Image Registration of Component Images for Creation of Color Composites

Edward Venator

Department of Electrical Engineering and Computer Science,

Case Western Reserve University, Cleveland, OH, Email: Edward.venator@cwru.edu

ABSTRACT

This paper presents a method of compositing red, green, and blue digital images, as from separate photographic plates, into a single color image. The advantages of this method are that it is mathematically simple and relatively simple computationally. However, this method is only appropriate when the source images are not rotated or skewed relative to one another, e.g. for images taken with a tripod of a stationary subject.

KEYWORDS

Color, Composite, Image, Registration, Translation, Correlation, Photograph

INTRODUCTION

In the first decade of the twentieth century, Sergey Prokudin-Gorsky toured the Russian Empire on a commission from the Czar to take color pictures of the Russian landscape, culture, and architecture. Since this was well before the advent of color film, Gorsky took three separate exposures of each scene—one each through a red, green, and blue color filter. The resulting photographic plates represent the red, green, and blue color planes of a color photographic image. [1]

The photographic plates have been digitized by the United States Library of Congress, and are available as "strip" images, with the three color components arrayed vertically in a single image. Each of these strip images is about 3700 by 9600 pixels, meaning that each of the component images is about 3700 by 3200 pixels, or nearly 10 megapixels.

The goal of this project was to digitally separate the strip image into the three color plane images and composite them into a single color image. The source images had some limited scratches and noise, which was worst around the edges of each image. The edges of the image were also uneven. Most importantly, the three images must be registered so that they are aligned properly.

Most recent publications on aligning multiple photographs are tailored for making high dynamic range (HDR) composite images. Although this is a similar task, the challenges of registering different color planes are different from those of registering the component images of an HDR composite. HDR images may have differences in luminance, but they contain the same features. These images have different features depending on which features appear in which color plane(s). Many HDR processing techniques are also designed to compensate for excessive camera movement, such as rotation and skew, that result from handheld photography. Since Gorsky's pictures were long exposures taken with the aid of a tripod, these techniques are entirely unnecessary for this task.

Instead, this method relies on the assumption that the three image plates need only be translated to register them properly. With this assumption, which proved valid, advanced transforms are unnecessary, and are replaced with simple shifting operations. The translational offsets are found by discarding the edges and then simply maximizing correlation between the color images. Once registered, the three images can be composited by using the values from each component image as the respective color component values in the output color image.

Based on these assumptions and ideas, several variations were tried for registering the images, and the results for a selection of Gorsky's images were compared to determine how effective each method was at creating composite images.

CHOOSING APPROPRIATE ALGORITHMS

The criteria for selecting the algorithms for this project were that they correctly register the source images and that they run in a reasonable time (less than ten seconds to create a single composite using a typical contemporary PC).

As discussed above, much contemporary research in this field is for creation of HDR composites. One technique, described by Greg Ward in [1], uses median-threshold binary (MTB) image pyramids to quickly find translational offsets between images. This method has two main advantages: it is very fast and it is insensitive to differences in median value between source images. The speed of this algorithm is desirable in this application. The second feature is highly desirable in HDR composites, but is of no use in generating color composites. Experiments with this method found that it discarded too much information from the component images to make them useful. As shown in figure 1, the results of median-thresholding on the composite images resulted in completely different images. Because each image represents a different color, the dark parts of one image do not always correspond with the dark parts of another image. This method was discarded.

Figure 1: Red (l), Green (m) and Blue (r) components of one of Gorsky's images after MTB.

It follows logically from the failure of MTB that an ideal algorithm would discard object value and keep only features. One popular method for extracting features from an image is corner detection. Experiments with Harris corner detection (CITE HARRIS PAPER?) were unsuccessful. The algorithm was not able to find corresponding features in each image. figure 2 shows the corners found in the red component of one of Gorsky's images.

Figure 2: Corners detected in the red component of one of Gorsky's images. Corners are marked by red stars.

The corner detection algorithm seems to be picking up on noise more than features, finding corners in the sky, the grass, and seemingly random parts of the man's clothing.

