Poisson Image Editing

CSC262 Computer Vision: Yuina Iseki, Shuta Shibue

17 December 2024

Abstract

This project focuses on seamlessly blending object images into background images using Poisson image editing, a gradient-domain technique for image manipulation. The core approach involves solving the discrete Poisson partial differential equation with Dirichlet boundary conditions to reconstruct an unknown region of interest (ROI) that matches the target gradient. Our implementation utilizes a sparse matrix representation to efficiently solve the system of linear equations to calculate pixel values in the ROI. The project evaluates three primary techniques: copy-and-paste, which replaces object image pixels with background image pixels with no blending, seamless cloning, where the target gradient is derived from the object image for solid object blending, and mixed gradient editing, which combines gradients from both the object and background images to preserve texture and accommodate partially transparency object images. Experimental results demonstrate superior performance of Poisson-based methods compared to basic copy-and-paste techniques, with significant improvements in color transitions, lighting consistency, and texture retention. The paper highlights challenges, particularly in detecting object regions and handling transparency, and proposes efficient solutions such as threshold-based and connected component algorithms. This work underscores the effectiveness of Poisson image editing for high-quality image blending and serves as a foundation for further applications in gradient-domain image processing.

Introduction

The aim of this project is to seamlessly integrate an object image into a background image using Poisson image editing. In this technique, we convolve an unknown region of interest(ROI) using the Laplacian kernel and match it to the target gradient. In mathematical terms, this is known as the Poisson partial differential equation with Dirichlet boundary conditions. In our project, we provide a discrete solution to this equation by using a sparse matrix. To demonstrate the feasibility of this approach, we first created a dataset of object and background images and a function that straightforwardly copy-and-pastes an object image into a background image, without any blending techniques. Subsequently, we implemented a function to solve the Poisson equation, which acts as a system of linear equations that relates the unknown ROI to the target gradient. We examine the success of our implementation by comparing the blending results of two types of target gradients: the object gradient, which achieves seamless cloning of a solid object, and the mixed gradient, which achieves the blending of a partially transparent object while preserving the texture of the background. By comparing these two approaches, our project provides a solid foundation for further exploration in gradient-domain image processing and its applications in computer vision. In doing so, it demonstrates the versatility and effectiveness of Poisson image editing in achieving high-quality image blending.

Related Work

In Pérez et al (2003), the authors originally introduced Poisson image editing, the technique at the heart of our project. Their foundational work on Poisson image editing proposed a novel framework for seamless image manipulation through gradient-domain techniques. The authors first achieved seamless cloning; by defining the target gradient as the gradient of the object image, they allowed the unknown region to be reconstructed with adjusted color while preserving the object's edges and details. They further extend this approach with mixed gradient editing, in which the target gradient is adaptively chosen as the dominant gradient between the object and background images. This idea enables enhanced boundary preservation and handling of overlapping textures. This framework established versatile and robust applications in image manipulation. In our project, we implement the discrete Poisson solver, which corresponds to equation (7) in Pérez et al (2003).

Building on this foundation, Tralie (2023) provided a comprehensive tutorial that guided our understanding and implementation of the Poisson partial differential equation and its discrete implementation as outlined in equation (7) of Pérez et al (2003). His work also offered practical examples that clarify the nuances of solving the equation in real-world scenarios, particularly through sparse matrix computations.

Experiment Setup

Dataset

This section discusses the data set used to test our Poisson solver algorithm. In total, we gathered four sets of images: 3D background images in nature (Fig.1) and a set of colored cut-out object images (Fig.2) to test seamless cloning, along with flat background images (Fig.3) and a set of partially transparent colored and grayscale object images (Fig.4) to test mixed gradient interpolation.

As the first milestone in this project, we created a dataset of preprocessed images and implemented a basic image cloning technique; the copy-and-paste of an object image onto a background image without applying any blending techniques. To construct the dataset, we sourced images from free, non-copyrighted image repositories at www.pexels.com and www.pixabay.com. We prepared the object and background images by manually cropping the target objects from the selected images using Paint.net - ensuring that for solid cropped object images for seamless cloning, a thin border was left around each object. We standardized all image sizes to 400x600x3 to simplify subsequent processing, and we designed algorithms such as ImagePaste.m to assume this input size. At this stage, the feasibility of our project was demonstrated by simply copy-and-pasting an object image to a background image. Example copy-and-paste images of a man on a raft and a bunny are presented below (Fig. 5, 6).

