Solving Optimization Problem in Poisson Editing

Qitong Gao

1 Optimization Problem in Poisson Editing

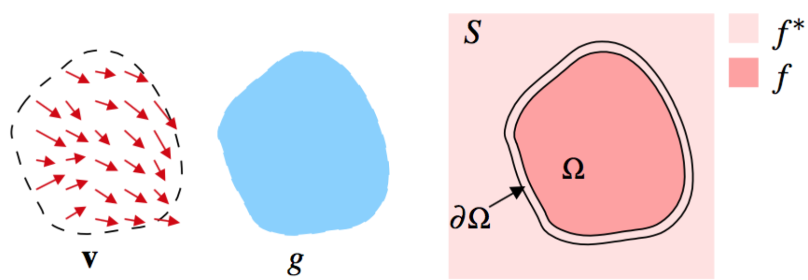

Figure 1: Notations [1]

The optimization problem is:

$$\min_f \iint_\Omega |\nabla f|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

with Dirichlet boundary conditions:

$$\triangle f = 0 \text{ over } \Omega \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

where, as shown in Figure 1, S is the image domain, f^* is a known scalar function defined over S minus Ω and f is an unknown scalar function defined over the Ω.

2 System of Linear Equations & Guidance Field

The guidance field is a vector field $\mathbf{v}$ defined over Ω. Then the optimization problem becomes to:

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{1}$$

with Dirichlet boundary conditions:

$$\triangle f = \text{div}\mathbf{v} \text{ over } \Omega \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{2}$$

The image domain, S and Ω, could be discretized into finite point sets defined on an infinite discrete grid, which is shown in Figure 2. For now we only focus on analyzing only one color channel because the different color channels (RGB) could be processed independently during the poisson processing progress. In addition, the divergence of guidance field, div$\mathbf{v}$, could be represented by a matrix, which is shown in Figure 3.

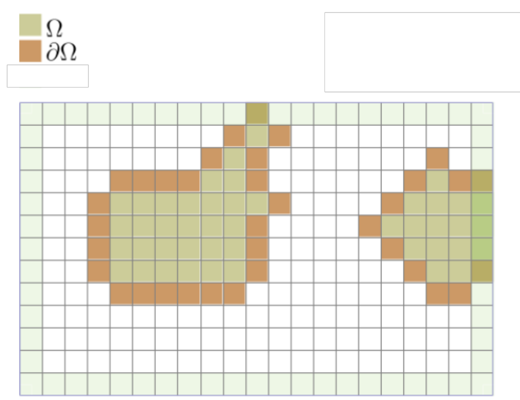

Figure 2: Visualizing the target image and the source image [2]

Expanding equation (2):

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \tag{3}$$

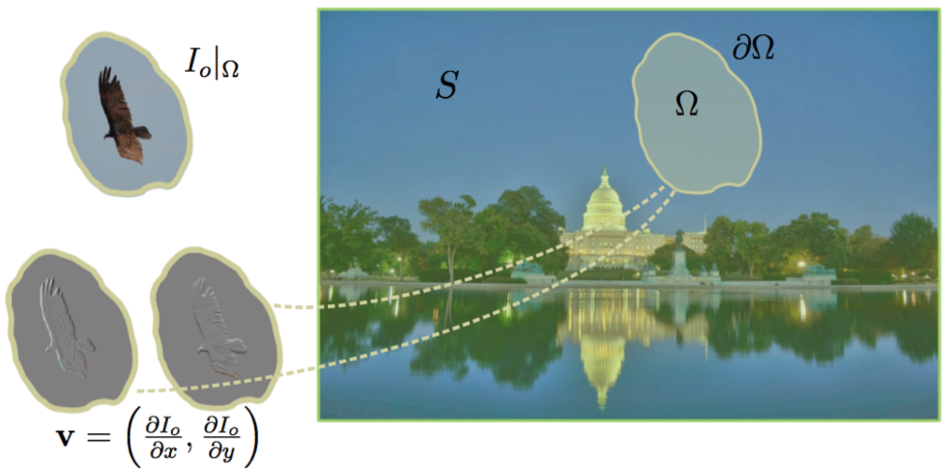

Figure 3: Visualizing divergence of $\mathbf{v}$ [2]

where $\mathbf{v} = (u, v)$. In the seamless cloning used in Homework #3, if the source image is denoted by I_0, $\mathbf{v} = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y}\right)$. Therefore, (2) becomes to:

$$\triangle f = \triangle I_0 \text{ over } \Omega \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{4}$$

and 3 becomes to:

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = \frac{\partial^2 I_0(x,y)}{\partial x^2} + \frac{\partial^2 I_0(x,y)}{\partial y^2} \tag{5}$$

By using finite differences, (5) becomes to:

$$U(i-1,j) + U(i+1,j) + U(i,j-1) + U(i,j+1) - 4U(i,j) = b(i,j) \tag{6}$$

The system of equations above has been derived for solving the optimization problem. And $(x(i), y(j))$ should be in domain Ω, where was implemented by `mIdx=find(Mask==1)` in the original code `poissonSolverJacobi.m` of Homework #3. In (6), we use $U(i,j)$ to denote the approximate solution of $f(x(i), y(j))$ and $b(i,j)$ to denote the right hand side of (5) evaluated at point $(x(i), y(j))$. So the matrix U in (6) corresponds to the `IGrad` matrix in `poissonSolverJacobi.m` and, analogously, the matrix b in (6) corresponds to the `B` matrix in the code. And we need to update matrix U iteratively until a termination criteria have been met.

