Delineation of Martian Craters Based on Edge Maps and Dynamic Programming

Jorge S. Marques[1] and Pedro Pina[2(✉)]

[1] ISR, Instituto Superior Técnico, Av. Rovisco Pais, 1049–001 Lisboa, Portugal
[2] CERENA, Instituto Superior Técnico, Av. Rovisco Pais, 1049–001 Lisboa, Portugal
jsm@isr.ist.utl.pt, ppina@tecnico.ulisboa.pt

Abstract. The delineation of impact craters is performed with a novel algorithm working in polar coordinates. The intensity transitions are determined along radial lines intersecting the center of the crater (Edge Map) being the optimal path, which corresponds to the minimization of an energy functional, computed by Dynamic Programming. The approach is tested on 8 HiRISE scenes on Mars, achieving a performance of 95 % of correct delineations.

Keywords: Crater rim · Edge map · Dynamic Programming · Mars

1 Introduction

The detection of impact craters on remotely sensed images from planetary surfaces is being done with an increasing number of automated approaches. A consistent evolution is observed in the last decade [1–9] with significant improvements that permit their use in the creation of crater catalogues [10–12]. Nevertheless, all these detections are represented in a simplified manner: each crater is described by a dimension (average diameter) and a location (coordinates of its centre), that is, by a perfect circular shape. The assimitries and irregularities of contours are thus not taken into account. Although these features are not fundamental for establishing surface chronologies [13], their availability at large scale is crucial to a better understanding of the resurfacing history and of the past climates on Mars [14]. The automated delineation of impact craters has only been done so far on two approaches: one based on a judicious sequence to find and link the crater edges in polar coordinates [15], the other based on the watershed transform and other mathematical morphology operators [16]. The initial results achieved a very good degree of success, but faced some difficulties when the datasets were enlarged, being not able to estimate a contour in a large amount of the samples and being too sensitive to local textural variations. Since there was an evident degradation of the performance in the most difficult examples, there still exists enough space for improvements. Therefore, we propose a novel algorithm to overcome those difficulties which is built into two main steps: edge enhancement in polar coordinates and crater delineation.

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part I, LNCS 8814, pp. 433–440, 2014.
DOI: 10.1007/978-3-319-11758-4_47

2 Algorithm

2.1 Overview

We assume that, to estimate the crater boundary contour, we know the location and radius of each crater. Even though we know this information in advance, crater delineation is a chalenging task since crater images often present low contrast between the crater rim and surrounding terrains making the detection of the rim very subtle.

The first step of the algorithm relies on intensity variation and tries to detect the intensity changes associated with the crater rim, while the second step tries to link the edges using geometric information.

Unfortunately, simple edge detection and linking approaches fail in this kind of images. Edge detection algorithms provide unreliable edges most of them associated to the terrain irregularities. To circumvent this difficulty, this paper defines a continuous edge map, $e(\mathbf{x}) \in [0, 1]$, which measures the amount of directional intensity variation in the vicinity of each point $\mathbf{x}$. A value $e(\mathbf{x}) = 0$ is assigned to a pixel $\mathbf{x}$ if there is strong intensity variation in the vicinity of $\mathbf{x}$ in a direction orthogonal to the crater contour. On the contrary, a value $e(\mathbf{x}) = 1$ is assigned if the image is constant in such direction. In the second step, we compute a closed contour, $\mathbf{x}(s)$, that minimizes an energy functional

$$E = \int e(\mathbf{x}(s))ds + E_{int}(\mathbf{x}) \ , \tag{1}$$

similar to the one used in the snake algorithm [17,18]; $E_{int}(\mathbf{x})$ denotes the internal energy which measures deviations of the crater contour, $\mathbf{x}(s)$, with respect to a circle and s denotes the arc length parameter of the curve.

It should be stressed that both operations become simpler and more effective if the image is converted from Cartesian to polar coordinates. This conversion is performed according to the procedure presented in [15].

2.2 Edge Map

We wish to define an edge map in polar coordinates $e(r, \theta)$. This map should assign a low value to points which are likely to be edges and high values to points which are not. We will assume that edges are associated to intensity transitions along radial lines intersecting the crater center, $\mathbf{c}$ (θ constant).

