AUTOMATED CRATER DETECTION AND COUNTING USING THE HOUGH TRANSFORM

M.J. Galloway[1], G.K. Benedix[2], P.A. Bland[2], J. Paxman[3], M.C. Towner[2], T. Tan[3]

[1]Faculty of Engineering, Computing & Mathematics, University of Western Australia
[2]Department of Applied Geology, Curtin University, GPO Box U1987, Perth, WA 6845, Australia
[3]Department of Mechanical Engineering, Curtin University, GPO Box U1987, Perth, WA 6845, Australia

ABSTRACT

A manual process for detecting and counting craters on the surface of a planetary body becomes impractical when attempting to survey a large surface area. Similarly, existing automated methods that are effective for specific areas of focus are also impractical for a large data set. We report on the work completed so far in developing a crater detection system to automatically detect craters down to sub-km sizes, across a large portion of a planetary surface. Specifically, we assess the performance of a Hough Transform (HT) for the application and in particular the influence of its preprocessing edge detection phase. Tests are performed on high resolution images of the Martian surface, anticipating a large scale crater counting application for crater chronology on the surface of Mars.

Index Terms— Mars, Craters, Hough-Transform, Surface-Dating, Automation

1. INTRODUCTION

Crater detection and counting of planetary bodies has important scientific applications including the chronology of surface geology [1, 2]. In particular, densities of superposed craters were used to model crater ages in [3], however this was achieved using a database of manually catalogued craters. Currently, the smallest craters recorded in the available database is around 1km in diameter [2].

The Mars Reconnaissance Orbiter's HiRISE instrument provides very high resolution images of 30cm/pixel (for altitude of 300km) [4]. Another high resolution image source is data from the instrument aboard ESA's Mars Express mission. HRSC images achieve a resolution up to 10m/pixel (250km altitude) [5], and as of January 2014 has covered about 90% of the Martian surface (as reported by official HRSC webpage `http://www.dlr.de/dlr/en`). A combination of this data will provide an opportunity to count craters down to sub-km sizes, with a good coverage of the planet's surface.

With reduced crater diameters, however, comes an increase in frequency and consequently a significant increase in time required to manually identify such geological features, rendering the process impractical for surveys covering large surface areas. Thus there is a need for a system which can identify sub-km impacts automatically.

One study exploring automated crater counting reported a modest rate of 93μs/pixel using a single processor [6]. For a single Mars Express HRSC image, this amounted to 14hrs of processing time. We can scale this to HiRISE imagery for comparison. Take HRSC image h8304_0000_nd3, which has pixel dimensions of 11211 x 47509, and compare it to HiRISE image ESP_017258_2045_RED, with pixel dimensions 19582 x 67489. The HRSC image in this case again amounts to 14hrs, while the HiRISE image could take 34hrs. For a focussed survey of a specific area, this is an effective method with good performance (a reported detection rate of 70% [6]), however to process only 100 HiRISE images could take up to 5 months of processing time. For this reason we envision the integration of an automated process with a supercomputing environment. This paper outlines the preliminary measures that have been taken in order to establish the crater detection itself.

Automated crater detection has been explored using various methods, including the HT. Michael [7] used HTs with elevation data from the Mars Orbiter Laser Altimeter (MOLA). While this method proved effective for craters larger than 10km in diameter, the spacial resolution of MOLA imagery (up to 64pixels/degree), is unlikely to be sufficient for smaller features. Sawabe et al. [8] combine a number of approaches in detection on the Lunar surface, only using the HT for larger incomplete (degraded or interrupted edges) craters.

Stepinski and Urbach [6] developed a sub-km crater *candidate* detection method, using mathematical morphology and shape filtering. As outlined in [9], this method works very well for detecting candidates. For testing, we developed a windowing method to pass the HT an image window that either contained a crater or non-crater feature. This methodology would be similar to applying a HT to the crater-candidate windows generated by Stepinski and Urbach's method, and our results suggest it would perform well in these circumstances.

This paper reports on the progress made so far in developing the desired system using a HT, with initial tests performed on HRSC images. We investigate the influence of the preprocessing edge detection on the HT performance, and discuss potential improvements to be explored in future work. A method was developed to segment images into positive and negative windows, which are required for the classification test used to produce comparative statistics. In addition, our approach keeps in mind large scale automation of the workflow in a parallel supercomputing environment.

