Simple Pattern Recognition via Image Moments

Matthew Brown
mattfbrown@gmail.com

Matthew Godman
mgodman@nmt.edu

20 April, 2011

Electrical Engineering Department
New Mexico Institute of Mining and Technology

Abstract

This report discusses a mathematical method that uses image moments to detect features and patterns. The method considered will be implemented on a general purpose graphics processing unit in order to understand the limitations of this spacific process. We take this method one step further, using a pattern mask correlated with a larger image, to detect feature locations in an image.

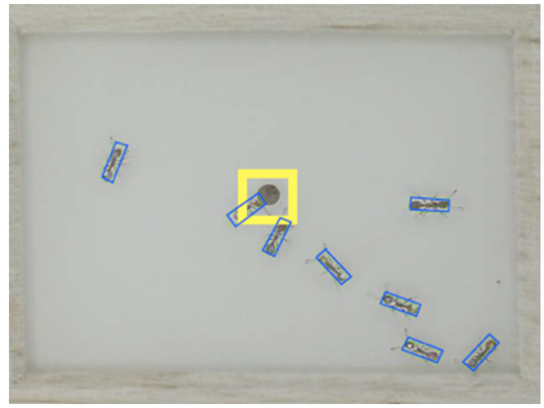

Figure 1: Ants

1 Introduction

Digital Image Processing uses many mathematical methods of analysing and filtering digital images to produce desired effects. This report considers the method of image moments to detect patterns or features in a large image. In the body of the text moments will be discussed in some detail. Several variants are considered and the variant that provided the best results is discussed further. Finally results and a conclusion will be at the end.

There are many methods used in pattern recognition. Figure 1 shows one application of pattern recogntion in which ants are identified, highlighted and tracked in an area. In many cases an image is split up into a tree description of the image (shown in Fig.2). This can be done by analysing the colors associated with certian parts of the photo and associating those colors to things like water, grass, streets, and so on. This is useful if a general description of an image is all that is needed.

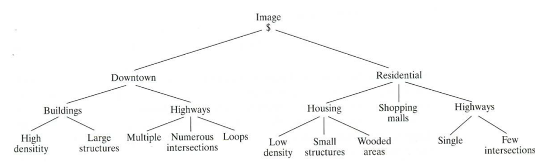

Figure 2: tree description

The banking industry uses pattern recognition when reading checks. This system incorporates a specially designed numbering font (MIRC font, shown in Fig.3) and a camera to read checks, money orders, etc [1]. This recognition method breaks down the two dimensional image into a one dimensional description of the same image to reduce complexity in the recognition method.

The applications of feature or pattern recognition has improved some of the high tech imaging systems in use today. Feature detection is used in image stabilization for unstable environments that introduce shaking. Object identification is still in its in-

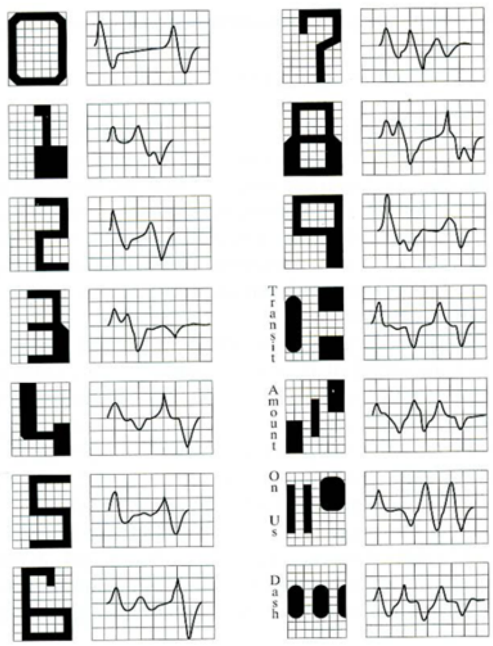

Figure 3: MICR font

fancy but aids security and tracking systems. A lot of the time, stabilization, indentification, and tracking are used all at once, news copters is just one example.

Figure 4: OJ

2 Project Scope

For the course of this project we implemented and tested the method of simple feature recognition via image moments on a general purpose graphics processing unit (GPGPU). This process starts usually starts with reducing the image complexity by converting an image to greyscale if necessary, then using a mask such as a Sobel Mask to detect edges to create a boundary edge map of the pattern of interest. We chose to start with binary images and goes straight to the image moment calculations to reduce the size of the project. It is assumed that patterns or features are not overlapping or too close together in order to reduce the complexity of the problem.

3 Method of Moments

This basis of this project was to calculate image moments from 2D binary images. An image moment is a particular order of the image pixel intensities weighted average. The moments can be invariant to scaling, translation, and rotation. One such set of these invariant moments are the Hu set[2], a commonly used set for applictations such like this one. We followed a procedure outlined in a paper by Mercimek, Gulez and Mumcu[3] that uses this set to detect 3D objects. We were able to get very good results from the first 6 of the 7 moment invariants in the Hu set that were invariant to translation and rotation. Scale however did not seem to be working so we kept exploring.