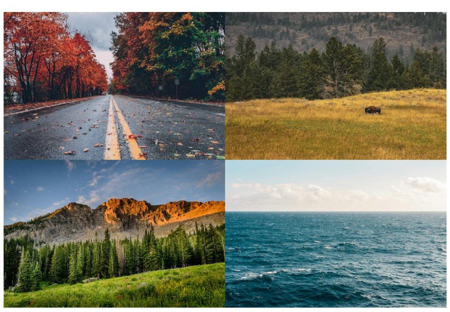

Figure 1: Set of 3D Nature Background images

Figure 2: Set of Solid Cropped Object images

Figure 3: Set of Flat Background Images with Paper Texture

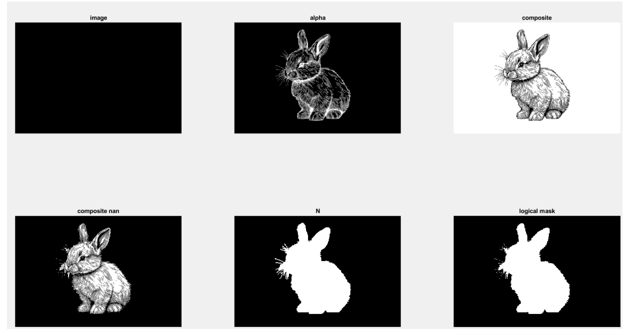

Figure 4: Example of Transparent Object and Corresponding Masks

Algorithm Development

In this section, we explore the Poisson image editing algorithm and how its implementation is organized. First, we implemented a simple copy-and-paste method as a baseline comparison for subsequent blending methods. Then, we created a function that solves the Poisson equation for seamless cloning by setting up a sparse matrix and calculating the target gradient of the object. Finally, we extended this function by calculating the target gradient for mixed gradient editing.

Simple Copy-and-paste

Initially, a straightforward copy-and-paste cloning method was developed on ImagePaste.m to serve as a baseline for subsequent comparisons. This method operates by directly replacing the background pixels with the object image pixels at the target location, effectively removing the underlying background information where the object is placed, and combining the two images without applying any blending or advanced techniques. Example outcomes of the copy-and-paste cloning method are presented below (Fig. 5, 6).

Figure 5: Copy-and-paste of raft image

Figure 6: Copy-and-paste of bunny image

Poisson Image Editing

We then implemented PoissonSolver.m, a function forming the core of this project. The goal of this function is to solve the Poisson equation — the solution is the unknown ROI, i.e. the Poisson-adjusted version of the original object image for seamless blending. Drawing on the methodology described by [Pérez et al 2003] in equation (7), in this function, we set up a sparse matrix A which relates the unknown ROI to the target gradient. We then calculated the target gradient and solved the sparse linear system to obtain the Poisson solution.

In the discrete Poisson equation, given the object image OBJ and background image BG, our goal is to solve for an unknown ROI H. Within H, we define the interior region to be Ω and its boundary region (pixels with less than 4 neighboring object image pixels) $\partial\Omega$. For Poisson editing of colored images, we solve the Poisson equation independently for each color channel. In this project, all results were obtained in the RGB color domain. To solve the discrete Poisson equation, first, let us consider the Dirichlet boundary condition.

1 Dirichlet Boundary Condition

The Poisson method proposed by [Pérez et al 2003] suggests fixing the values in $\partial\Omega$, the boundary pixels of H, as the corresponding pixels in BG on the same boundary. This ensures that our Poisson solution is anchored in the surrounding context of the background image and guarantees continuity. In this project, we followed the same method to determine the pixel values in $\partial\Omega$.

This theory arrives at the following equation:

$$H_{(x,y)} = BG_{(x,y)} \vee (x,y) \in \partial\Omega \tag{1}$$

In essence, if the pixel at (x, y) in H is a boundary pixel, it is set as the pixel value at $BG_(x, y)$. The boundary conditions act as a bridge between the gradient within the ROI and the background, which is the cornerstone of the Poisson method to achieve visually convincing image blending. Now, we solve for the interior pixels in H, i.e. Ω.

