Alexia Hartogensis
Computational Photography
Fredo Durand
December 12, 2022

Deconvolution and Poisson Image Editing

Introduction:

For my final project I decided to implement the previous problem set assignment Deconvolution and Poisson Image Editing. I choose this previous problem set because I was interested in methods of using the idea of the Poisson image editing equation to create compositions that blended a source image onto a background. This is very similar to the tool of the healing brush or patch tool on photoshop. I am not very familiar with using photoshop, but I have used this tool to perform some basic edits on photos.

Background:

I used the paper *An Introduction to the Conjugate Gradient Method without the Agonizing Pain* by Jonathan Schewchuck to get a method for Conjugate Gradient Method. This method increases the efficiency of the convergence gradient descent. We need convergence gradient descent to be able to create a gradient for images which will be later used for the Poisson image editing. The Conjugate Gradient method is a method that solves large linear systems in a process similar to Gram-Schmidt orthogonalization. We want to solve Ax = b where vector b and A are known and vector x is not. The method is iterative which helps us solve with sparse matrices in a more memory and time efficient way than factoring and back solving. In simplest terms, we are solving for the search directions using the conjugation of the residuals. We can do this because each new residual is a linear combination of the previous residuals.

The next article I used for my project was Perez et al.'s article "Poisson image editing". This article gives an in-depth look into how we can implement compositing with Poisson Image Editing. The outcome is a seamless transition from the source image to the background image. Perez uses the Poisson equations to interpolate a gradient between a source image and a target image for seamless edits of certain regions. First the article talks about interpolating an opaque and transparent source image onto a region of a destination image. Second the article talks about how to change the appearance of the image to have a seamless transition between the source and target. The changes are for either slight distortion for photos that are blurry or complete replacement such as copy pasting a source image onto a background.

Outside of the articles for the problem set, I found two articles on different ways to implement Poisson Image editing for other tools. In the article “An Implementation Method of Poisson Image Editing on FPGA” by Maeda and Maruyama explain another use for Poisson image editing. They explain how to use Poisson Image Editing in movies. First the article explains briefly the editing method. Then it goes on to explain how the image process can be implemented onto a special chip (FPGA) that can be inserted into a computer to have faster image processing. The image processing is too computationally complex to operate in real time on images for a movie without the use of a custom FPGA chip. Next, I read an article “Drag-and-Drop Pasting” by Jiaya Jia about a user-friendly system that performs the seamless image composition from the Poisson Image Editing methods. They compute an optimized boundary condition to make the process more efficient and easier to use. In other words, they implement a structure of the system by outlining the region of interest and drag-and-drop it onto a target image. Both of these articles helped me learn more about the powerful techniques and uses of Poisson Image Editing.

Algorithm:

The algorithm I used replicated a gradient of a source image into a masked region of the target image. I implemented two different versions of Poisson: using gradient descent and using the conjugate gradient. First, I will explain the gradient descent algorithm. I used a Laplacian Kernel which is a matrix that manipulates each pixel in the image. Thus, I convolve an image by a Laplacian kernel. The Laplacian kernel is the second derivative of the composition of two gradient operations. The algorithm will start by creating a constant image and then through a serious of manipulations convert the constant image into the image that is desired. We use the Laplacian image and a mask of the source image to create steps to slowly change the target image. In order to not have the process continue infinitely, we bound the iterations by an iterator.

The second implementation of the Poisson algorithm we use is with the conjugate gradient. In this version we create the steps with a Conjugated Direction Step and the Laplacian kernel. We pick a direction for the step and blend the image gradually until we reach the number of iterations we want. This increases efficiency of the gradient between the two images significantly. In the original implementation of the algorithm, 2,000 iterations were needed in order to create the same blend of the two images that 200 iterations of the conjugate gradient implantation needed.

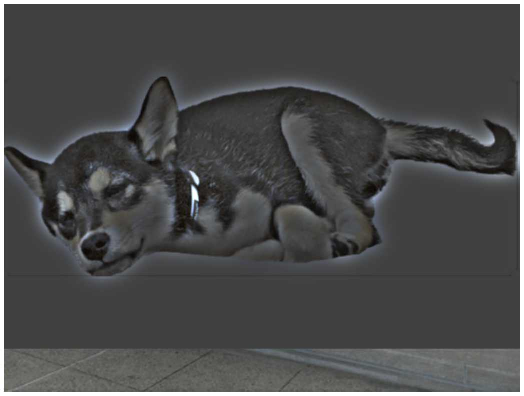
Implementation:

On top of implementing the Poisson with both algorithms I described above, I also implemented different ways of deblurring an image. One way was with Gradient descent. Similar to the Poisson algorithm, this function was completed by creating an optimal alpha step from the constant image and morphing into the target image. Next, I implemented another way of deblurring the image with conjugate gradient descent. For this implementation, I used *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain* of figuring out how to create an optimal alpha step and a direction for the constant image to convert into the target image. The hardest part of the implementation for me was figuring out how the conjugated steps changed the initial image. However, once I figured out the algorithm, I was able to move onto the other implementations.

Poisson – 2,000 iteration

Poisson with Conjugate Gradient

My own composition



Ethical Issues

If my supervisor gave me the task of creating a system to manipulate images of people to make them look more attractive, I would focus on blending the person's skin. For example, I would create a system that removes blemishes and creates an even skin tone. I would choose to focus the design on the skin because I feel that this is the most ethical focus out of all the beauty enhancements. Of course, there are many unethical issues of changing your skin appearance, such as excluding people with rare facial conditions. However, I understand the appeal of wanting the makeup look without effort. Therefore, I would like an app that gives the person fake makeup rather than a different face. It is irresponsible for developers to create apps that change a person's appearance. The unethical part of creating a system that makes people look more attractive is when the system creates an unhumanly attainable image. If a person can not reach the standards represented, the app's effects are unethically belittling the users. In addition, it becomes unethical to create an app that promotes beauty standards that would cause people to change themselves in unhealthy ways, such as promoting unachievable weight goals. The assignment should be changed to encourage the user's natural beauty.

Instead, the assignment should be how to enhance a person's natural beauty. A more ethical way of having the person become more attractive is creating something similar to real-life makeup that takes what a person already has and enhances the features. Therefore, I would rescope the project from "manipulate images of people to make them look more attractive" to "how enhance a person's features." As I mentioned, there becomes a problem when a person's image is changed to either something unattainable or belittles their features. I thought about applying real-world makeup when I mentioned that I would create a system that evened the person's tone. In other words, I would want a system that would enhance a person's eyes by adding framing around them, such as virtual eyeliner. Or even improve a person's facial appearance by evening the color. And there are still some ethical repercussions of changing a person's face this way, but the more ethical way of solving this issue would be to enhance instead of change or morph.

There are many issues of making a person more "attractive" since there are so many different beauty standards, and some are more unattainable than others. However, as software engineers, we should focus on portraying standards that people can achieve daily rather than something they could never get.

Works Cited

Shewchuk, J. R. *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*. Carnegie-Mellon University. Department of Computer Science, 1994.

Pérez, Patrick, et al. "Poisson Image Editing." *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*, 2003, https://doi.org/10.1145/1201775.882269.

Maeda, Ryouhei, and Tsutomu Maruyama. "An Implementation Method of Poisson Image Editing on FPGA." *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, 2017, https://doi.org/10.23919/fpl.2017.8056813.

Jia, Jiaya, et al. "Drag-and-Drop Pasting." *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, 2006, https://doi.org/10.1145/1179352.1141934.