3 Matrix Structure & Iterative Methods

$$
\left[\begin{array}{cccc|cccc|cccc|cccc}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
& -1 & 4 & -1 & & & -1 & & & & & & & & & \\
& & -1 & 4 & & & & -1 & & & & & & & & \\
\hline
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
& -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
& & -1 & & & -1 & 4 & -1 & & & -1 & & & & & \\
& & & -1 & & & -1 & 4 & & & & -1 & & & & \\
\hline
& & & & -1 & & & & 4 & -1 & & & -1 & & & \\
& & & & & -1 & & & -1 & 4 & -1 & & & -1 & & \\
& & & & & & -1 & & & -1 & 4 & -1 & & & -1 & \\
& & & & & & & -1 & & & -1 & 4 & & & & -1 \\
\hline
& & & & & & & & -1 & & & & 4 & -1 & & \\
& & & & & & & & & -1 & & & -1 & 4 & -1 & \\
& & & & & & & & & & -1 & & & -1 & 4 & -1 \\
& & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right]
*
\begin{bmatrix} U(1,1) \\ U(2,1) \\ U(3,1) \\ U(4,1) \\ U(1,2) \\ U(2,2) \\ U(3,2) \\ U(4,2) \\ U(1,3) \\ U(2,3) \\ U(3,3) \\ U(4,3) \\ U(1,4) \\ U(2,4) \\ U(3,4) \\ U(4,4) \end{bmatrix}
=
\begin{bmatrix} b(1,1) \\ b(2,1) \\ b(3,1) \\ b(4,1) \\ b(1,2) \\ b(2,2) \\ b(3,2) \\ b(4,2) \\ b(1,3) \\ b(2,3) \\ b(3,3) \\ b(4,3) \\ b(1,4) \\ b(2,4) \\ b(3,4) \\ b(4,4) \end{bmatrix}
$$

Figure 4: Matrix structure of discrete poisson problem [3]

As Figure 4 shows, now we convert the optimization problem into a solving $Ax = b$ problem formed by system of equations (6). Firstly, we convert matrix U, or b, to a vector by concatenating each column of U, or b, under the previous column. Secondly, we update vector of U in each iteration. In the SOR method, we need divide all the points in domain Ω into two different color sets. Moreover, as shown in Figure 5, we require black grid points are connected only to red grid points and vice versa. So the black(red) points can all be updated simultaneously using the most recent red(black) values. Furthermore, after integrating Successive Overrelaxation, the algorithm should be:

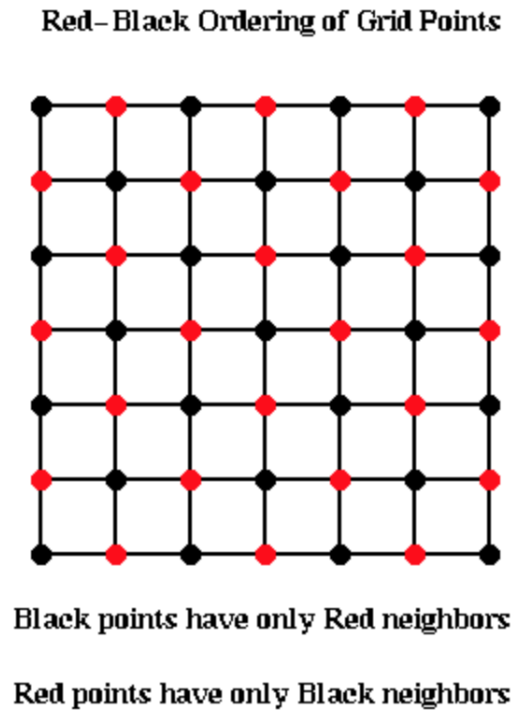

Figure 5: Black-Red Ordering of SOR [3]

`for all black (i,j) grid points`
$U(i,j,m+1) = U(i,j,m) + w(U(i-1,j,m) + U(i+1,j,m)$
$+ U(i,j-1,m) + U(i,j+1,m) - b(i,j) - 4U(i,j,m))/4$
`end for`
`for all red (i,j) grid points`
$U(i,j,m+1) = U(i,j,m) + w(U(i-1,j,m+1) + U(i+1,j,m+1)$
$+ U(i,j-1,m+1) + U(i,j+1,m+1) - b(i,j) - 4U(i,j,m))/4$
`end for`

where $U(\cdot,\cdot,m)$ denotes the structure of matrix U at the mth iteration.

4 Termination Criteria

In each iteration, the error at the mth iteration:

$$e_m \leqslant \rho[(D - wL)^{-1}((1-w)D + wU)] \cdot e_{m-1} \leqslant \rho^m[(D - wL)^{-1}((1-w)D + wU)]e_0$$

So the error decreases by a factor $\rho[(D - wL)^{-1}((1 - w)D + wU)]$ at each step. And the difference between $U(\cdot, \cdot, m)$ and $U(\cdot, \cdot, m - 1)$, denoted by $e_{m,m-1}$, would be very small if the solution is converged.

So we could set $e_{m,m-1} \leqslant \tau$ as a termination criteria and $e_{m,m-1}$ is calculated by:

$$e_{m,m-1} = \sqrt{\sum_{i,j} [U(i, j, m) - U(i, j, m - 1)]^2}$$

References

[1] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 313–318. ACM, 2003.

[2] J. Matías Di Martino, Gabriele Facciolo, and Enric Meinhardt-Llopis. Poisson Image Editing. *Image Processing On Line*, 6:300–325, 2016.

[3] Jim Demmel. https://people.eecs.berkeley.edu/ demmel/cs267/lecture24/lecture24.html.

[4] Khaled Hussain et al. Efficient poisson image editing. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 14(2):45–57, 2016.