The radial gradient is defined as

$$g(r, \theta) = |P(r, \theta) * h(r)| \ , \tag{2}$$

where $|.|$ is the absolute value, $P(r, \theta)$ is the input image in polar coordinates (r, θ), $*$ denotes the convolution operation along the columns of P and $h(r)$ is the impulse response of a highpass filter, defined by $h(r) = -u(r - T) + 2u(r) - u(r + T)$ where $u(r)$ is the unit step function. This convolution can be computed extremely fast if we compute the integral image along the columns of P [20].

After computing the gradient, the edge map is obtained using the logistic function

$$\epsilon(r,\theta) = \frac{2}{1+e^{sg(r,\theta)}} , \tag{3}$$

which is often used to map the gradient intensity $g \in [0, +\infty[$ into an edge confidence $\epsilon \in [0, 1[$; s is a scale parameter. Since we have a good estimation of the radius of the crater, R, we will restrict r to an interval $[r_{min}, r_{max}]$ centered on R.

Fig. 1 shows the conversion from Cartesian to polar coordinates, assuming that $r_{min} = 0.8R, r_{max} = 1.2R$, and the corresponding edge map (right). The first and last rows of the edge map are padded with high intensity values since the highpass filtering results are unreliable.

Fig. 1. Image transformation: original image and sampling points (left), polar image (centre) and edge map (right)

2.3 Crater Delineation

The second step concerns crater delineation. We will assume that the edge map, ϵ, has M lines and N columns. The crater boundary is characterized by a sequence of row indices $\mathbf{r} = (r_1, r_2, \ldots, r_N)$ such that $r_t \in \{1, \ldots, M\}$. These indices represent the crater radius for each direction. If the crater boundary was a circle centered at $\mathbf{c}$, then the index sequence would be constant. In practice, the radius r_t changes slowly and must obey the boundary condition $r_1 = r_N = k$ (k unknown), since it represents a closed contour. Fig. 2 shows the edge map and the estimated countour in polar and Cartesian coordinates.

The contour sequence, $\mathbf{r}$, is chosen to minimize an energy functional

$$E(\mathbf{r}) = \epsilon(1, r_1 = k) + \sum_{p=2}^{N} \epsilon(p, r_p) + c(r_{p-1}, r_p) , \tag{4}$$

where $\epsilon(p, r_p)$ is the edge map and $c(r_{p-1}, r_p)$ denotes the cost associated to the transition from r_{p-1} to r_p. For the moment, we assume that r_1 is known ($r_1 = k$). In addition, we also assume that $|r_p - r_{p-1}| \leq 1$ to enforce smooth transitions and the transition cost is defined by

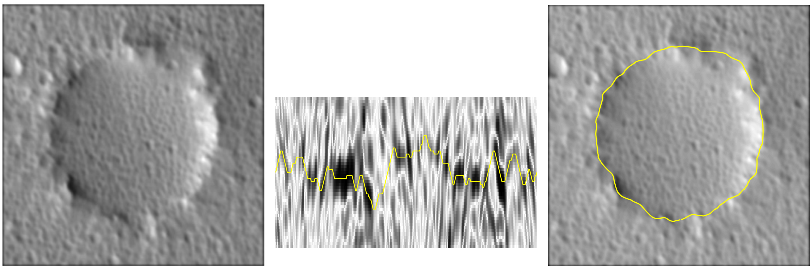

Fig. 2. Contour delineation: original image (left), edgemap and optimal contour (centre) and transformed contour (right)

$$c(r_{p-1}, r_p) = \begin{cases} 0 & \text{if } |r_p - r_{p-1}| = 0 \\ \alpha & \text{if } |r_p - r_{p-1}| = 1 \\ +\infty & \text{otherwise} \end{cases} . \quad (5)$$

The minimization of $E(r)$ under the constraint $r_1 = r_N = k$ can be solved by Dynamic Programming [21,22]. Dynamic Programming minimizes $E(r)$ in two steps. The first step computes the optimal costs to go from column 1 and line k to column t and line j, $E_t(j)$,

$$E_t(j) = \min_{r_2,\ldots,r_t : r_t = j} \left[\epsilon(1, r_1 = k) + \sum_{p=2}^{t} \epsilon(p, r_p) + c(r_{p-1}, r_p) \right] . \quad (6)$$

