

Portrait Obama-fication

K. Paprocki

Computer Science Undergrad
UW-Madison
kpaprocki@wisc.edu

S. Bric

Computer Engineer Undergrad
UW-Madison
sbric@wisc.edu

Abstract

The goal of our project was to create an algorithm that turned regular portraits into the popular Obama 2008 campaign poster style created by Shepard Fairey (see fig.1 below). There were several problems we faced when designing the algorithm: discerning what was foreground and what was background, preserving important features while ignoring texture, and mapping the colors in the photo to a four color palette.

To differentiate the background, we relied on a simple mask. We use a bi-lateral filter on our image to remove texture and high frequency color and intensity while preserving edges. Finally we map the four color palette onto the image by setting different ranges of intensity. These ranges currently need to be tuned by hand to create an acceptable image.

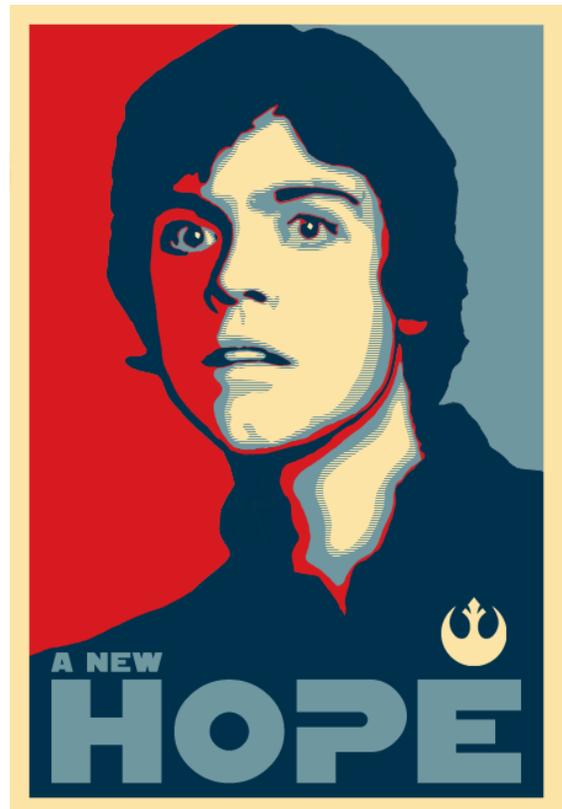




Figure 1



Figure 2

Our algorithm uses the matlab bilateral filter function written by Douglas R. Lanman. This algorithm was implemented using the method outlined in C. Tomasi and R. Manduchi's paper *Bilateral Filtering for Gray and Color Images*. The results of our algorithm come very close to producing the original image, but still require some amount of user input.

Introduction

Obama's iconic 2008 campaign poster was created by artist Shepard Fairey in under one day as a mixed media stencil portrait. Over the past several years many people have made their own version of this portrait with motivations ranging from comedy to malicious mockery. Our goal was to develop an algorithm that would automatically convert ordinary portraits into the portrait style created by Fairey. With this algorithm even the inartistic could create their own Obama style campaign poster with little effort.

Motivation

The motivation for this algorithm came from the many parody posters that have circulated the internet in recent years. Though some of them looked convincing, a majority of them did not. We created this algorithm to assist our fellow internet dwellers in keeping this funny meme classy.

The Problem

Our initial idea was to map the four tones of the original poster to other portraits based on intensity and possibly color. There would be predefined ranges which would map to each individual tone. Comparing figure 1 and figure 2, this is most likely what Fairey did. You can tell that Obamas hair, suit, and the shadows were all drawn in deep navy blue while the highlights on his face and his white shirt were drawn as beige. This naive approach produced similar results to the original photo, but suffered from many different problems.



Figure 3

One of the first problems we had to solve was segmenting the person from the background. In the original poster the image background is partitioned into red and blue segments down the middle of the portrait. To get a clean and distinct partition we knew that the background would have to be clearly defined. We considered manually changing the color of the background so that we could treat it separately, but that still occasionally left foreground pixels being incorrectly treated as background pixels.