2. METHOD

2.1. Hough Transform

The HT is an algorithm used for detection of features in digital imagery. Originally developed for the extraction of lines, the HT may be adapted to arbitrary shapes by exploiting the parameterization of the shape's bounding curves [10]. Of particular interest is the HT applied to circular feature detection, known for its robustness when faced with occluded or incomplete boundaries [10].

The HT fundamentally relies on edge information in an image, which is usually produced by calculating local intensity gradient magnitudes. Thresholding this data produces a binary image, where pixels of value are termed *edge pixels* and represent the edges found in the image.

Using a HT for circles, each edge pixel is transformed into its parameter space as a right circular cone, the intersections of which identifies edge pixels lying on the same circle defined by the parameters [11]. The parameter space may be quantized into an *accumulator* array, which in this case is three dimensional and acts as a voting accumulator for circular features.

Storage efficient alternatives exist, reducing the accumulator into two dimensions. By making use of edge orientation, votes are cast only along normal lines at each edge pixel. Intersection of these normal lines will occur most frequently at the centre of circular features within the image, allowing circle centres to be identified by local maxima in the accumulator [10].

2.2. Canny Edge Detection

Edge detection in digital imagery relies on locating areas with contrasting values. In a grayscale image, the pixel values represent only intensity and so can be exploited rather simply to obtain *changes* in intensity by measuring their rate of change, otherwise known as the gradient magnitude (g). Once this magnitude is calculated at each point, pixels with g less than a specified threshold are deleted or set to 0, thus leaving pixels that are likely to be edges.

Canny edge detection is more careful about which pixels it chooses to cull, using a method called hysteresis thresholding [12], which takes an upper and lower threshold, henceforth referred to as $C_{\max}$ and $C_{\min}$ respectively. Pixels with g greater than $C_{\max}$ are unconditionally accepted as edge pixels, while values of g less than $C_{\max}$ but greater than $C_{\min}$ are accepted *if* adjacent to a pixel with g greater than $C_{\max}$. Any pixel with g less than $C_{\min}$ is set to zero [12]. This results in well connected edges or contours, which is desirable for a HT. Additionally, Canny edge detection performs an edge-thinning algorithm called *non-maximum suppression* (NMS), leaving thin edges by taking only pixels with g values that are local maximums (relative to its immediate neighbourhood) [13].

2.3. Implementation and Workflow

The OpenCV library includes an implementation of a circle HT. The testing in this paper is performed using this implementation, which is a version of the 2-1 Hough Transform (21HT) as described by [14]. The 21HT uses the two dimensional accumulator referred to in Sec. 2.1, accumulating votes along the edge normal. OpenCV also contains an implementation of Canny edge detection, which it uses in the preprocessing stages of the HT. We experimented with blurring images prior to edge detection, however any benefits in performance were not evident.

3. EXPERIMENT AND RESULTS

3.1. Data

The initial test data was taken from the Mars Express HRSC instrument's image database. 15 regions (2598 by 2664 pixels) were cropped out of 5 nadir images (h8304_0000, h0466_0000, h2530_0001, h9615_0000, and h7347_0000 [5, 15]). For each of these 15 image regions, craters were identified by inspection, with their radius and x, y pixel coordinates recorded. Crater identification was restricted to a minimum of a 5 pixel radius, at which point the image quality limits the ability to resolve crater features under edge detection.

The performance analysis (discussed further in Sec. 3.2) required a classification system that encapsulated a positive/negative test. Detection data is not inherently classifiable in this way, due to the difficulty in defining a negative test. Table 1 illustrates the test outcomes required, i.e. *True Positives* (TP), *False Negatives* (FN), *True Negatives* (TN), and *False Positives* (FP).

Table 1: Classification for detections with ground truth (GT) tests.

	Detected Circle	No Detection
Matching GT Crater	TP	FN
No Matching GT Crater	FP	TN

Difficulty arises when attempting to test the outcomes in Table 1. For example, one method may check for TPs, FPs, and FNs by simply comparing ground truth data with detection data. A TP occurs when a detected circle has a matching ground truth crater, a FP occurs when a detected circle has no matching ground truth crater, and a FN occurs when a ground truth crater has no matching detection. How then does one test for a TN? It is not reasonable to count a TN as everything that is *not* a crater and was *not* detected; the metric is then dissimilar to the others.

An alternative method is to test the HT's performance on a series of windows, which are regions of images which can be evaluated as either positive or negative; they either contain a crater, or they do not. The classification system follows naturally. We count a TP when a detection occurs in a positive window, or a FN when no detection occurs. For negative windows, a TN occurs with no detection, while a detection will count as a FP.