We found a procedure by (insert name here) that used complex version of the moments that we were using [4]. Using this set, we were able to detect our desired feature even with scaling. . A feature or simple pattern mask can be correlated with a larger image to detect the locations of features. Moments of an image are found via the following equation,

$$M_{uv} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^u y^v f(x, y)$$

Where M is a particular moment indicated by the subscripts u and v of a binary image f(x,y). The components of the centroid and it's area can be found from fundamental moments as follows.

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}}$$

$$A = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

The complex central moment equation comes from a book that was published in 2009 by Flusser, Suk, and Zitova[5]. Compared to the other moment equations that were found for this project, the complex moments showed the best results when the pattern was scaled in size. Rotation and translation didn't give us any problems.

$$\mu_{uv} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} ((x - \bar{x}) + i(y - \bar{y}))^u \\ ((x - \bar{x}) - i(y - \bar{y}))^v f(x, y)$$

The scale invariant moments are found from the complex central moments and the feature area.

$$s_{11} = \frac{\mu_{11}}{A^2}, s_{20} = \frac{\mu_{20}}{A^2}, s_{21} = \frac{\mu_{21}}{A^2.5}, \\ s_{12} = \frac{\mu_{12}}{A^2.5}, s_{30} = \frac{\mu_{30}}{A^2.5}$$

Finally, we get 6 rotation invariant moments from the scale invariant moments.

$$\begin{aligned}
r_1 &= real(s_{11}) \\
r_2 &= 1,000 * real(s_{21} * s_{12}) \\
r_3 &= 10,000 * real(s_{20} * s_{12}^2) \\
r_4 &= 10,000 * imag(s_{20} * s_{12}^2) \\
r_5 &= 100,000 * real(s_{30} * s_{12}^3) \\
r_6 &= 100,000 * imag(s_{30} * s_{12}^3)
\end{aligned}$$

These three equations can be combined and compared with a mask to detect the presence of a feature from an image window. This method of feature detection is invariant to translation, scaling, and rotation.

4 Results

The results that we obtained were much better than expected. The following figures show all of the results that we obtained in a single window. Figures 5-10 show the results we got from single window results. The later figures then show the result that we obtained from parsing through a larger image. Figure 5 shows the result we obtained from translating our original image (left) to the lower right of the corner (right).

We were able to show by comparing the moment statistics of the two images a match of over 99%.

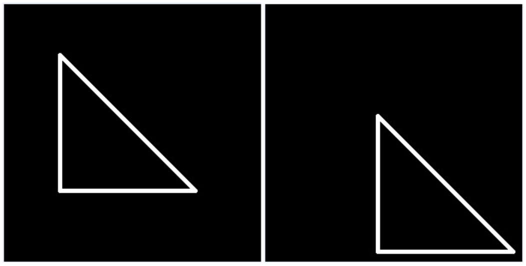

Figure 5: left: orginal image, right: translated image. Result: 99.175 Percent Match

Figure 6 shows the result that we obtained from rotating the image. Again we were able to obtain a match of over 99%. In all of these single window cases we used images that were 300x300 pixels and used 5 of the complex moments discussed earlier. Figure 7 shows a 92% match when we scaled the image down by 75%

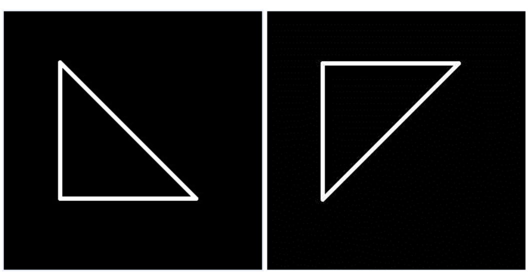

Figure 6: 99.937 Percent Match

Figure 8 shows the result when we altered the image by all of the previous methods: translation, rotation and scaling. In this case we sacled the image by 200%, rotated by 90, and translated the image to the lower right.

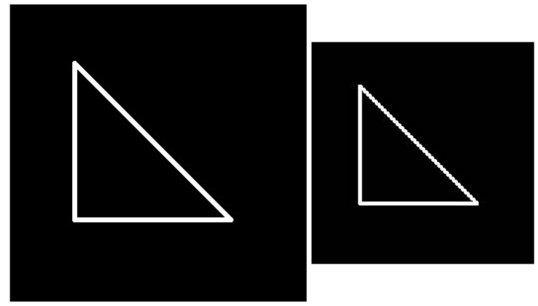

Figure 7: 92.079 Percent Match

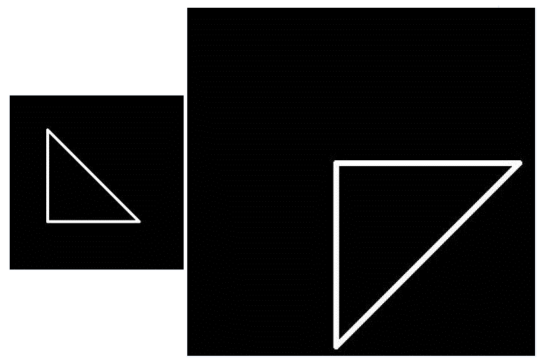

Figure 8: 99.977 Percent Match

Figures 9 and 10 show the results that we got when we compared our image to two patterns that we weren't looking for, namely a circle and a square respectively. Both of them showed very high mismatches as can be seen in the Figure captions.

