Implementation of Poisson Image Blending

Reza Akbarian Bafghi

Principles of Image Processing, HW5

Abstract

The process of blending two or more images in order to create a natural image is termed image composition. Poisson image blending [4] is a popular image composition approach that, seamlessly with lightly mask adjustment, blends an object from a source image onto a target image. In the blending image, not only is the gradient-domain of the blending boundary smoothed out, but also consistent texture into the blending region is added. Due to the fact that humans are susceptible to an unexpected change in intensity, this method tries to guarantee the smoothness of blended images in the gradient domain. In this project, the Poisson image blending will be described thoroughly, and a numerical implementation will be proposed to address the integration problem.

1 Introduction

The main aim of image composition approaches is improving the spatial and color consistencies between the source and target images. Some ideas of combining images have been proposed besides Poisson image blending [4]. in [1], they proposed that by applying the dense image matching method, corresponding pixels can be copied and pasted. But, this method would not work when there are significant differences between the source and target images. The other way is to make the transition as smooth as possible for hiding artifacts in the composite images. Alpha blending [5] is the simplest and fastest method, but it blurs the fine details when there are some registration errors between the source and target images.

However, methods based on the manipulation of image gradients are a powerful tool for processing or combining images. Image blending can be considered as the main building block for numerous practical applications in seamless cloning, texture flattening or seamless tiling, which can be performed in a straightforward and efficient way by combining/modifying the image gradients; for instance, in Synthetic Text in the Wild [2], they blend the text on to the base image by using this approach to maintain the illumination gradient in the synthetic text image, as illustrated in Figure 1.

The following section gives an overview of Poisson editing theory. Then, a numerical implementation is discussed, and finally, section 4 and 5 summarise the project and present the results by presenting some shortcomings of this method.

2 Poisson Editing Theory

Seamlessly cloning a region selected from an image over a background image can be formulated as the variational problem [4, 3]. Let $\mathcal{R}$ a closed subset of $\mathbb{R}^2$ and represent the image domain, $\mathscr{C}^2(\mathcal{R})$ represents the set of real functions twice differentiable over the interior of $\mathcal{R}$, f^* represents the target image, and finally, v is a differentiable gradient field obtained from the selected region as it is depicted in Figure 2.

$$\min_{\substack{f\in\mathscr{C}^2(\mathcal{R}) \\ \text{st. } f|_{\mathcal{R}\setminus\omega}=f^*|_{\mathcal{R}\setminus\Omega}}} \int_\Omega \|\nabla f - v\|^2 dx \tag{1}$$

Figure 1: Comparison between simple alpha blending and Poisson Editing. Poisson Editing preserves local illumination gradient and texture details [2].

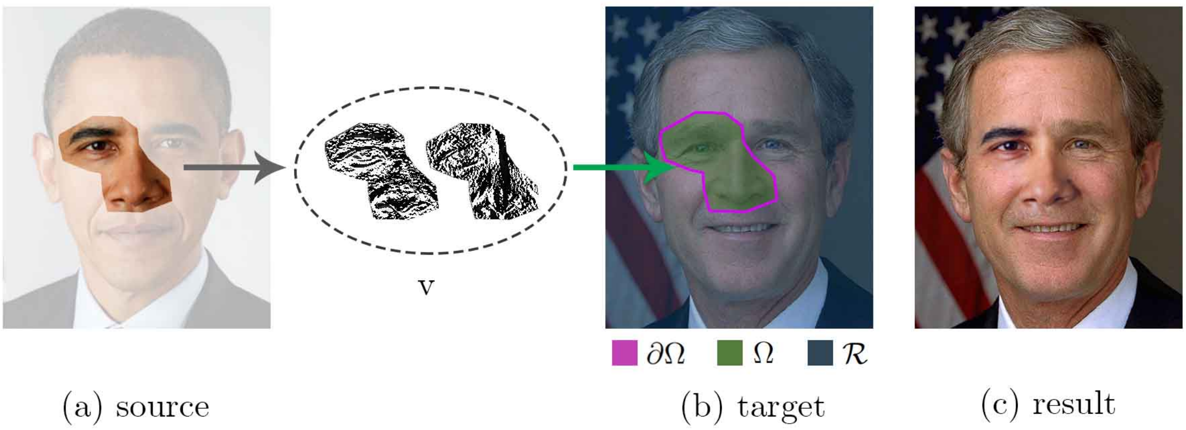

Figure 2: Illustration of the notation introduced in Equation 1 and relationship between the target image and the source one.

The solution of Equation 1 must satisfy the Euler-Lagrange equation[6]

$$\begin{aligned} \Delta f(x) &= \mathrm{div}(v(x)), \quad \forall x \in \Omega \ \text{ and } \ f|_{\mathcal{R}\setminus\Omega} = f^*|_{\mathcal{R}\setminus\Omega} \\ f(x) &= f^*(x), \quad \forall x \notin \Omega \end{aligned} \tag{2}$$

By solving Equation 2, The region of interest is filled intuitively following the variations of intensity and color that connected to the image, and at the same time, by regarding the intensities of the background image, it creates a solution. See the example illustrated in 2. Equation 2 is called the Poisson equation, which is independently solved for three channels of the image in RGB color space to obtain the interpolant f.

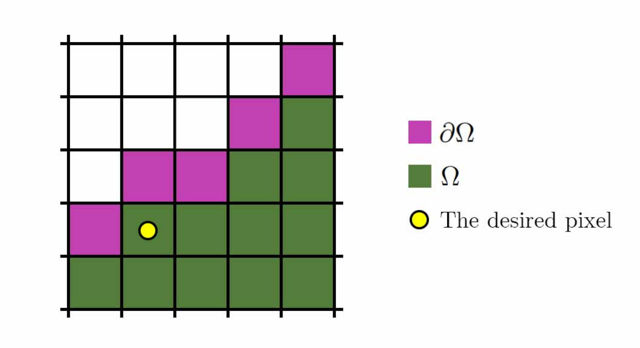

Figure 3: Pixel p interior to Ω with $|N_p| < 4$.

