CG_3_Poisson Editing Report

PB21010479 Caoliwen Wang

March 17, 2024

1 Problem Statement

Image editing tasks concern either global changes (color/intensity corrections, filters, deformations) or local changes confined to a selection. Here we are interested in achieving 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. Classic tools to achieve that include image filters confined to a selection, for slight changes, and interactive cut-and-paste with cloning tools for complete replacements. With these classic tools, changes in the selected regions result in visible seams, which can be only partly hidden, subsequently, by feathering along the border of the selected region.

We propose here 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. The motivation is twofold.

The essence of this problem is essentially an interpolation task: inserting pixels from the source image into the selected region of the target image to achieve a seamless effect. Our requirement is to minimize gradient changes, which transforms the problem into a Poisson equation problem with given boundaries. After discretization, it can be converted into a sparse matrix solving problem. Once the pixel values are obtained, they are drawn onto the target image.

2 Propose Algorithms

Let S be a closed subset of $\mathbb{R}^2$. It represents the image definition domain. Let Ω be a be a closed subset of S with boundary $\partial\Omega$. Let f^* be a known scalar function defined over S minus the interior of Ω and let f be an unknown scalar function defined over the interior of Ω. Finally, let $\boldsymbol{v}$ be a vector field defined over Ω.

The simplest interpolant f of f^* over Ω is the membrane interpolant defined as the solution of the minimization problem:

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \tag{2.1}$$

The minimizer must satisfy the associated Euler-Lagrange equation.

$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \tag{2.2}$$

According to this principle, we give the discretization of equations. For all $p \in \Omega$,

$$|N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}. \tag{2.3}$$

In equation 2.3, N_p represents a four-connected neighbor. We realize too different methods, they are only different from the value of v_{pq}. The definitions of v_{pq} are listed:

1. Seamless:
$$v_{pq} = g_p - g_q;$$
2. mixed:
$$v_{pq} = \begin{cases} f_p^* - f_q^* & \text{if } |f_p^* - f_q^*| > |g_p - g_q|, \\ g_p - g_q & \text{otherwise,} \end{cases}$$

In this way, we can construct the sparse matrix equation according to 2.3.

3 Programming

3.1 Class

One of the primary objectives of this task is to recall knowledge related to C++ classes, including inheritance and polymorphism. I write a Cloning class to become the father class, and mix, seamless, paste class are son classes. I write a construction function for father class and use this function to construct son classes directly.

3.2 Improve Code Efficiency

It's important to use functions in Eigen library in this project. We use the SparseLU solver to realize the algorithm because we provide the square matrix.

4 Experimental Results

I think the results of this algorithm are fantastic.

First, we use the standard figure to test, the results are in Figure 1.

We might not find lots of differences between them. So I give the second test, we can find the mix method has more advantages on the embedding project. The original Figures are shown in Figure 2, the outcomes are shown in Figure 3.

(a) Seamless (b) Mix

Figure 1: Standard Test

(a) Wall (b) Love

Figure 2: Another Test Original Figures

5 Summary

In this experiment, we achieved interactive image deformation, gaining a deeper understanding of the essence of image deformation and revisiting concepts such as inheritance, polymorphism, and constructors in C++ classes. We use much knowledge of PDE and its numerical solution method. However, there are several issues with the algorithms we implemented:

1. We just realize the rectangle method, the polygon case is much more sophisticated because we need to realize the searching line method. I have tried to do that but don't come up with a good idea.

2. We have realized the operation for updating in time, but it might be a little slow if we choose a large region.

(a) Seamless (b) Mix

Figure 3: Another Test Outcomes

3. If we drag the figure near the boundary, the pixel is out of range.