2 Solving the Discrete Poisson Equation

To solve for pixel values in Ω, the Poisson equation is set up as follows:

$$A\mathbf{x} = \mathbf{b}. \tag{2}$$

Where

- $\mathbf{x}$: a 1D vector containing all pixel values in H, including $\partial\Omega$ (now known due to boundary conditions) and Ω (unknown, yet to be solved)
- $\mathbf{b}$: the 1D vector containing the target gradient
- A: a sparse matrix encoding the stencil of the Laplacian operator

The structure of A ensures that each row corresponds to a single pixel in H, with the following properties:

- The diagonal element for pixel (x, y) is 4, representing the weight of the central pixel in the Laplacian operator.
- Off-diagonal elements correspond to neighbors $(x \pm 1, y)$ and $(x, y \pm 1)$, each with a weight of -1.
- For pixels along the boundary, $\partial\Omega$, matrix A incorporates the Dirichlet boundary condition by setting the corresponding row to enforce $H_{(x,y)} = BG_{(x,y)}$.

The resulting matrix A is large and sparse, with most entries being zero due to the localized nature of the Laplacian operator. To overcome the challenges of the matrix size being too large, we utilized the MATLAB *spalloc* function to allocate space efficiently for a sparse matrix.

After creating the matrix A and calculating the target gradient (more in the subsequent section), in this project, we solved the Poisson equation using the $MATLAB$ backslash operator to solve for x. However, we would like to remark that x could also be approximated more efficiently using the Jacobi method. For higher-resolution images, utilizing the Jacobi method may be more suitable for more cost-efficient and time-efficient results.

3 The Target Gradient

Here we discuss how target gradients were computed for seamless cloning and mixed gradient editing.

Seamless Cloning Our aim for seamless cloning is for the gradient of Ω to match the target gradient, which is the gradient of OBJ. We calculated the gradient at (x, y) in OBJ by convolving it with a Laplacian kernel:

$$\Delta OBJ(x, y) \approx 4OBJ(x, y) - OBJ(x-1, y) - OBJ(x+1, y) - OBJ(x, y-1) - OBJ(x, y+1) \quad (3)$$

where $OBJ(x, y)$ is the intensity at pixel (x, y), and $OBJ(x \pm 1, y)$ and $OBJ(x, y \pm 1)$ correspond to the neighboring pixel intensities.
Example results of seamless cloning are presented below (Fig. 7, 8).

Figure 7: Seamless Cloning of raft image

Figure 8: Seamless Cloning of bunny image

Mixed gradient editing In mixed gradient editing, the target gradient is defined as the dominant gradient between the object image OBJ and the background image BG. Therefore, in addition to OBJ, we also convolve BG with the Laplacian kernel to obtain $\Delta BG(x, y)$. Then, we determine

$$\mathbf{b} = \begin{cases} \nabla BG & \text{if } |\nabla BG| > |\nabla OBJ|, \\ \nabla OBJ & \text{otherwise.} \end{cases} \quad (4)$$

Where:

- ∇BG: Gradient of the background image.
- ∇OBJ: Gradient of the object image.
- $|\nabla BG|$ and $|\nabla OBJ|$: Magnitudes of the gradients.

The example results of Poisson editing using mixed gradients are presented below (Fig. 9, 10).

Figure 9: Mixed Gradient editing of raft image

Figure 10: Mixed Gradient editing of bunny image

Evaluation

We qualitatively evaluated each of the three algorithms: copy-and-paste, seamless cloning, and mixed gradient editing based on the following criteria.

- Smoothness of the Color Transition: assess how seamlessly the colors blend at the boundaries between the object and the background.
- Consistency of Light Transition: determine whether the lighting at the boundaries transitions naturally between the object and the background.
- Retention of textures: examine whether the textures in both the object and the background are preserved without noticeable distortion.
- Presence of Unwanted Artifacts: identify and minimize unexpected or undesirable elements introduced during the cloning process.

For each image editing technique and its resulting images, we measured each criterion on a 0-3 scale and determined overall performance by the sum of these scores. Here, we provide a written analysis of the example images included in this paper, followed by a numerical score for each criterion of color, light, texture, and artifacts respectively.

Finally, we evaluate each algorithm through a numerical score average for all the image editing results that were present in our dataset.

1 Copy-and-Paste

Consider Fig. 5 and Fig. 6, which are images created using a simple copy-and-paste cloning method of unedited object images using the ImagePaste.m function.