A window region for a *positive* window is simply a square surrounding a crater, with width and height approximately equal to the crater's diameter. A *negative* window is a square of the image containing no crater (henceforth referred to as negative space). It is desirable to cover the majority of the negative space with window sizes of a similar distribution to the positive windows. This is so that the number of negative windows is comparable to the positive windows, but with a tendency to have more negative ones, which reflects the ratio of negative space to crater covered area.

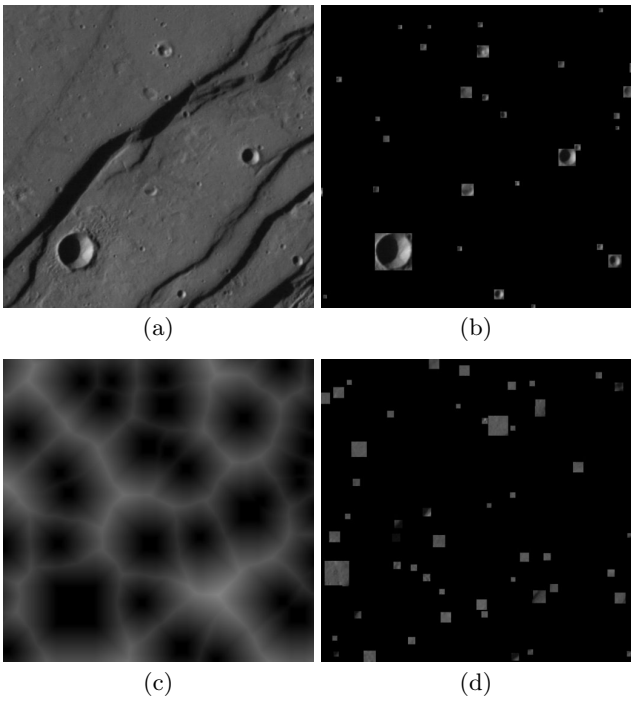

Fig. 1: HRSC image (a) with the corresponding positive (b) and negative (d) ground truth windows. (c) is the distance transform of (b).

Due to the large number of windows required, automation was a desirable method to generate them for the crater detection testing process. Generating positive windows given the existing ground truth crater data is trivial, just requiring an iteration over the ground truth data, assigning a window to each crater. While negative windows could have been generated at random, a more rigorous approach was required in order to avoid overlap between positives and negatives. Furthermore, it was necessary for the windows to have a good distribution of sizes, and a good coverage of the area between the craters.

To achieve this coverage and range in sizes, positive windows were first generated, and projected into a binary image space as pixels of value greater than 0. A distance transform was then performed on this image, producing another image where the pixel intensity represents the Euclidean distance to the closest pixel covered by a positive window. This distance transform is illustrated in Fig. 1c.

After the data from the distance transform is obtained, a random sample of crater diameters is taken, and corresponding windows are generated in areas to which they will fit without intersecting a positive window. This results in a nice spread of negative windows, shown in Fig. 1d.

It is important to note that the performance of the HT in detecting multiple craters, with multiple sizes, across a larger image is not encapsulated by this testing method. It is however a necessary preliminary performance analysis to assess the viability of using a HT. This does not imply that the assessment is irrelevant. Indeed, the workflow may be integrated with such methods used in [6] and [9], using the HT on windows containing crater-candidate features. This will be explored in future work.

Table 2: Canny edge detection values in ROC.

	$C_{\min}$	$C_{\max}$
Curve (a)	10	60
Curve (b)	40	90
Curve (c)	70	120

3.2. Results

Performance of the HT was measured using a Receiver Operating Characteristic curve (ROC curve). The x-axis represents the False Positive Rate (FPR), while the y-axis is the True Positive Rate (TPR). These metrics are similar to those used in [9], however we include the TN quantification. Using the values discussed in Sec. 3.1, the TPR and FPR are calculated as follows:

$$TPR = \frac{TP}{TP + FN}, \tag{1}$$

and

$$FPR = \frac{FP}{TN + FP}. \tag{2}$$

The curve is parametric, with each point being representative of the performance at some varying sensitivity parameter within the detection scheme. In this case, the accumulator threshold was chosen, with results being recorded and plotted for the HT performance across various values. The shape of an ROC curve is indicative of the detection algorithm's performance. Ideally, an ROC curve will include points $(0, 0)$ for the lowest sensitivity parameter, $(1, 1)$ for the highest (i.e. detect everything) and $(0, 1)$ for some optimal parameter value.

