Development of Image Processing Tools for Analysis of Laser Deposition Experiments

Todd Sparks
Department of Mechanical and Aerospace Engineering
University of Missouri, Rolla

Abstract

Microscopical metallography is a well established field with a long history of techniques for measuring features. Modern computational power allows much more rigorous investigation than simple area fractions and intercept measurements. This paper details the development of image processing tools using GNU Octave, a free, open source numerical computation package. Proposed in the paper is a method for characterizing features within a digital microscope image.

Introduction

Using a computer for metallographic analysis can eliminate the subjectiveness inherent in old, manual analysis techniques as well as speed up the process. Batch processing of many images can save many hours of manual labor and eyestrain. Tools for doing this sort of work already exist, but they tend to be expensive, commercial products. Others are free (Scion Image), but are limited in that they are closed-source, and only work on specific platforms. By using GNU Octave, an open source numerical computation package similar to Matlab, the analysis tool can be freely distributed and easily modified to suite many needs.

For testing purposes, a sample image was chosen from a previous metallographic study done on laser deposited H13 samples. The chosen image is shown below in Figure 1. The goal of this first step in the development of a metallographic analysis routine is to identify interesting regions in the image for later analysis.

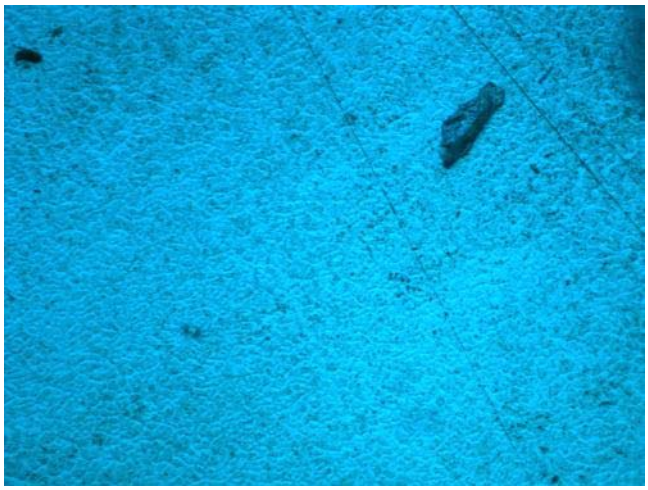

Figure 1 – Example Image

Evaluation of Common Grey Image Filters

There are several filters commonly used in image processing. Six standard filters were evaluated: Gaussian, Mean, Median, Smoothing, Gradient, and Laplacian. Since all of the filters assume a gray image, the original image was reduced to 256 shades of gray

(Figure 2).

Figure 2 – Example Image in 256 Grays

Figures 3-6 show the effect of applying filters designed to blur or smooth the image. All four filters operate by removing or averaging color information. The Gaussian, median, and mean filters are visually indistinguishable from each other, while the smoothing filter seems to blur out more of the sample's surface.