Copy-and-paste of raft image (Fig. 5): Score 2 Notice the clear boundary between the pasted object image and background image; neither the transition of color nor light is consistent. Moreover, textures are retained, but with clear distortion, and a thin white boundary around the edge of the object image is noticeable due to cropping. These result in a score of 0, 0, 1, 1, resulting in an overall score of 2.

Copy-and-paste of bunny image (Fig. 6): Score 3 The color and light transition between the object and background is abrupt, but since the object image itself is cut out in a way that leaves no border around the object, aesthetically, the object appears naturally on the background. Notice however that the object does not retain the texture of the paper in the background. There is an artifact near the whiskers of the bunny, where the fine lines were unable to be cropped out accurately and left a small white area instead. With a score of 1, 1, 0, 1, this image has an overall score of 3.

2 Seamless Cloning

Now, let us transition to Fig. 7 and Fig 8, which are images created with seamless cloning techniques.

Seamless cloning of raft image (Fig. 7): Score 10 This image has very few noticeable artifacts and an aesthetically convincing result. The color and light transition is seamless, although the texture of the water in the Poisson-adjusted object image is not completely continuous from the surrounding context. There are no unwanted artifacts, except that the man in the raft has a much darker color compared to the man in the original object image. This image has a score of 3, 3, 2, 2 and an overall score of 10.

Seamless cloning of bunny image (Fig. 8): Score 5 The light and color transition in this image is more natural than that of copy-and-paste, but still not completely smooth. The texture of the paper is still not retained, and the color is distorted near the bunny's lower back. The whiskers to the left of the bunny's face has disappeared, which we hypothesize is because the lines were too fine and thus the gradient too small. With 2, 2, 0, 1, this image has a score of 5.

3 Mixed Gradient Editing

Finally, we examine Fig 9 and 10, which are edited using mixed gradients.

Mixed gradient editing of raft image (Fig. 9): Score 10 The light and color are transitioning perfectly from the object to the background. The texture of the wave, especially the slight rise in the water in front of the raft, is retained in this image. However, the raft seems to have a slightly distorted color, which makes the image editing appear unnatural. With 3, 3, 3, 1, the overall score is 10.

Mixed gradient editing of bunny image (Fig. 10): Score 9 The light and color transition has remained similar to the seamless cloning result, but the texture of the paper from the background is retained. Notice the lower back and leg of the bunny; the crease of the paper can be observed on the bunny's body. No artifacts, except the whiskers disappearing, are seen. With 2, 2, 3, 2, the image has a score of 9.

Overall Evaluation

In a similar fashion, we qualitatively evaluated the results of all image editing techniques for all images in the prepared datasets. We calculated the average score for each criterion in each technique, resulting in the table below.

Table 1: Evaluation of Image Editing Techniques

Criteria	Copy-and-Paste	Seamless Cloning	Mixed Gradient Editing
Smoothness of Color Transition	0.50	2.50	2.50
Consistency of Light Transition	0.50	2.50	2.50
Retention of Textures	0.50	1.00	3.00
Presence of Unwanted Artifacts	1.00	1.50	1.50
Overall Score	2.50	7.50	9.50

Overall, compared to the conventional copy-and-paste method, which serves as a baseline for all subsequent comparison of Poisson image editing techniques, both seamless cloning and mixed gradient exhibited significantly superior performance across the majority of evaluation criteria. In particular, with respect to color and lighting consistency, both techniques effectively generated smooth transitions without noticeable artifacts. However, texture synchronization, a critical aspect for perceptual coherence, was achieved flawlessly only by mixed gradient seamless blending. This indicates that while both methods excel in mitigating color and lighting discontinuities, mixed gradient method offers a distinct advantage in maintaining texture consistency across blended regions.

Challenges

Object Outline Detection

Initially, we came up with a simple threshold-based algorithm to distinguish the object with a white background in the images. However, this method resulted in inaccuracies, as parts of the white object regions were also detected as background regardless of threshold level. This problem resulted in creating undesirable "pierced"objects, as shown in Figure 11.

To mitigate this problem, we introduced an additional constraint during the creation of the non-transparent object set: *each background pixel must be connected horizontally to either the left or right edge of the image*. This simple yet effective assumption allowed for fast and reasonable object determination and cropping. Although this method is not perfect, it provides a computationally efficient solution for defining object regions.

Figure 11: Threshold-based method incorrectly detecting pixels inside object

Figure 12: Threshold-based method with constraint. Notice that the area between a leg and tail is not detected as background because there is no connection to the edge horizontally.