The optimal costs are computed by a forward recursion

$$E_t(j) = \epsilon(t, j) + \min_i \left[E_{t-1}(i) + c(i, j) \right] . \quad (7)$$

Since we want to retrieve the optimal path, it is important to store which value of i minimizes $[E_{t-1}(i) + c(i, j)]$ in (7). This information can be stored using a set of a pointers

$$\psi_t(j) = \arg\min_i \left[E_{t-1}(i) + c(i, j) \right] . \quad (8)$$

After computing the optimal costs $E_t(j), t = 1, \ldots, N, j = 1, \ldots, M$, we know what is the minimum energy associated to an optimal path $r_1^*, \ldots, r_N^*$ ending in $r_N^* = k$. The optimal path $\mathbf{r}^* = (r_1^*, r_2^*, \ldots, r_N^*)$ such that $r_N^* = k$, can be obtained by backtracking

$$r_{t-1}^* = \psi_t(r_t^*) \qquad t = N, \ldots, 2 . \quad (9)$$

The Dynamic Programming algorithm under the restriction $r_1 = r_N = k$ is summarized in Table 1. It provides the optimal path assuming that we know the boundary conditions k. Since the optimal k is unknown we repeat this procedure for all possible values of $k \in \{1, \ldots, M\}$ and choose the one which minimizes the energy.

Table 1. Dynamic Programming algorithm with boundary conditions $r_1 = r_N = k$

Forward recursion: computation of the optimal energies

$$E_1(j) = \begin{cases} \epsilon(1,k) \text{ if } j = k \\ +\infty \quad \text{otherwise} \end{cases}$$

$$E_t(j) = \epsilon(t,j) + \min_i \left[E_{t-1}(i) + c(i,j)\right], \qquad t = 2, \ldots, N$$

$$\psi_t(j) = \arg\min_i \left[E_{t-1}(i) + c(i,j)\right], \qquad t = 2, \ldots, N$$

Backward recursion: computation of the optimal contour

$$r_N^* = k$$

$$r_{t-1}^* = \psi_t(r_t^*) \qquad t = N, \ldots, 2 .$$

3 Experimental Results

We tested the algorithm on the highest resolution images presently available from the surface of Mars, that is, those captured by the HiRISE camera onboard the Mars Reconnaisance Orbiter in the two commonly provided resolutions, 0.25 and 0.50 m/pixel, in a map projected product. Thus, we selected regions in both hemispheres, with noticeable differences in the amount of craters, also exhibiting a wide variety of erosions rates, from pristine craters (with sharp rims) to degraded structures (with irregular, faint or missing parts of the rim), and also some examples with craters hardly noticeable. The testing datasets are constituted by 8 HiRISE images and a total of 805 craters depicted from them. The following parameters were heuristically chosen: $N = 61$, $M = 360$, $T = 6$ and $\alpha = 0.02$.

We evaluate the performance of the algorithm through the comparison of the delineated contour with a manually created contour (ground-truth contour) for each and every crater of the dataset. Each crater was manually delineated, also estimating a contour in regions where the crater rim was absent, that is, creating always one single closed contour for each impact structure. The distortion between those pairs of contours was measured by the percentage of correct points (*cp*), small errors (*se*) and gross errors (*ge*), as defined in [15,16].

Each crater of the 8 images was individually analysed and a closed contour estimated by the current algorithm ('Dynamic Programming') and by one of the previous approaches ('Morphologic') [16]. In many pratical applications, like in this crater delineation problem, small errors are acceptable, so we focus mainly our attention on gross errors (those whose distance between contours is superior to 0.05 of the crater diameter).

The average performances obtained by both methods are shown in Table 2. The Dynamic Programming algorithm performs very well and leads to an overall error of only 5% of incorrect delineations. In comparison, the 'Morphologic' algorithm obtained a lower performance with an overall error of 10.5%.