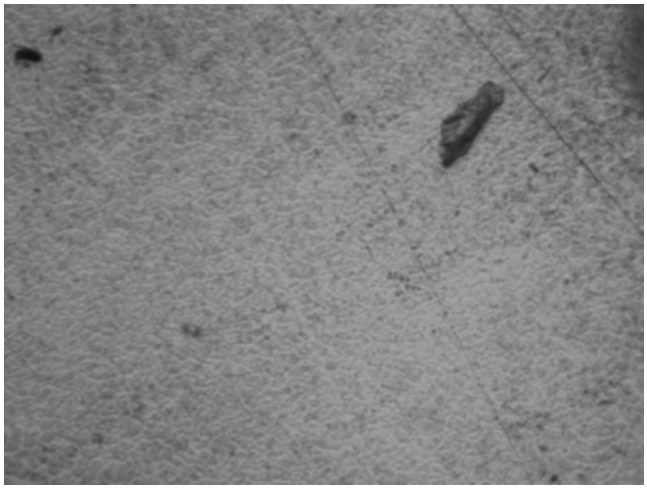

Figure 3 – Example Image with Gaussian Filter

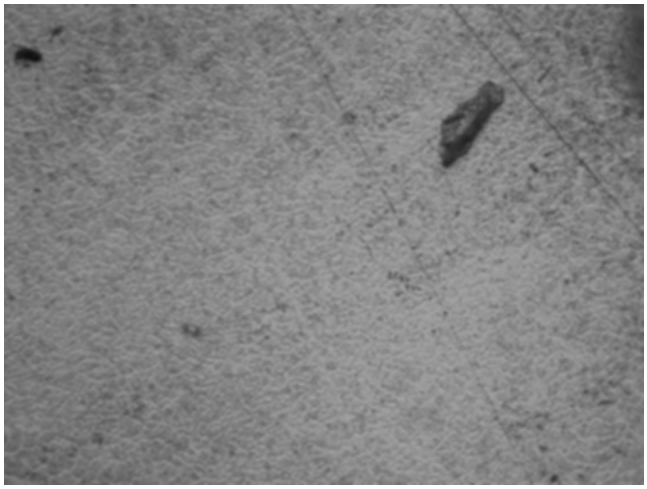

Figure 4 – Example Image with Mean Filter

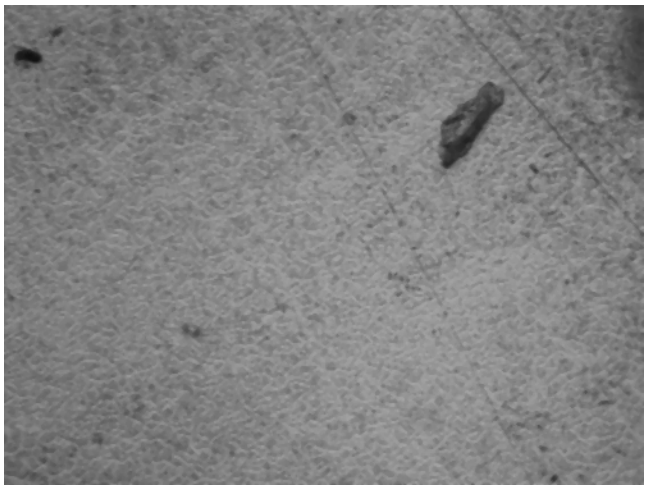

Figure 5 – Example Image with Median Filter

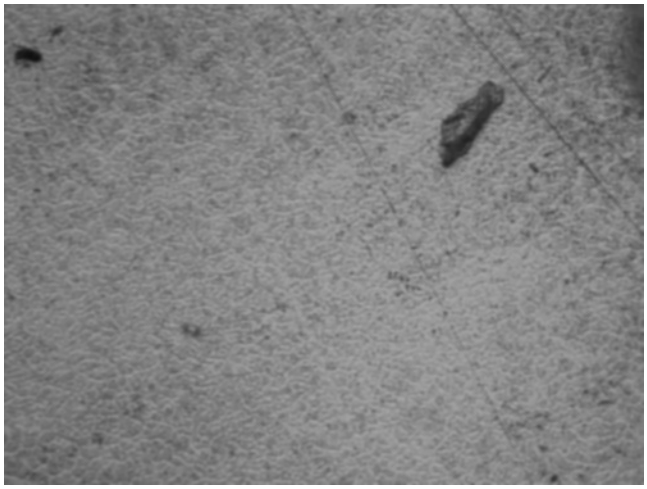

Figure 6 – Example Image with Smoothing Filter

Figures 7 and 8 show the effect of applying filters designed for detecting edges within the image. Both the gradient and Laplacian filters seem very sensitive to the texture of the image, while missing the larger features that are obvious to the human eye.