Challenges with Transparent Objects

Transparent objects present a more complex challenge due to their intricate outlines and ambiguous boundaries. The threshold-based method described above caused significant errors, failing to accurately detect these regions. To address this problem, we made improvement based on a similar concept of connected background regions.

Assumptions and Algorithm

The proposed method is based on the following assumptions:

- The pixel at location $(1, 1)$ belongs to the background.
- All background pixels are connected, forming a single connected component.
- A breadth-first search (BFS) starting from pixel $(1, 1)$ can capture all background pixels by propagating to their neighbors.
- White pixel is defined as $all(R, G, B) > 0.99$. We set this threshold to ensure that the detected pixel is pure white created by the the system.

Given the above assumptions, the algorithm proceeds as follows:

1. Add pixel $(1, 1)$ to the queue.
2. Repeat the process below until the queue is empty:
3. Take a pixel from the queue. If the pixel is white, propagate to its neighboring pixels (8-connected neighbors).
4. Add $(x + 2, y), (x - 2, y), (x, y + 2), (x, y - 2)$ pixels to queue.

This algorithm simultaneously marks nine pixels as background in a single operation. While it does not achieve perfectly accurate outline detection, the design prioritizes computational speed over precision. Although this optimization does not alter the asymptotic complexity of the algorithm, it proves to be highly effective in practical applications, particularly under the assumption that the input image size remains relatively small (e.g., 400×600 pixels or less).

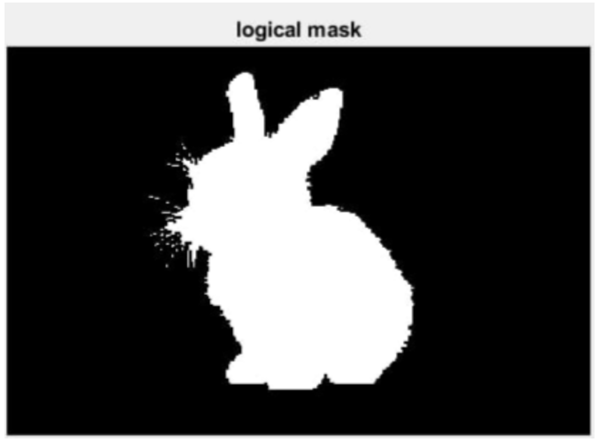

Figure 13: Improved method for detecting transparent object regions.

Conclusion

In this project, we explored Poisson image editing as a gradient-domain technique for seamless object blending. We implemented three main methods: copy-and-paste, seamless cloning, and mixed gradient editing. Our results demonstrate the limitations of naive copy-and-paste methods, particularly in achieving smooth color and lighting transitions. In contrast, seamless cloning significantly improved blending by aligning object gradients with the background context, while mixed gradient editing excelled in maintaining texture continuity.

Despite the effectiveness of Poisson image editing, we faced some challenges in the process. Transparent objects and fine structures, such as thin lines, introduced inaccuracies during object detection and boundary handling. These challenges were mitigated using threshold-based and connected component algorithms, though further improvements are needed for edge cases involving transparency and complex textures.

Future work can address these limitations by perhaps incorporating machine learning-based object segmentation methods for more accurate region detection. Moreover, for more complex and detailed images, we hypothesize that adaptively choosing target gradients depending on the region's textures could yield robust results for any type of image. Additionally, optimizing the Poisson solver for higher-resolution images, such as using the Jacobi method, and exploring its applications in video editing or real-time image manipulation could extend the practical utility of this technique.

Overall, our implementation highlights the versatility of Poisson image editing for high-quality image blending and demonstrates its potential as a foundation for advanced image manipulation techniques in computer vision.

Acknowledgments

We would like to thank Professor Jerod Weinman and our classmates in *CSC262: Computer Vision* at Grinnell College for guiding our project and providing feedback. https://weinman.cs.grinnell.edu/courses/CSC262/2024F/.

References

[1] Pérez, P., Gangnet, M., & Blake, A. (2003). *Poisson image editing*. ACM SIGGRAPH 2003 Papers, 313-318. https://www.cs.jhu.edu/~misha/Fall07/Papers/Perez03.pdf

[2] Tralie, C. J. (2023). *Poisson Image Editing*. https://www.ctralie.com/Teaching/PoissonImageEditing/

All images are non-copyright, taken from www.pexels.com and www.pixabay.com