Report on Poisson Image Editing

Zhan Zhang, PB17000123

March 7, 2020

Abstract

This report are interested in the Poisson Image Editing Method witch achieved local changes, ones that are restricted to a region manually selected, in a seamless and effortless manner. The extent of the changes ranges from slight distortions to complete replacement by novel content. They proposed a generic machinery from which different tools for seamless editing and cloning of a selection region can be derived. The mathematical tool at the heart of the approach is the Poisson partial differential equation with Dirichlet boundary conditions which specifies the Laplacian of an unknown function over the domain of interest, along with the unknown function values over the boundary of the domain.

Figure 1: Display

1 Algorithm

The Poisson equation therefore has a unique solution and this leads to a sound algorithm. So, given methods for crafting the Laplacian of an unknown function over some domain, and its boundary conditions, the Poisson equation can be solved numerically to achieve seamless filling of that domain. This can be replicated independently in each of the channels of a color image. Solving the Poisson equation also has an alternative interpretation as a minimization problem: it computes the function whose gradient is the closest, in the L2-norm, to some prescribed vector field, the guidance vector field, under given boundary conditions. In that way, the reconstructed function interpolates the boundary conditions inwards, while following the spatial variations of the guidance field as closely as possible. Section 2 details this guided interpolation.

1.1 Poisson solution to guided interpolation

Let open set Ω denote the cut region, f denote the unknow function, with the boundary condition f^* at $\partial\Omega$, g be the function of cut region. Our target is to minimize the difference between the ∇f and ∇g. So, the fitting problem changes into the minimize the following function:

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

The minimizer must satisfy the associated Euler-Lagrange equation:

$$\Delta f = \Delta g \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

This is the fundamental machinery of Poisson editing of color images: three Poisson equations of this form are solved independently in the three color channels of the chosen color space. All the results reported in this paper were obtained in the RGB color space, but similar results were obtained in CIE-Lab for instance.

1.2 Discrete Poisson solver

For discrete images the problem can be discretized naturally using the underlying discrete pixel grid. Without loss of generality, we will keep the same notations for the continuous objects and their discrete counterparts. For Dirichlet boundary conditions defined on a boundary of arbitrary shape, it is best to discretize the variational problem directly, rather than the Poisson equation. The finite difference discretization of variational problem yields the following discrete, quadratic optimization problem:

$$\min_{f|_\Omega} \sum_{p\in\Omega, q\in N_p} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^* \text{ for all } p \in \partial\Omega$$

where N_p denote the set of p's 4-connected neighbors which are in $\overline{\Omega}$, and v_{pq} is the gradient between p and q. Its solution satisfies the following simultaneous linear equations:

$$|N_p| f_p - \sum_{q\in N_p} f_q = \sum_{q\in N_p} v_{pq}, \ \forall p \in \Omega$$
$$f_p = f_p^*, \ \forall p \in \partial\Omega$$

1.3 Seamless cloning

We will examine a number of possible choices for the guidance vector field. We show in particular that this interpolation machinery leverages classic cloning tools, both in terms of ease of use and capabilities. The resulting cloning allows the user to remove and add objects seamlessly. By mixing suitably the gradient of the source image with

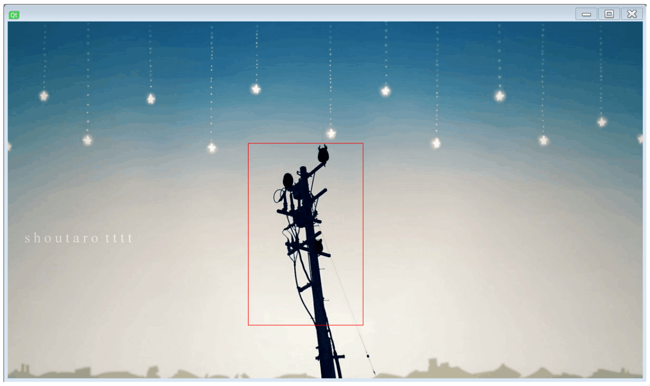

Figure 2: Cut Region

that of the destination image, it also becomes possible to add transparent objects convincingly.

Importing gradients The basic choice for the guidance field v is a gradient field taken directly from a source image. Denoting by g this source image, the interpolation is performed under the guidance of:

$$v = \nabla g$$

As for the numerical implementation, the continuous specification translates into:

$$v_{pq} = g_p - g_q$$

Mixing gradients With the tool described in the previous section, no trace of the destination image f^* is kept inside Ω. However, there are situations where it is desirable to combine properties of f^* with those of g, for example to add objects with holes, or partially transparent ones, on top of a textured or cluttered background.