The last thing we wanted to try was to see if we could parse through an image and see if we could identify somehow the location of patterns that we wanted to identify. To do this, we made an image with several correct and incorrect patterns where we also altered the correct images (triangles) by rotating or scaling them.

Figure 11 is the test image that we cre-

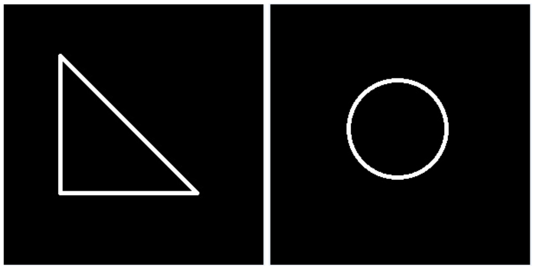

Figure 9: 10000000000 Percent Error

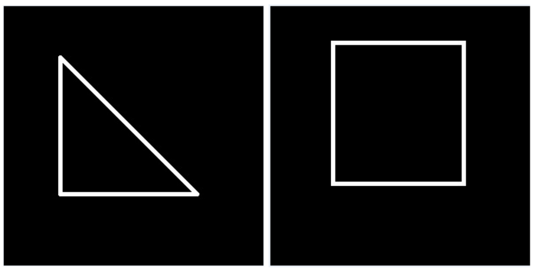

Figure 10: 32380 Percent Error

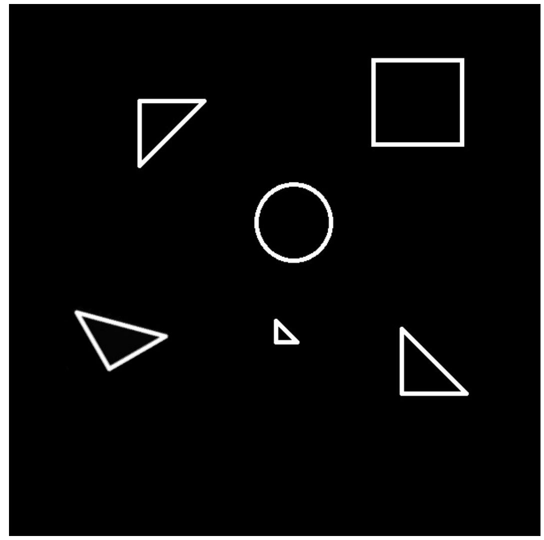

Figure 11: A test image we parsed over

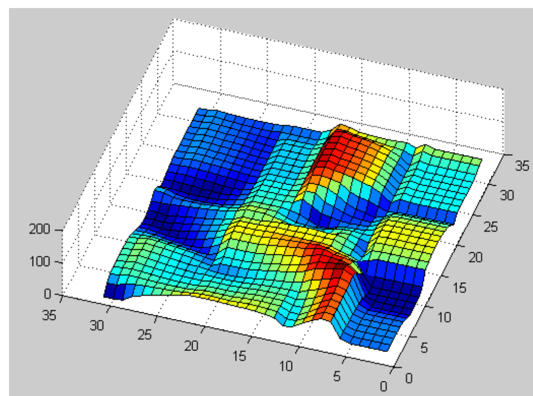

Figure 12: Contour map plotted in MATLAB of relative error to desired pattern match.

ated to try this. Figure 12 shows the result that we plotted in MATLAB. We were able to show that the triangles that were rotated, but not scaled we easily identified in the image, although the scaled one did not show up. This was due to the fact that in order to correctly identify it, the window would have to scaled to the scale of the pattern, which we did not do. To do this, one would have to recognize that a pattern was present, and use a learning algorithm to try different window sizes to see if a match was a correct one.

5 Conclusion

Through the use of complex invariant moments, we were able to identify quickly and to a high degree patterns that were translated, rotated and scaled. We showed that patterns that we were not looking for did not come close to giving a false positive. We also showed an application of parsing through an image to locate patterns of interest. All of the coding for the project was done in MATLAB with the use of GPUMAT to speed up the process.

References

[1] Gonzalez and Woods, *Digital Image Processing, Third Edition*. Upper Saddle, NJ: Prentice Hall, 2008.

[2] J. Shutler. (2008) Hu invariant set. [Online]. Available: homepages.inf.ed.ac.ek/rnf/CVonline/ LOCAL$_{C}OPIES/SHUTLER3/node8.html$

[3] M. Mercimek, K. Gulez, and T. Mumcu, "Real object recogntion using moment invariants," *Sadana*, vol. 30, pp. 765–775, Dec 2005.

[4] B. Fisher. (2009) 2d binary image moment invariants. [Online]. Available: homepages.inf.ed.ac.ek/rnf/CVonline/ LOCAL$_{C}OPIES/FISHER/MOMINV$

[5] B. Z. J. Flusser, T. Suk, *Moment and Moment Invariants in Pattern Recogntion*. John Wiley and Sons, 2009.