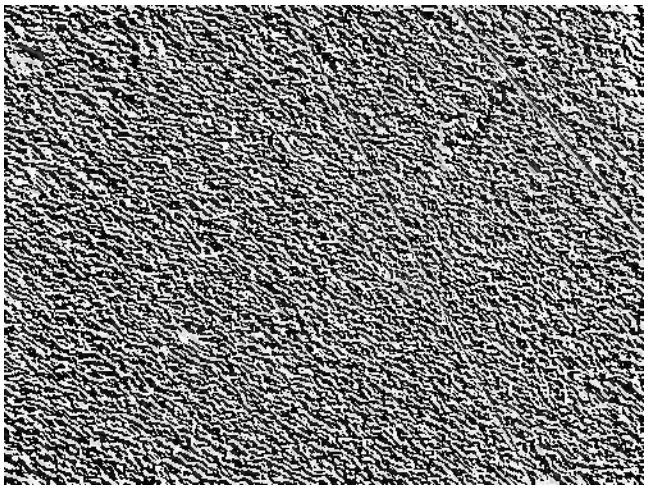

Figure 7 – Example Image with Gradient Filter

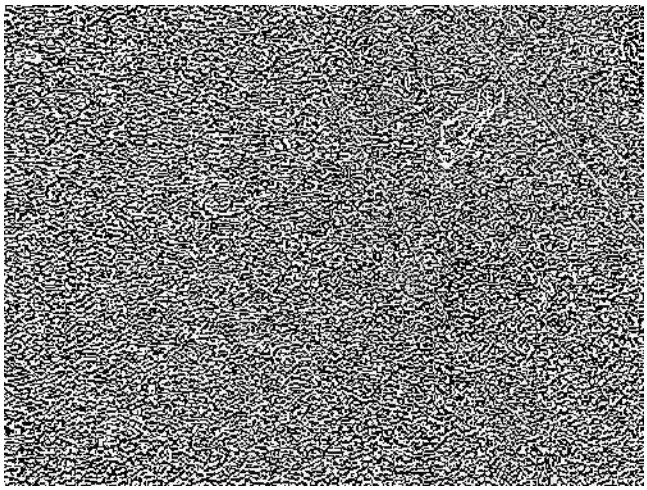

Figure 8 – Example Image with Laplacian Filter

Edge and Region Detection

The edge finding image filters tested above did not successfully find the edges of the obvious objects in the image. Figure 9 shows the result of summing the magnitude of the color gradients in each of 8 possible directions. This does an excellent job detecting the edges of the objects in the image, but the image has become very cluttered due to the effect the filter has had on the overall texture of the image.

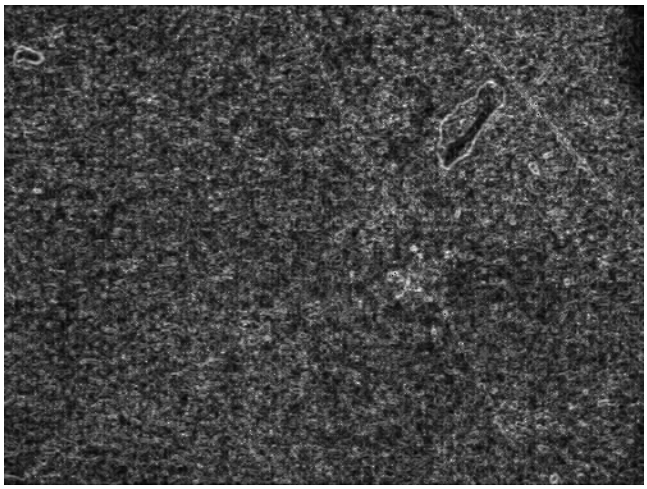

Figure 9 – Example Image with Edge Filter

Image 10, below, shows the result of applying he smoothing filter before the edge filter. The blurring effect of the smoothing filter eliminates many of the smaller objects and makes the edge defined around the larger objects more clear.