Three sets of results were compared using ROC curves (Fig. 2). Each set of results (each curve) represents a different set of $C_{\min}, C_{\max}$ values for the Canny edge detection. By comparing these, we can infer the influence of the edge detection over the HT's performance. Table 2 outlines the values used, which were based around the values of curve (b), selected after a basic observational optimization (what intuitively looked like a suitable amount of edge data). An illustration of the detection performance using parameters from curve (b) is illustrated in Fig. 3.

Each curve's right most data point represents the lowest practical accumulator threshold for OpenCV's HT, which is when the threshold equals 1. Since vote counts are integer increments in the accumulator, the only lower value is 0, which theoretically detects everything and is reflected by the $(1, 1)$ points in Fig. 2. These right most data points are indicative of the influence the C values have over the FPR; the proportion of non-crater data that is being detected as a crater. As the C values are increased from curve (a) to (c), the right most points travel left, reducing the FPR significantly. This reduced FPR however comes with a cost of reduced TPR, and so there is a trade-off between the two.

The inference we make from the curves in Fig. 2 is that the TPR can be increased with a higher amount of edge data, resulting from low C values in this case. However, since decreasing the C values for the Canny edge detection increases the edge data uniformly across the image, we see a rise in FPR. This suggests that an increase in edge data only around craters would be beneficial.

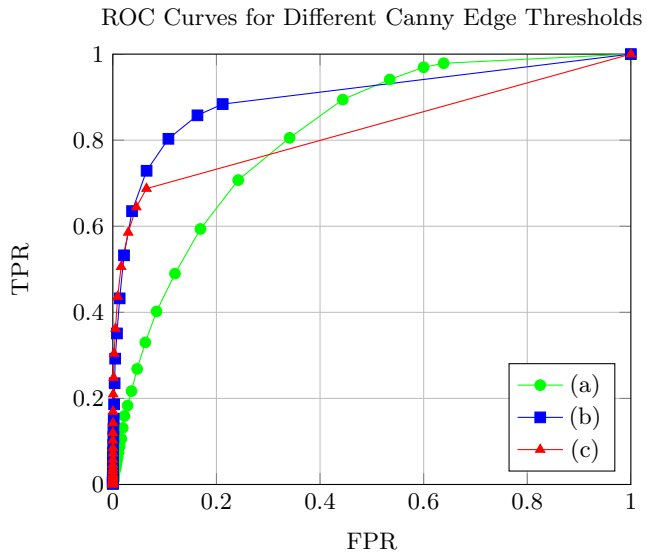

Fig. 2: ROC Curves for Canny edge minimum and maximum thresholds at (a) 10 and 60, (b) 40 and 90, and (c) 70 and 120 respectively.

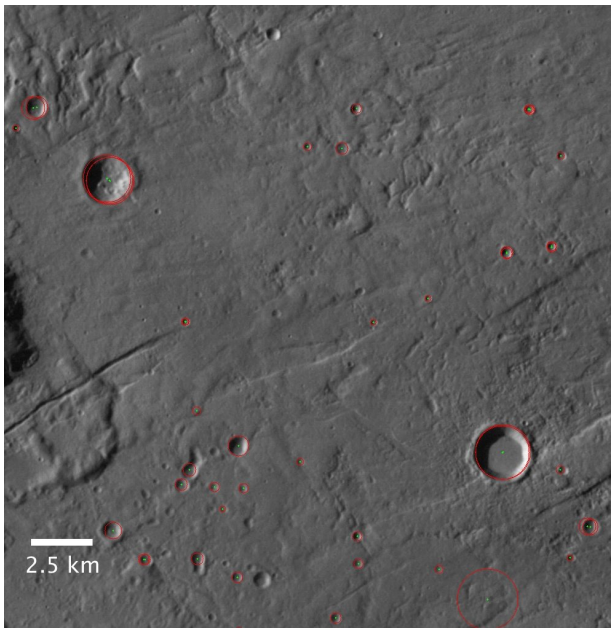

Fig. 3: HRSC image with projected detection circles in red.