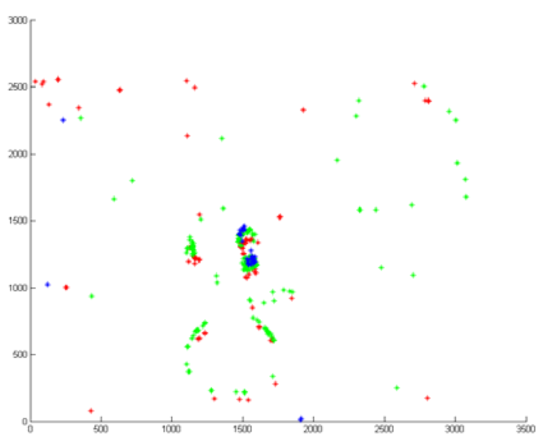

Figure 3: Red, green, and blue corners from the image in Figure 2

However, the true test of corner detection is to compare the results on all three component images. figure 3 shows the corners found in the three component images, color coded by which image they are from. There are many corners from the green image that do not appear in the red or blue image at all, and the blue image has a completely different set of corners. These results are even after attempts at tuning the quality and sensitivity factors of the corner detection algorithm.

Since corner detection could not be used to simplify the problem, it would be necessary to compare the images in their entirety. One method of comparing images in their entirety is correlation. When an image is correlated with a translated version of itself, the result of the correlation has a maximum at the pixel coordinates corresponding to the offset of the translation. For perfectly matching translated images, this pixel is the only non-zero image in the correlation result. Since the component images are not exactly the same, the correlation images (Figure 4) had more non-zero values, but the maximum pixel, shown as the brightest white in figure 4 was at the coordinates of the best offset to align the images. (Figures 4 and 5 have had their grey values rescaled to maximize visibility.)

Figure 4: Correlation between red and blue components of the image in figure 2.

In an effort to further improve upon the correlation method, several edge detection methods were used prior to correlating the images. The most effective was Sobel edge detection. The difference can clearly be seen in figure 5, which is the same correlation as figure 4, but for the edge-detected images. However, the final color composite showed no advantage to the edge-detected images. As such, the final algorithm skips edge detection to save computational resources.

Figure 5: Correlation between red and blue components of the image in figure 2 after edge detection.

SOFTWARE IMPLEMENTATION

Figure 6: The original image. Note that the order is Blue, Green, Red, going down.

The final process of image registration consists of six steps, implemented in Matlab using the image processing toolbox.

First, the input image is split vertically into three equal component images.

Figure 6 shows the original image, and the left side of figure 7 shows the three component images generated from splitting it.

Second, ten percent is cropped off of each side of each of the component images. This step is very important to minimize false maxima in the correlation due to the strong black and white bands of the component images. Figure 7 shows the component images before and after cropping. Notice that the frames, as well as the scratches and cracks around the edges, are removed.

Figure 7: Red (t), Green (m), and Blue (b) component images before (l) and after (r) cropping

Next, the correlation is calculated between the blue and red images. The correlation is calculated using Equation 1, in a Matlab function described in [3].

$$Corr(f,g) = Re[IFFT(FFT(f) * FFT(g)^{*})] \qquad 1$$

Because this Matlab function uses the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform(IFFT), it is relatively fast, even for the large, 12 megapixel images of this project. Each correlation takes less than seven seconds.

Once the correlation has been calculated between the blue and red images, the coordinates of the maximum value pixel in the correlation are used as offsets to translate the red image. The translation is accomplished using Matlab's circshift() function. The process is then completed for the green image, which is correlated with the blue image and similarly translated.

Once the red and green images have been translated, all three images have been aligned to the blue image. Creating a color image from the three is accomplished with Matlab's cat() function, which concatenates the three images into a single color image.

RESULTS

Despite the simplicity of this method, it proved rather effective.

Figure 8: A color composite image generated using the method described in this paper.

Figure 8 shows the resulting color image from the component images used as examples throughout this paper. Note that in order to avoid changing the artistic compositing of the picture, the image is not cropped as it was for processing. This task is left to the user.

