Frankie Schulte

6.8371 Problem Set 9 Write-Up

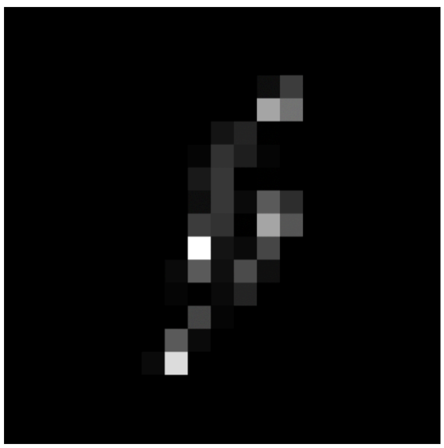
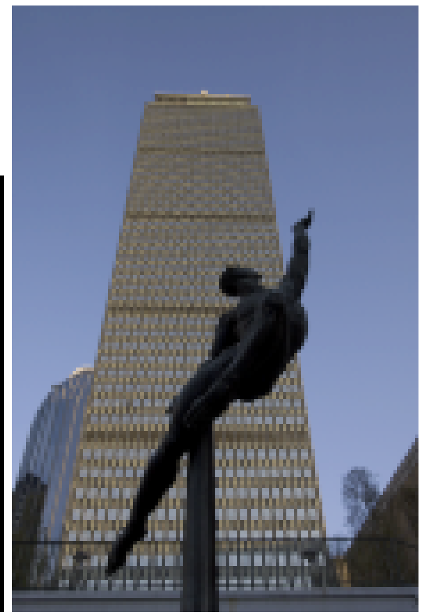
For my problem set 9, I decided to implement a previous years' problem set, specifically the Deconvolution and Poisson Editing assignment (https://stellar.mit.edu/S/course/6/fa13/6.815/homework/assignment10/). One of the reasons I ended up choosing this problem set was because I remembered the lecture slides from the first day, and seeing one of the slides showing off this problem set's results really intimidated me as an assignment that I would eventually have to do. But now that I've finished all of the previous problem sets in the class and have a full semester's worth of computational photography experience, I wanted to see if I could tackle this algorithm and properly implement it. I also have a lot of experience in Lightroom and Photoshop, and have many times been challenged with seamlessly blending one object into another, so implementing an algorithm that did exactly that seemed like it would be an exciting challenge. However to be completely honest, I had originally hoped to do the dual photography problem set, but since I picked up the projector after Thanksgiving break I wasn't confident I'd have enough time to get the dual photography working (I hope to come back to it as a future project however!)

Deconvolution is often used to eliminate image defects and increase image resolution, a method that is implemented through following a point spread function that approximates how was blurred, and can therefore be deblurred (1). This technique is often used to improve the quality of microscopes, as the resolution can be improved without having to improve a microscope's engineering (2). One particular deconvolution method is known as the Poisson algorithm, which aims to recover a clean image from a blurry and noisy image when the noise follows a Poisson distribution, commonly seen in low-light images. This technique allows for adjustments to be made in proportion to the intensity of the noise, creating a much more accurate final result than models that assume Gaussian noise (3). For the purposes of this problem set however, the Poisson algorithm is used for the seamless integration of one image over another, using the algorithm to replicate the gradient difference between two images.

The first algorithm implemented in the problem set is the deconvolution algorithm. Deconvolution is an algorithm that aims to reverse the effects of blur in an image. The blurring is often modeled using a Point Spread Function (or PSF) as seen in Figure 1. This PSF is used to enact convolution on an initially sharp image to blur it in a particular pattern. Given this blurred image y, a PSF k, and assuming no noise, deconvolution allows for one to recover the sharpness of the image x by solving the equation:

$$y = k * x$$

With * denoting convolution. Directly applying this equation will amplify many small errors in the image or PSF; for this reason, gradient descent and conjugate gradient methods are used to approximate a more accurate solution. For gradient descent, we minimize the residual error between the convolution and the blurry image. The gradient from this error function is what determines the direction the solution is updated. For the conjugate gradient, it improves upon gradient descent by ensuring each new direction is conjugate to the previous directions, resulting in a more efficient convergence.

Figures 1 and 2. The PSF is shown in Figure 1 on the left, which when used in convolution on the input image of Figure 2. The results of the convolution and deconvolution are shown later in Figures 3 and 4.

The algorithm runs in multiple iterations to better approximate the final solution. For each step, it first computes the residual as the difference between the target image and the PSF convolution estimate. It then applies the conjugate transpose of the PSF to refine the residual's direction. It then computes a step size to minimize the error along the direction of the residual, and to take smaller steps as it gets closer to the solution. Finally, it updates the estimate and repeats the cycle. All of these steps combine to get a much more accurate approximation of the deconvolved image, with the before and after shown in Figures 2 and 3.

Figures 3 and 4. Figure 3 on the left is the convoluted image from the PSF. Through deconvolution with the PSF as reference, the blurry image can be de-blurred into Figure 4 shown on the right.