3 Discrete Poisson Solver

Solution developed in section 2 applies to continuous case of functions. However in real life images encountered are in discrete domain, so the solution of 2 has to be modified to suit discrete images. Without loss of generality let S and Ω be discrete grids with pixels. Let x and y denote the co-ordinates of the 2D grid. The condition in 2 reduces to

$$f(x,y) = f^*(x,y), \quad \forall (x,y) \in \partial\Omega \tag{3}$$

Let p be a pixel in S such that $p = (x,y)$, let N_p be the set of its 4-connected neighbours which are in S, and let (p, q) denote a pixel pair such that $q \in N_p$ with $q = (x_1, y_1)$. Let f_p be the value of f at p. The minimization problem of 1 in discrete domain reduces to

$$\min_f \sum_{(p,q)\in\Omega} (f_p - f_q - v_{pq})^2, \quad f_p = f_p^*, \quad \forall p \in \partial\Omega \tag{4}$$

As depicted in Figure 3, for all pixels p interior to Ω, the solution of 4 satisfies

$$\begin{aligned} \sum_{q\in N_p} v_{pq} &= |N_p|\, f_p - \sum_{q\in N_p} f_q, \quad |N_p| = 4 \\ \sum_{q\in N_p} v_{pq} &= |N_p|\, f_p - \sum_{q\in N_p} f_q - \sum_{q\in N_p\cap\partial\Omega} f_q^*, \quad |N_p| < 4 \end{aligned} \tag{5}$$

Equations 5 form a sparse, symmetric, positive-definite system. The linear system has a size $N \times N$ where N is the number of pixels in the image. The solution to 5 can be solved in an iterative manner or using an exact closed form solution. According to 2, let g denote the source image, then

$$\Delta f(x,y) = \Delta g(x,y), \quad \forall (x,y) \in \Omega \text{ and } f|_{\mathcal{R}\setminus\Omega} = f^*|_{\mathcal{R}\setminus\Omega} \tag{6}$$

Continuous differential operator is replaced by their discrete counterparts,

$$\Delta f(x,y) = 4g(x,y) - g(x-1,y) - g(x+1,y) - g(x,y-1) - g(x,y+1) \tag{7}$$

Let N is total number of all pixels contained in f (those that correspond to pixels in interior of Ω). Equation 7 can be written in the form $Ax = b$, where b is an $N \times 1$ column vector, which can be calculated for all pixels since g is known image. Also, A is a sparse matrix dimension of $N \times N$ giving rise to solving it easier.

(a) source (b) target (c) result

Figure 4: Example of the results that can be obtained by mixing the gradients of two source images

(a) source (b) target (c) result

Figure 5: One of the drawbacks associated with this method

4 Results

One of the most notable applications of the Poisson equation, besides other remarkable applications, is seamless cloning, as shown in 4.

When the source and target images have a similar background color, as observed in Figure 5, Poisson blending produces excellent results. One of the shortcomings of this method is that during the optimization of the source image's color because Poisson attempts to keep the same color contrast in a blended image as that of the original source image, color consistency is not necessarily maintained. This results in unnatural colors in some cases when source and image have significant color differences in their background, which is shown in Figure 5; dolphins that are originally gray-colored appear to have a reddish-brown color in the blended image.

5 Conclusion

In the present work, the Poisson image editing method was described in detail; besides, the integration problem was reviewed and analyzed both from a theoretical and numerical perspective in depth. Furthermore, an up-to-date review of the latest contributions on the subject was provided. For solving this equation, Discrete Poisson solver uses discrete versions of differential operators to convert the problem into a linear system of equations. Implementation of this method, in addition to its results, is publicly available on my GitHub webpage[1].

[1] `https://github.com/rezaakb/ImageProcessing/tree/master/HW5/Q1`

References

[1] Nuno Gracias, Mohammad Mahoor, Shahriar Negahdaripour, and Arthur Gleason. “Fast Image Blending using Watersheds and Graph Cuts”. In: *Image Vision Comput.* 27 (Jan. 2009), pages 597–607. https://doi.org/10.5244/C.20.49.

[2] A. Gupta, A. Vedaldi, and A. Zisserman. “Synthetic Data for Text Localisation in Natural Images”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pages 2315–2324. https://doi.org/10.1109/CVPR.2016.254.

[3] J. Martino, Gabriele Facciolo, and Enric Meinhardt-Llopis. “Poisson Image Editing”. In: *Image Processing On Line* 5 (Nov. 2016), pages 300–325. https://doi.org/10.5201/ipol.2016.163.

[4] Patrick Pérez, Michel Gangnet, and Andrew Blake. “Poisson Image Editing”. In: *ACM Trans. Graph.* 22.3 (July 2003), pages 313–318. ISSN: 0730-0301. https://doi.org/10.1145/882262.882269. https://doi.org/10.1145/882262.882269.

[5] Matt Uyttendaele, Ashley Eden, and Richard Szeliski. “Eliminating Ghosting and Exposure Artifacts in Image Mosaics.” In: volume 2. Jan. 2001, pages 509–516. https://doi.org/10.1109/CVPR.2001.991005.

[6] Wikipedia contributors. *Euler–Lagrange equation — Wikipedia, The Free Encyclopedia.* [Online; accessed 3-December-2020]. 2020. https://en.wikipedia.org/w/index.php?title=Euler%E2%80%93Lagrange_equation&oldid=984541656.