CS172 Computer Vision I:
Homework 1 Report

Boyi Li
2018533187
liby@shanghaitech.edu.cn

Abstract

This LaTeX file is a homework report about Poisson Image Editing. The first section is about the mathematical background about the method. It proves the correctness of the method. The second section is the result of the method. And the final is the conclusion.

1. Introduction

Mathmetics is no doubt the basis of modern digital image processing technique, some outstanding and popular image processing methods make use of the achievement of classic mathematical theorem. However, these classic mathematical theorem in their original background may not strongly connected with the image processing. The application of Poisson Equation in Poisson Image Editing and Poisson Matting is one of the typical examples.

2. Background theorey

Image editing tasks concern either global changes (color/intensity corrections, filters, deformations) or local changes confined to a selection.

When mixing two images together, it is important to make the boundary between the original image and the interpolation look smooth. So we choose poisson image blending method.

We have the source image, destination image and the blended image. Let f represents the function of blended image, f^* represents the function of destination image, u represents the source image. Ω represents the part we mixed in blended image, $\partial\Omega$ represents the boundary.

We want to make the image look natural, and not like the one mixed by two, so we want the gradient of the edited part be the least. So we have such goal:

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

What's more, we want to remain the original feature of the destionation image. So we have

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

Now, we have the function

$$F = (f_x - u_x)^2 + (f_y - u_y)^2 \tag{3}$$

According to The the Euler-Lagrange equation, we have:

$$\frac{d}{dx}\left[\frac{\partial F}{\partial (f_x - u_x)^2}\right] = \frac{d}{dx}\left[2(f_x - u_x)\right] = 2\left(\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 u}{\partial x^2}\right) \tag{4}$$

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \tag{5}$$

Laplacian is the gradient of divergence:

$$\Delta f = \operatorname{div}(\nabla f) = \nabla \cdot (\nabla f) = \nabla^2 f \tag{6}$$

In the 2-D space, f is the function of x and y.

$$\Delta f = \operatorname{div}\left(\frac{\partial f}{\partial x} i, \frac{\partial f}{\partial y} j\right) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{7}$$

And it can be represented in Poisson form:

$$\Delta f = \operatorname{div}(\nabla u) \tag{8}$$

According to the equation, we can take the derivative of gradient to get divergence.

Remeber that our goal is equation (5), we want the laplacian of the blended image function f equals to the laplacian of original destination function u. From (8), we get:

$$4f_p - \sum_{q \in N_p} f_q = 4u_p - \sum_{q \in N_p} u_q \tag{9}$$

The function u is known. And using the Dirichlet boundary conditions, let the $f = u$ in all boundary pixel.

We just need to solve equation (9) in all pixel for three channels R, G and B.

mixed gradient In above equation (2), we want to minimize $\nabla f - \nabla u$. It just focus on the source image but not the original destination function. If the source image has low gradient around, the blended image will also have low gradient in corresponding location, thus causing vague in the image. Instead, we choose:

$$\text{for all } \mathbf{x} \in \Omega, \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla u(\mathbf{x})| \\ \nabla u(\mathbf{x}) & \text{otherwise} \end{cases} \tag{10}$$

Then we can get the better blended image we want.

3. Result display

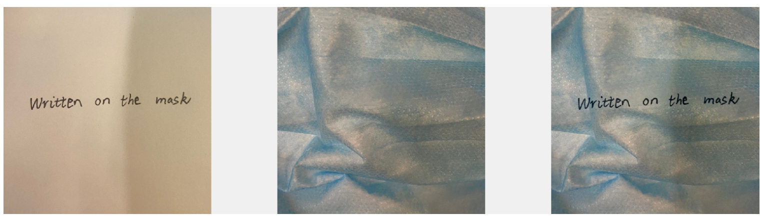

Figure 1. result1

Figure 2. result2

4. Thoughts and Conclusion

Although the paper Possion Image Editing was first published on Micaosoft Research UK even 16 years ago, it is still a paper worth reading because of its strong mathematics and excellent result even in today.

However, like Figure.2, when the destination image has big gradient in Ω, that place may not well replaced by source image. Local illumination changes may help that.

What's more, solving the final equation costs time a lot, using GPU to solve the equations may be much more efficient.

References

[1] Perez P , Gangnet M , Blake A . Poisson image editing[J]. ACM Transactions on Graphics (TOG), 2003, 22(3):p.313-318.

[2] Hao Wu, Dan Xu. Improved Poisson Image Editing and its Implementation on GPU[C]// IEEE International Conference on Computer-aided Industrial Design Conceptual Design. IEEE, 2010.