The second half of the problem set focuses on Poisson image editing, which is an algorithm that aims to seamlessly blend content from one image into another through the use of masks and gradients. Unlike naive compositing, which simply pastes contents of one image over another, Poisson blending creates a smooth transition by maintaining the gradient continuity of the combined image. The algorithm works off of a foreground image *f*, a background image *b*, and a binary mask *m* that aim to have the gradients within the masked region (*b* within the bounds of *m*) to match that of the foreground image *f*. The values in the new image *x* should look something like:

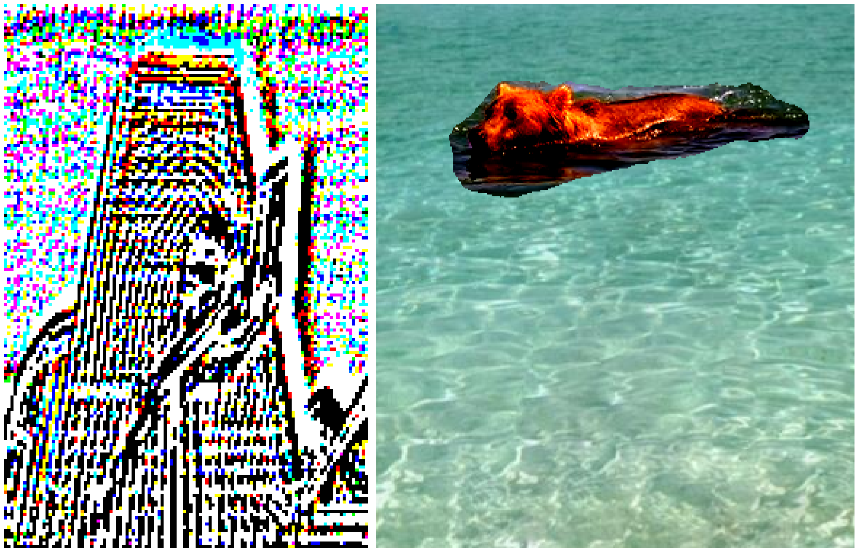
$$\nabla^2 x = \nabla \cdot g \text{ where } g = \nabla f$$

Figure 5. This is a naive approach to compositing an image of a bear over another body of water; the water colors don't blend whatsoever and the bear very clearly has been pasted from another image. Unbearable!

Implementing this algorithm follows a similar iterative process to the deconvolution algorithm, as it also uses conjugate gradients to improve the final output. It first begins with a naive guess as to how the foreground and background should combine, and from there iteratively updates the image by computing the residual error between the desired gradient and the Laplacian of the current image's state. The goal is to converge to a solution where the gradient constraints are satisfied while only updating elements within the region of the mask, which ultimately creates the result shown in Figure 7.

Figures 6, 7, and 8. The background image is shown in Figure 5 on the left, with the foreground bear to be blended shown in Figure 6 in the middle. Using the Poisson algorithm and conjugate gradients, the bear can be seamlessly blended into the water in a much more convincing manner compared to Figure 5.

There were many challenges with implementing the problem set, most notably all of the math involved. Much of the challenge with implementing both deconvolution and Poisson is the math required for both, as there's a lot of complex image manipulation required to get the exact desired output. Testing was very challenging, as much of the time if even a single sign or number was wrong the whole image output would look incomprehensible. Additionally, there were some times where I implemented a solution that worked, but didn't properly implement the algorithm or was "cheating" to get the solution (such as when I implemented the Poisson function using for loops and kernel manipulation, which while it worked, it wasn't particularly a "Poisson algorithm"). Testing was certainly made easier thanks to the problem set having intermediate functions outlined. Much of these intermediate functions could be tested before attempting the final algorithms for each.



Figures 9 and 10. Some interesting results I saw while testing, often these were one sign change or one function reorder away from the correct implementation.

With my final implementation I was able to create my own test cases and use the algorithm to blend my own images, which I did by blending my friend Elena’s face over a pineapple. Something like that would ordinarily be extremely tedious for me to attempt in Photoshop, but only took a little bit of image formatting using the algorithm, which I think is very fun. The pineapple test case is shown below for your viewing pleasure!

Figures 11, 12, and 13. Took this photo of my friend Elena, made a mask over the pineapple, and blended her face into the pineapple. The rough pineapple surface doesn't match well with the smooth skin on a human face, but the Poisson algorithm is able to color match her face much more accurately than anything I could make in Photoshop.

Ethics Write-Up

When tackling the creation of software that interfaces with touchy subjects, in this case the human body and the topic of beauty, it's very easy to overstep boundaries into creating research that sends the wrong message. The idea of creating a program that's designed to "make someone more attractive" is so general it raises a lot of questions. By what standard of "attractive" are we defining? Will there be a point where the program is functioning well enough, or will there be no resolution of how to make someone "attractive?" And that's not even delving into the sensitivities involved in body modification, much less non consensual body modifications. If this technology were to be made available broadly (which it is now), it comes with concerns of spreading misinformation, and could be easily abused in the wrong hands. Social media has made it easy to put out a false image of oneself, and having access to technology that makes you more attractive in the digital world is just another tool to fuel deceit and manipulation on social media. There are so many questions that need to be nailed down before a program that "makes people look more attractive" can even be considered.

To avoid facing some of these issues, I think there would certainly be some guidelines I'd put down before putting together the program. Firstly, defining specifically what the program modifies as opposed to "program that makes people attractive" will allow for better objective use of the program. Some modifications such as smoothing out rough or flaky skin, or whitening teeth could be safe picks, as these are modifications that don't cause much harm to either the one being modified nor whoever is observing the images. Keeping the scope to these light modifications would be a good way to avoid problems that may arise from say, a program that thins one's waist, or a program that restructures one's facial structure, as these programs are disillusioned to one's real self, and would ultimately support the concept of beauty standards mandating a definitive "attractive body." Beyond just limiting what the program can do, limiting who can access it would also be an important consideration. Broadly publicizing a program that can conduct harmful body modifications could easily be put to use negatively in the wrong hands, whether by large corporations aiming to publicize beauty products, or individuals lurking hoping to torment people who may be insecure. There's a lot to consider beyond just these aspects, but I see those two guidelines as the most important to avoid creating an unethical product.

Works Cited

1. https://pmc.ncbi.nlm.nih.gov/articles/PMC8707587/
2. https://www.nature.com/articles/srep02523
3. https://www.sciencedirect.com/science/article/pii/S0307904X17300148