} \|\nabla Z_p\|_{2,0} + \frac{\mu}{2}\|AS^k + \lambda^k - BZ\|^2 \tag{7}$$

For each pixel, subproblem is

$$\min_{p\in\widetilde{P}, Z_{p,x}, Z_{p,y}} \|(Z_{p,x}, Z_{p,y})\|_{2,0} + \frac{\mu}{2}[(Z_{p,x} - S_{p,x}^k + \lambda_x^k)^2 + (Z_{p,y} - S_{p,y}^k + \lambda_y^k)^2 + (Z_{p,x} - Z_{p,y} - S_{p+(0,1),x}^k + S_{p+(1,0),y}^k + \lambda_t^k)^2] \tag{8}$$

to calculate this problem, we introduce matrix A_z, B_z, C_z.

$$\begin{aligned} A_z &= S_x^{k,u} + \lambda_x^k \\ B_z &= S_y^{k,l} + \lambda_y^k \\ C_z &= S_x^{k,d} - S_y^{k,r} + \lambda_t^k \end{aligned} \tag{9}$$

Solution of subproblem 1. is

$$(Z_{p,x}^{k+1}, Z_{p,y}^{k+1}) = \begin{cases} (\frac{2A_{p,z}+B_{p,z}+C_{p,z}}{3}, \frac{A_{p,z}+2B_{p,z}-C_{p,z}}{3}) & \text{if } A_{p,z}^2 + B_{p,z}^2 + C_{p,z}^2 - \frac{(A_{p,z}-B_{p,z}-C_{p,z})^2}{3} > \frac{\mu}{2} \\ (0, 0) & \text{otherwise} \end{cases} \tag{10}$$

2. Fix Z, λ, Update X subproblem 2.

$$\min_S \sum_{p\in\widetilde{P}} \frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \frac{\mu}{2}\|AS + \lambda^k - BZ^k\|^2 \tag{11}$$

For each pixel, subproblem is

$$\min_{p\in\widetilde{P}, S_{p,x}, S_{p,y}} \frac{\omega}{2}[(S_{p,x} - I_{p,x})^2 + (S_{p,y} - I_{p,y})^2] + \frac{\mu}{2}[(Z_{p,x} - S_{p,x}^k + \lambda_x^k)^2 + (Z_{p,y} - S_{p,y}^k + \lambda_y^k)^2 + (Z_{p,x}^k - Z_{p,y}^k - S_{p+(0,1),x} + S_{p+(1,0),y} + \lambda_t^k)^2] \tag{12}$$

to calculate this problem, we introduce matrix A_s, B_s, C_s.

$$\begin{aligned} A_s &= \omega I_x^{k,d} + \mu(Z_x^{k,d} - \lambda_x^k) \\ B_s &= \omega I_y^{k,r} + \mu(Z_y^{k,r} - \lambda_y^k) \\ C_s &= \mu(Z_x^{k,u} - Z_y^{k,l} - \lambda_t^k) \end{aligned} \tag{13}$$

Solution of subproblem 2. is

$$(S^{k+1}_{p+(0,1),x}, S^{k+1}_{p+(1,0),y}) = (\frac{(\omega+2\mu)A_{p,s} + \mu B_{p,s} + (\omega+\mu)C_{p,s}}{(\omega+\mu)(\omega+3\mu)}, \frac{\mu A_{p,s} + (\omega+2\mu)B_{p,s} - (\omega+\mu)C_{p,s}}{(\omega+\mu)(\omega+3\mu)}) \tag{14}$$

3. Fix X, Z, Update λ

$$\lambda^{k+1} = \lambda^k + AS^k - BZ^k \tag{15}$$

Termination Criteria We terminate the iteration when the following primal and dual residuals are both small enough:

$$r^k = AS^k - BZ^k \tag{16}$$

$$s^k = \mu A^T B(Z^{k+1} - Z^k) \tag{17}$$

Reconstruction Once we obtain the solution of S_x, S_y, we can reconstruct the image by solving minimization:

$$\min_S \{\sum_p (I_p - S_p)^2 + K[(\partial_x S - S_x)^2 + (\partial_y S - S_y)^2]\} \tag{18}$$

solution of above minimization is:

$$S = \mathscr{F}^{-1}(\frac{\mathscr{F}(I) + K(\mathscr{F}(\partial_x)\mathscr{F}(S_x) + \mathscr{F}(\partial_y)\mathscr{F}(S_y))}{\mathscr{F}(1) + K(\mathscr{F}(\partial_x)\mathscr{F}(\partial_x) + \mathscr{F}(\partial_y)\mathscr{F}(\partial_y))}) \tag{19}$$

Then ADMM algorithm of L_0 Smoothing is:

Algorithm 1 ADMM algorithm of L_0 Smoothing

Input: input image I
Output: output image S

1: calculate I_x, I_y
2: set $S_x = Z_x = I_x$,$S_y = Z_y = I_y$,$\lambda = \mathbf{0}$.
3: **while** $\|r^k\|_2 \geq \epsilon^{pri} \| \| \|s^k\|_2 \geq \epsilon^{dual}$ **do**
4: Update Z
5: Update S
6: Update λ
7: Update r^k, s^k
8: Reconstruct S by S_x, S_y.**return** S.

Penalty Method is similar algrorithm without λ:

Algorithm 2 Penalty Method of L_0 Smoothing

Input: input image I

Output: output image S

1: calculate I_x, I_y

2: set $S_x = Z_x = I_x$,$S_y = Z_y = I_y$,$\mu = \mu_0$.

3: **while** $\mu < \mu_{max}$ **do**

4: Update Z

5: Update S

6: $\mu = \mu * 2$

7: Reconstruct S by S_x, S_y.**return** S.

If we change residue term $\|\nabla S - \nabla I\|_2^2$ to $\|\nabla S - \nabla I\|_2$, we need to modify the step of **Update S**. We can denote $N_s^t := \|\nabla S^t - \nabla I\|_2$, we have $\frac{\|\nabla S - \nabla I\|_2^2}{NS}$ in stead of $\|\nabla S - \nabla I\|_2$, we can have ω/NS instead of ω in step of **Update S**.