Another problem we encountered was high frequency information. Using the simple threshold algorithm, any abrupt changes in intensity that bordered a threshold would get mapped simply as noise. If you examine figure 3 you'll notice that the colors are generally in the correct place, but there are no smooth edges.

Theory

The challenge to creating a good algorithm is removing this high frequency noise while preserving hard edges and relative differences in intensity. A Gaussian filter would eliminate high frequency textures, but would also degrade the hard edges of the image. However, a bilateral filter fits

all of our requirements: it's designed to preserve edges while blurring areas of similar color and intensity.

It produces this effect by filtering over the domain and range of the image simultaneously. The range of the image is averaged depending on the photometric similarity between two pixels, meaning how similar the two colors are. The domain of the image is averaged depending on the geometric closeness between two pixels, meaning how far the distance between two pixels are. Since the two filters operate independently, pixels are averaged only based on other pixels that are both close in distance and similar in color. This allows edges to remain out of the average, preserving their crispness, while blending interior colors to help smooth out details.

Method

The algorithm we implemented combines the effects of the bilateral filter with the thresholding technique we describe earlier to get the desired output. To partition the back and foreground we require a separate mask image with the background in black and the foreground in white. This guarantees a good partition, but unfortunately requires the user to create their own mask.

The steps to the algorithm are:

1. Load the image
2. Load the mask
3. Apply D. Lanmans bilateral filter to the image
4. Convert the image to greyscale
5. Map the colors based on intensity, using user defined threshold values
6. Color the background based upon the mask
7. Output the result



We found that the threshold values used to map the colors are very sensitive to what image is being input. Because of this we left them as user defined values. The effect of changing the threshold is almost analogous to exposure. If thresholds are too high lots of darker intensities are generalized to one color, as if it was underexposed. If thresholds are too low then a larger range of lighter intensities are mapped to one color, as if the image was overexposed.

Experimental Results

To test our algorithm we decided to run 5 separate images with 5 different sets of threshold values (see appendix A). Each threshold is no smaller than 10% of the total intensity, and no larger than 25%. They are ordered so that the first column has the largest range in the lightest color, and the last column has the largest range in the darkest color.



The results from these tests look surprisingly good. All the picture portraits of people turned out about as good as we could expect. Obama looks best somewhere between the second and third “exposure” level. Mao looks the best at the second. Finally the beautiful lady at the bottom looks best on the third exposure.

Some of the pictures didn’t turn out as well. The Koala portrait doesn’t really look great at any exposure because of the high contrast and frequency of the image. Unfortunately it doesn’t seem like we designed our algorithm for koala portraits. The painted portrait of Vincent van Gogh also didn’t turn

out well for the same reason – high frequency and high contrast. As a brief aside, the photos with the least exposure are actually slightly darker because of how we create the half tones, they are a percentage of the last thresholds total frequency, and therefore cover more area when that threshold is larger.

Of course the most important test is how our algorithm compares to the original poster itself, when based off of the same image. At a distance the portraits aren't too different. Most of the shadows and highlights correspond, as well as the half-tones. Unfortunately our algorithm fails when it comes to fine details that Fairey decided to keep in his stencil, like the whites of Obamas eyes, and the exaggerated detail in his ear. It is of course these differences that make Faireys work a piece of art.



Concluding Remarks

This algorithm was in large part a success. Using the bilateral filter made the job of colorizing the photo's simple and efficient. Unfortunately it still requires a fair bit of manual tuning before you get a picture perfect output. Automatic background segmentation and threshold adjustment are both area's that could be improved on and maybe even automated in the future.

References

- Matlab Bi-Lateral Filter, Douglas Lanman
 - <http://www.mathworks.com/matlabcentral/fileexchange/12191>
- C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Proc. ICCV, 1998,
(2) A. Buades, B. Coll, and J. Morel, "A non local algorithm for image denoising," Proc. CVPR, 2005.