The Poisson methodology allows non-conservative guidance fields to be used, which gives scope to more compelling effect. At each point of W, we retain the stronger of the variations in f^* or in g, using the following guidance field:

$$v(x) = \begin{cases} \nabla f^*(x) & \text{if } |\nabla f^*(x)| > |\nabla g(x)| \\ \nabla g(x) & \text{otherwise} \end{cases}$$

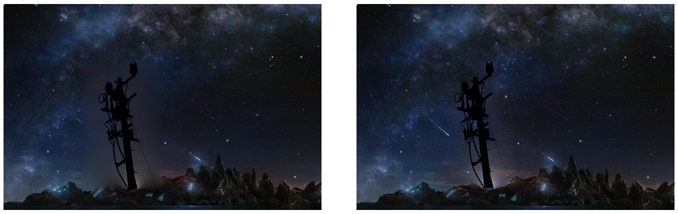

Figure 3: Importing gradients vs Mixing gradients

2 Implementation

Pictures below will show the GUI of my application:

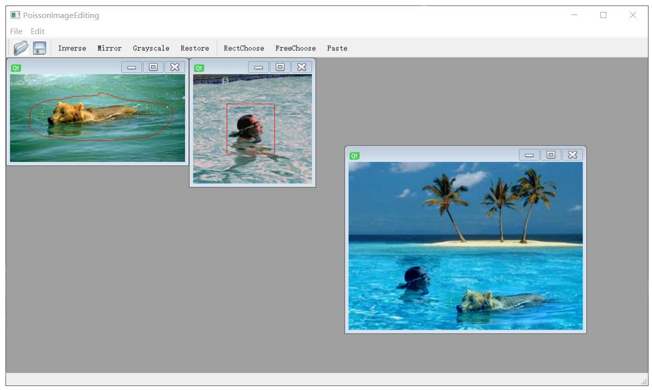

Figure 4: GUI

You can use rectangle or free hand to chose cut region. When pasting the pastregion's boundary will turn green if it is able to eidt, else the boundary will turn orange.

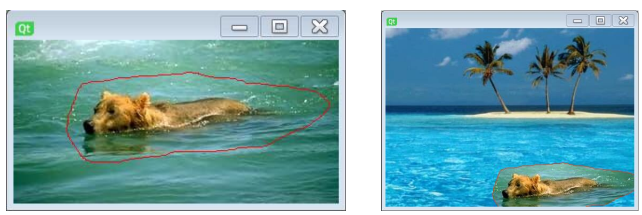

Figure 5: chose & Paste

If the cut region is small enough(pixels<1500) my application can show the after-editing image imediately while moving the paste region by count the inverse matrix first. This may consume a little bit time when pressing the mouse.

2.1 Step by Step

Here comes the sevral step of editing a picture:

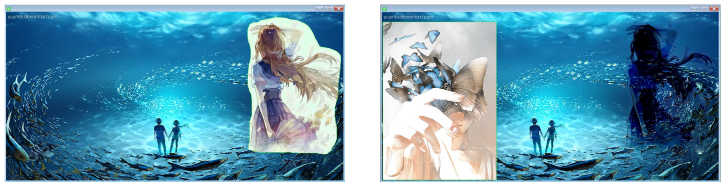

Figure 6: Step 1, 2

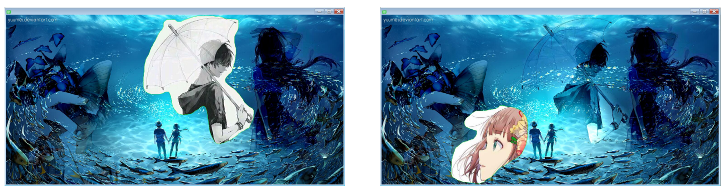

Figure 7: Step 3, 4

Then press save can save the image:

Figure 8: Final

3 Drawbacks

Although, this method preserve the gradient of the image, which means it preserve the outline of the picture, it is unable to keep the right color. So, when editing something strange will happen if the cut region and the background don't fit enough:

Figure 9: Drawbacks 1

Also, if the background's gradient changes dramaticly, the paste edition is not very good. And it depends on the order of pasting very much:

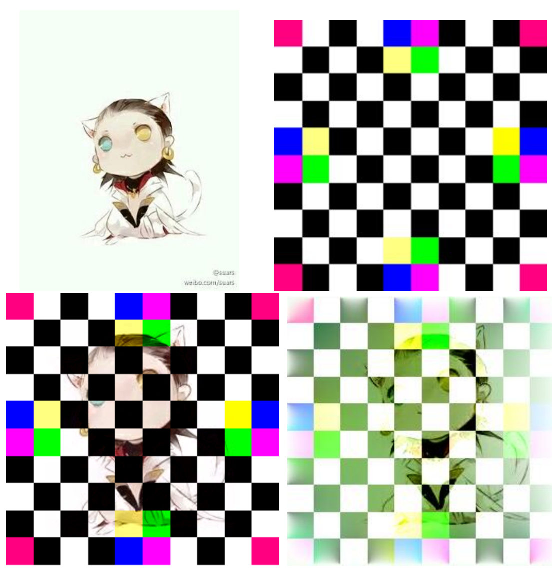

Figure 10: Drawbacks 2: A,B, A over B and B over A