

Project 3 Writeup

Process and Algorithm

The process for Poisson blending is quite simple, but implementing it is tedious. There were a number of optimizations I made to try to make my code run faster, and eventually it was able to do the rainbow image in a few seconds.

First, I created a map that mapped each pixel in the mask to a variable number. This variable number describes the number of equation that will be used to populate the A matrix.

Next, I populated the A matrix according to the Laplacian distribution of coefficient 4 for the current pixel, and -1 for all surrounding neighbors. This is so that overall the photo does not get brighter/dimmer after the Laplacian is applied.

If the pixel in question was spanned by the mask, then I would take the gradient ($\text{source}(y-1, x, 1) - \text{source}(y, x, 1)$), or whatever the corresponding neighbor's gradient was, and populate my b (right side of the equation's coefficients) matrix with that value. If it is not in the mask, that means we want to keep those values, so I simply made it equal to the target image.

Finally, using SVG I solved the system of equations, leading me to new pixel values for each of the color channels for each pixel in the mask. I overlaid these pixels over the target image ones, and this is my final output.

The failure case happened because the region that we are trying to mask out is on the edge of the photo. This means that the gradient isn't very well defined, as you can't take the change from a value that exists to something that does not exist. If you try to take this difference, you'll get an array out of bounds exception. It also looks extra trash because it's trying to blend the green in the trees with the rainbow.

Input/Output photos

Input photos





Naive Blending



Gradient Domain blending



Input photo





Naive Blending



Gradient Domain blending



Input photo



Naive Blending



Gradient Domain blending



Input photo





Naive Blending



Gradient Domain blending



Input photo





Naive Blending



Gradient Domain blending



Input photo





Naive Blending



Gradient Domain blending



Input photo





Naive Blending



Gradient Domain blending