4. DISCUSSION AND FUTURE WORK

A method that we are currently developing involves an alternative edge detection stage. Rather than using g in x, y directions (as Canny edge detection does), g is taken only in the direction of the solar azimuth angle. A basic threshold and no edge thinning generally leaves a pair of large segments of edge pixels around the crater's rim. Intuitively, more edge pixels around a circle should result in more accumulator votes for that circle in the HT. It is not possible to exploit this method using the OpenCV HT implementation, due to the nature of the 21HT variation. The approximation of the crater's edge normal requires the gradient to be taken in x, y directions, which leaves a thinner set of edge pixels. For these reasons, we have begun implementing a general version of the HT, which does not require the edge normals, and votes in a circle for each edge pixel. This method is expected to exploit this increase in edge pixels around the crater, allowing for increased TPR without as much increase in FPR as currently experienced.

Currently the convolution kernels used for the sun direction gradient edge detection are only representative of north, south, east, west, and their combinatorial counterparts (e.g. north-east, south-east etc.). It will be possible to improve on this discretization once the metadata of HiRISE images are considered, since the sun direction can be calculated more precisely. This also makes it possible for the process to be automated.

The progress so far leaves the detection phase as completely automated. However, as of yet there has been no optimizations made or attempts to parallelize the algorithms or workflow. Due to the volume of data we anticipate processing, access to a parallel supercomputing system will provide the opportunity to process more of it concurrently, and make processing of a large scale area practical. In addition, the algorithms offer themselves to parallelization. For example, the general HT may vote in the accumulator space at different radii in parallel, since each radius' voting space is independent. This too will be explored in future work.

The Square Kilometre Array (SKA) project coming to Western Australia has accelerated the development of supercomputing facilities such as iVEC's Pawsey Centre, which will provide the opportunity for applying this project and its future work to the supercomputing resources required.

Acknowledgements

This work was funded by the iVEC 2013-2014 Summer Internship program.

5. REFERENCES

[1] R. E. Arvidson, J. Boyce, C. Chapman, M. Cintala, M. Fulchignoni, H. Moore, G. Neukum, P. Schultz, L. Soderblom, R. Strom, A. Woronow, and R Young, "Standard techniques for presentation and analysis of crater size-frequency data," *Icarus*, vol. 37, pp. 467–474, February 1979.

[2] S. J. Robbins and B. M. Hynek, "A new global database of mars impact craters 1 km: 1. database creation, properties, and parameters," *Journal Of Geophysical Research*, vol. 117, May 2012.

[3] S. J. Robbins, B. M. Hynek, R. J. Lillis, and W. F. Bottke, "Large impact crater histories of mars: The effect of different model crater age techniques," *Icarus*, vol. 225, pp. 173–184, July 2013.

[4] A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansenm, et al., "Mars reconnaissance orbiter's high resolution imaging science experiment (hirise).," *Journal of Geophysical Research: Planets*, vol. 12, 2007.

[5] R. Jaumann, G. Neukum, T. Behnke, T. C. Duxbury, K. Eichentopf, J. Flohrer, S. v. Gasselt, et al., "The high-resolution stereo camera (hrsc) experiment on mars express: Instrument aspects and experiment conduct from interplanetary cruise through the nominal mission," *Planetary and Space Science*, vol. 55, pp. 928–952, 2007.

[6] E.R. Urbach and T.F. Stepinski, "Automatic detection of sub-km craters in high resolution planetary images," *Planetary and Space Science*, vol. 57, pp. 880–887, 2009.

[7] G.G. Michael, "Coordinate registration by automated crater recognition," *Planetary and Space Science*, vol. 51, pp. 563–568, 2003.

[8] Y. Sawabe, T. Matsunaga, and S. Rokugawa, "Automated detection and classification of lunar craters using multiple approaches," *Advances in Space Research*, vol. 37, pp. 21–27, 2006.

[9] L. Bandeira, W. Ding, and T. F. Stepinski, "Detection of sub-kilometer craters in high resolution planetary images using shape and texture features," *Advances in Space Research*, vol. 49, pp. 64–74, 2012.

[10] D.H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111–122, 1980.

[11] R.O. Duda and P.E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, 1972.

[12] G. Bradski and A. Kaehler, *Learning OpenCV*, O'Reilly Media, Inc., 2008.

[13] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *18th International Conference on Pattern Recognition*, August 2006, vol. 3, pp. 850–855.

[14] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and Vision Computing*, vol. 8, pp. 71–77, February 1990.

[15] G. Neukum, R. Jaumann, the HRSC Co-Investigator, and Experiment Team, "Hrsc: The high resolution stereo camera of mars express, in mars express: The scientific payload," *ESA Special Publication SP-1240*, pp. 17–35, 2004.