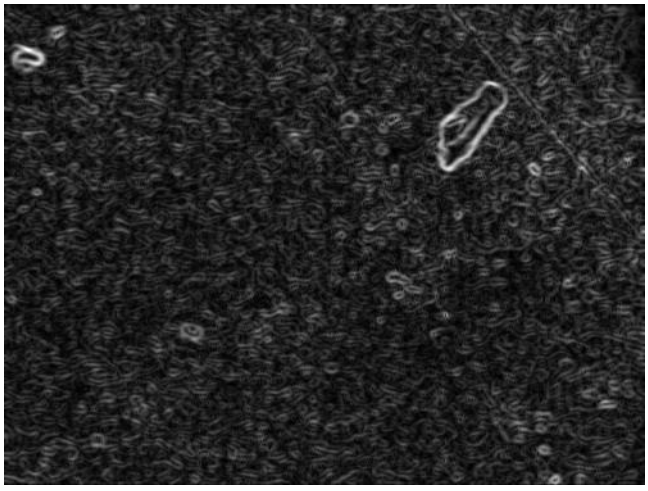

Figure 10 – Example Image with Smoothing and Edge Filters

Region Detection

Region detection was accomplished through a fairly simple algorithm. First, the image from Figure 10 was thresholded to produce a binary image (Figure 11). This results in an image where the white pixels represent borders of objects and the black pixels are the objects themselves. To separate the regions partitioned by the white pixels, the following algorithm was used:

1. Select an ungrouped black pixel at random.
2. Perform at 8-way flood fill operation on the selected pixel.
3. Assign the flooded pixels as a new group.
4. If ungrouped black pixels exist, return to step 1.

The result of the region finding algorithm is shown below in Figure 12. The regions have been color coded according to the average color of the identified region from the original

image.

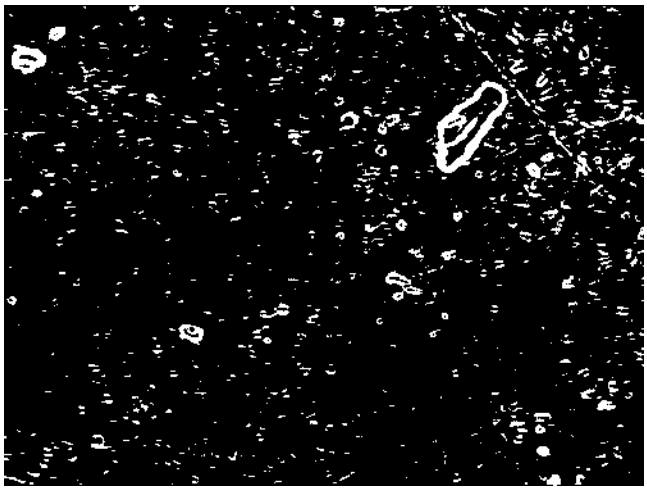

Figure 11 – Example Image Thresholded After Filtering

Figure 12 – Detected Regions

Considerations and Further Work

The method outlined above is effective in identifying large objects within the image, but fails to pick out smaller or less clear objects. The edge detection routine needs to be refined to create more enclosed regions from less information. Another possibility would be to add a routine to close the nearly-enclosed regions. Increasing the image resolution would also alleviate the problem, but is less desirable since it would increase the computation time.

The next step for this project is to build routines to do direct measurements on the detected regions. Area, circumference, length, width, and other such direct can be used along with some simple calculated values such as circularity or more complex statistics describing the region's texture to categorize the region. Such a database, when complete would allow for completely automated, fast analysis of images.

Another item of interest is the possibility of moving the graphics computations to the system's GPU. This would turn a process that takes a few minutes into something that takes a trivial amount of time. A GPU would not be a practical solution for processing a single image, but it would be a boon to batch processing.

References

Friel, John, et al. *Practical Guide to Image Analysis*. ASM International, 2000.

Kenny, Philip J, ed. *Image analysis and metallography : proceedings of the Twenty-First Annual Technical Meeting of the International Metallographic Society*. Columbus, Ohio, USA : The Society ; Metals Park, Ohio, USA : ASM International, c1989.

Eaton, John W. "Octave Home Page." University of Wisconsin. 17 Apr. 2004. http://www.octave.org

"General-Purpose Computation Using Graphics Hardware" 20 Aug 2004. http://www.gpgpu.org/