

Shroud of Turin Research Project:

Bloodstain Identification and Analysis

Objective

The objective of this research project was to examine the bloodstains on images of the Shroud of Turin, and to compare various methods of identifying and classifying the bloodstains in order to find the most accurate way to detect them. Identifying the bloodstains requires a method that can separate them from other markings on the image, such as water stains and scorch marks; therefore, identifying these other markings became an important part of the process. Another part of the process was dividing the image up into regions of interest, in order to study specific bloodstains more closely.

The next part of the objective was to map the blood density, either by using the color data in the normal images or the luminance data from transmitted-light images of the Shroud. Information on blood density could be useful to future Shroud research for many purposes, such as tracing the flow of the blood or learning more about the state of the body at burial.

Technical Approach

The primary technology used by this experiment was ImageJ¹, an extensible image processing application. Several image-processing methods were used to identify the bloodstains, many of which involved writing custom ImageJ plugins.

The first approach used was to manually select the RGB values of blood pixels in the Shroud images. Once these values were obtained, a plugin would be built that would select RGB values within this range and turn them red. This turned out to be a difficult method to use; the RGB values of the bloodstains did not differ significantly from those of the rest of the image, and were extremely similar to those of the scorch marks, which made it difficult to tune the RGB values precisely enough to identify only the blood. This method also created a lot of unwanted noise in the image.

An advancement on this process was a custom RGB Selector plugin, which provided an interface that allowed a user to select a range of RGB values on a set of sliders, and dynamically updated the image as the sliders were adjusted, allowing the RGB values to be carefully fine-tuned. In conjunction with this, several filters were applied to the image to remove noise and make the bloodstains stand out more—specifically, an Unsharp Mask and a Median filter were used. The RGB Selector would highlight areas of the image that fell within the specified color range and darken those outside it, and could create a binary image from the RGB values; this allowed the bloodstains to be highlighted more accurately.

Another plugin created for this project was the Color Sequencer, which allowed the user to click on a sequence of pixels to define a series of shades, which would be sorted by

luminance and then highlighted in shades of red. This plugin could be used to highlight the density of the blood, and it also provided a somewhat more accurate method of identifying the blood than the RGB Selector. It determined if a color was within the range primarily by luminance, but it also checked for hue within a certain tolerance, which could be adjusted with a slider.

Several other plugins were written for the experiment as well, including a Color Sobel Edge Detector—a “3-dimensional” convolution matrix that was applied to all 3 color channels of the image and created a colored edge map. This edge detector provided interesting information about the color distribution of the image, but by itself it was not useful for identifying anything. Another plugin—this one was downloaded, not built—was a version of the Canny edge detector², which was also applied to all 3 color channels separately. This created a contour map of the image which could be used to identify the bloodstains in a significantly different way than the color-based methods used previously.

Narrative

The first attempts at identifying the bloodstains were to manually select RGB ranges for the bloodstains and turn them a pure red color to identify them. This became extremely time-consuming and had a large chance for error. The Color Selector plugin was built to help us find the RGB ranges of the bloodstains; this worked well enough, but it was difficult to distinguish the bloodstains from the scorch marks on the shroud. It was necessary to find a method that would provide a clear distinction between these markings.

One reason for this was the type of image being used; the Shroud of Turin has been photographed only a few times over the course of its existence, so there was a limited selection of images to choose from. The primary image used for this project was the Durante 2002 (“restoration”) image; other images were tried, but in general they only made it more difficult to identify the bloodstains. In order to allow for careful study of specific parts of the image, the image was cropped into smaller sections. The face, back of the head, arms, ankles, back, chest and front left side, and back were all separated into their own images.

The Color Sequencer plugin was built to select a series of colors by luminance and use this to both identify the blood and sort it by density. This plugin was used on each of the regions of interest; although its original purpose was simply to map the blood density, it actually provided a more accurate identification of the blood than any other plugin so far, and did not lump the scorch marks with the bloodstains nearly as much. It was still important to remove the scorch marks, though, and the Color Sequencer could be used to identify and remove them as well, although this was eventually abandoned as it proved too time-consuming.

Other experiments were attempted that involved adjusting contrast and brightness, applying different LUT’s, and isolating the hue channel of the image, but they yielded very little

useful data. It was also attempted to find plugins online that could be useful to the project, but nothing of any significance was found. Transmitted-light images of the shroud were used in an attempt to map the blood density, and while, in theory, they should have yielded more accurate information about the bloodstains, in practice they displayed even more similarity between bloodstains and scorch marks, as well as large amounts of noise, making an accurate analysis difficult.

A color version of the Sobel edge detector was built to help determine where the bloodstains ended and the scorch marks; however, its output was not very helpful, and did not appear to show any significant difference between the bloodstains and scorch marks at all. However, it provided the inspiration to use another plugin, a Canny edge detector, which was found online. This plugin created clearly defined contours that separated sections of the image, and finally provided a clear distinction between the bloodstains and the scorch marks. At first, it was assumed that this could be used to remove the scorch marks from the image; however, this became a very time-consuming process (contours had to be filled manually using GIMP) and still did not remove the scorch completely. Instead, the contours that enclosed the bloodstains were filled, which allowed a remarkably accurate identification of the major bloodstains (although fainter stains were ignored completely). The Color Selector was applied to these isolated bloodstains to highlight differences in density, providing a reasonably accurate final image of the major bloodstains.

Conclusion

After several methods of bloodstain identification were attempted, no one method seemed to clearly work better than the others. A combination of the Canny-edge-detection contour tracing and the Color Selector seemed to be the most useful method for tracing the major stains, but it completely ignored fainter stains; other methods, or perhaps a more sensitive edge detection filter, would be required for these. Hence the use of regions of interest: it is almost impossible to find a method that works equally on all parts of the Shroud image, therefore different image processing techniques would need to be applied to different stains to get a truly complete map of the blood on the Shroud.

Acknowledgements

¹ ImageJ

(<http://rsbweb.nih.gov/ij/>)

² Canny Edge Detector Plugin – Written by Thomas Boudier

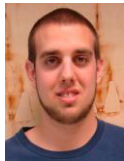
(http://imagejdocu.tudor.lu/doku.php?id=plugin:filter:edge_detection:start)

Recommendations

Future researchers could expand on the work of this project in the following ways:

- Apply the contour-tracing technique to the entire image to create a complete map of the bloodstains, perhaps using this in conjunction with another method to detect the fainter stains.
- Experiment with contour tracing in more ways: use the Canny edge detector with different parameters, try to trace the contours created by the Sobel edge detector (which picks up fainter bloodstains), etc.
- Research the origins of the blood (at what time each of the stains might have been created, the cause of the wounds, etc.) and categorize the bloodstains. This could have some effect on the density mapping.
- Attempt to trace the blood back to its origins, perhaps with some kind of fluid dynamics simulation.

Biographies



Justin Eckard is a sophomore Computer Science major, Class of 2012 at Bridgewater College.



Morgan Foster is a sophomore Mathematics major and Computer Science minor, Class of 2012 at Bridgewater College.



Adam Nelson is a freshman with a dual major in Physics and Computer Science, Class of 2013 at Bridgewater College.