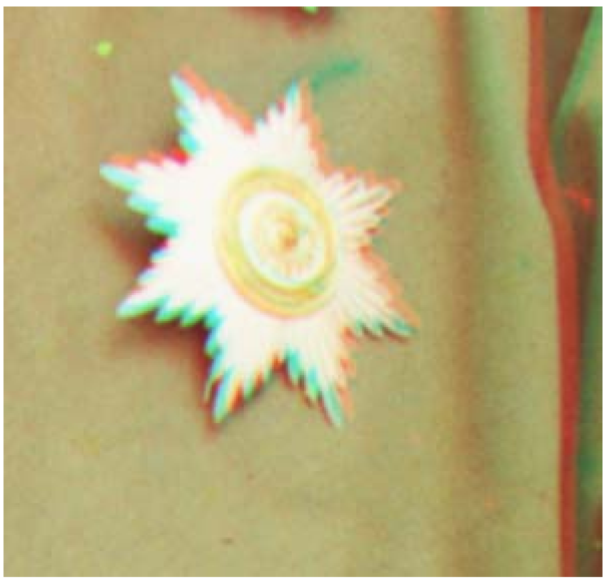

Figure 9: One section of figure 7, expanded to show detail.

The blown-up section in Figure 9 how closely the layers are aligned. Misaligned color layers cause a tell-tale “ghosting” of edges. Though this ghosting is visible here, it is minimal—no more than a ten pixels out of over three thousand in each dimension. Figures 10 through 15 show similar results for other images from Gorksy’s work.

Figure 11 clearly shows that some of Gorsky’s images have very poor white balance. This method does not perform any color correction. Color correction would be an excellent direction for future work on this project, but time did not allow for it to be implemented.

Figure 10: An original Gorsky image

Figure 11: The color composite created from Figure 10.

Figure 12: A blown-up section of Figure 11, showing a complete lack of "color ghosting."

Figure 13: Another Gorsky original.

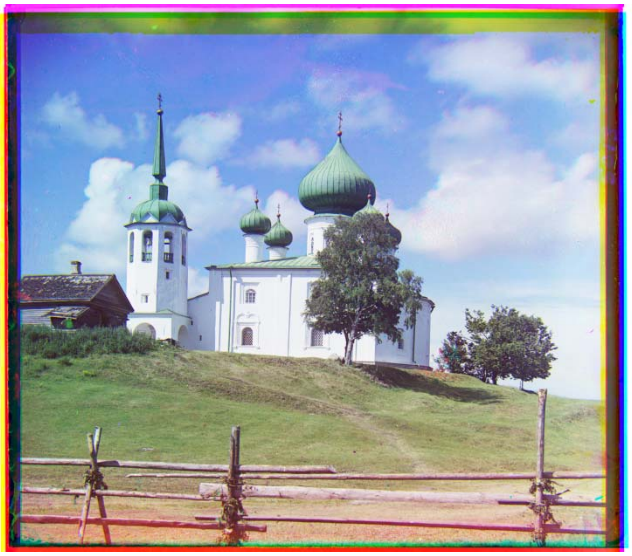

Figure 14: Color composite created from Figure 13.

Figure 15: A blown up section of figure 14 showing minor ghosting.

SUMMARY

Red, green, and blue color images were registered and composited into color images. The method used to register the images was simple translation using offsets determined by performing correlation between the component images. The results were visually impressive and showed color misalignment of about ten pixels over a three-thousand pixel distance, less than 1% misalignment. Future directions for this project include color correction based on white-balancing techniques.

ACKNOWLEDGMENTS

This project was supported by Professor Frank Merat, and was done as part of his class EECS 490 at Case Western Reserve University

REFERENCES

[1] Library of Congress, "The Empire that Was Russia: The Prokudin-Gorskii Photographic Record Recreated," 23 July 2010. [Online]. Available: http://www.loc.gov/exhibits/empire/gorskii.html. [Accessed

20 October 2011].

[2] G. Ward, "Fast Robust Image Registration for Compositing High Dynamic Range Photographs from Handheld Exposures," *Journal of Graphics Tools*, pp. 17-30, 2003.

[3] R. C. Gonzalez, R. E. Woods and S. L. Eddins, Digital Image Processing Using MATLAB, Gatesmark Publishing, 2009.