Table 2. Average performances (%) of automated crater delineation (*cp*-correct points, *ge*-gross errors, *se*-small errors)

Dataset	Craters	Dyn. Prog.			Morphologic		
	(#)	*cp*	*se*	*ge*	*cp*	*se*	*ge*
8 images	805	60.1	34.9	5.0	45.8	43.7	10.5

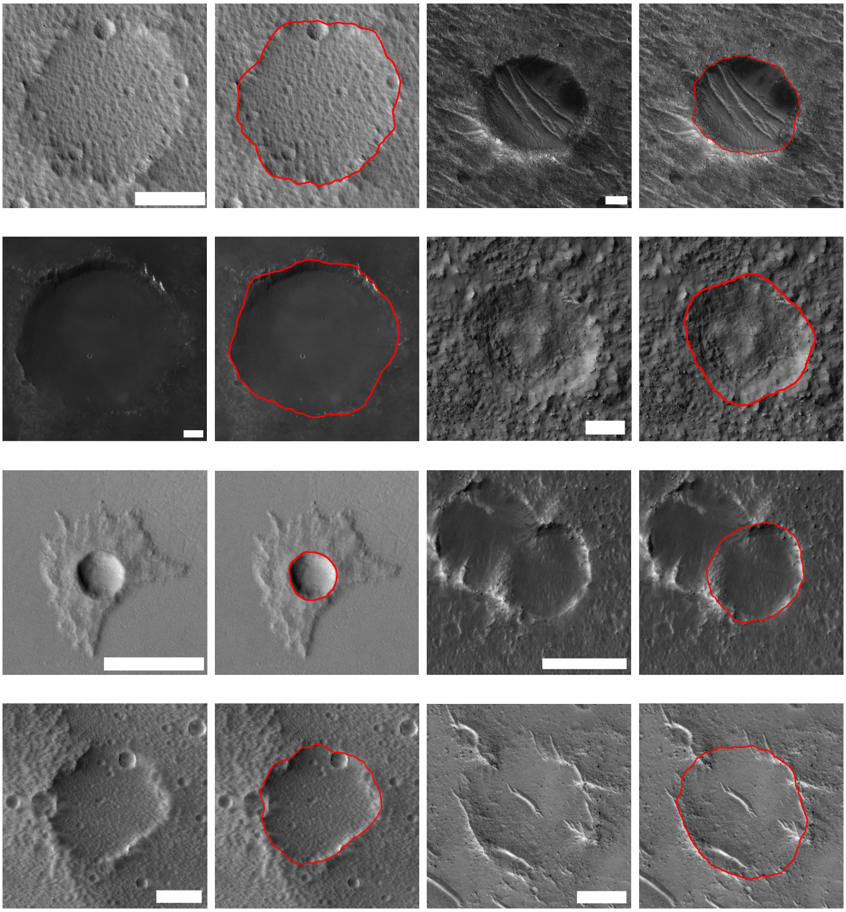

Fig. 3. Successful crater delineation examples (the white scale bars correspond to 50m) [image credits: NASA/JPL/University of Arizona]

The images of positive and negative examples, provided respectively in Fig. 3 and Fig. 4, are also a comprehensive illustration of the performances achieved by the algorithm. The proposed algorithm manages to delineate very difficult examples with high texture and missing rims. The number of failures is small and usually associated to strong geometric deformations of the crater rim in which the circular shape can no longer be assumed. These cases are very rare.

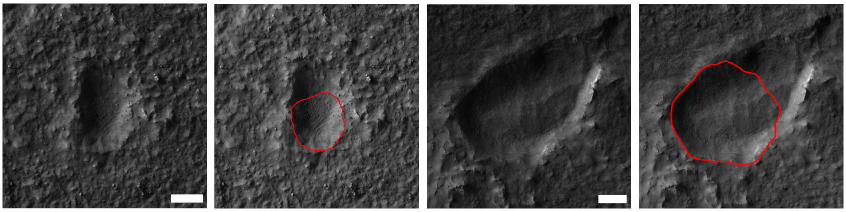

Fig. 4. Incorrect crater delineation examples (the white scale bars correspond to 50m) [image credits: NASA/JPL/University of Arizona]

4 Conclusions

In this study we presented a novel algorithm to delineate the boundary of impact craters previously detected on the surface of Mars. The proposed algorithm achieves very high performances (average error of 5%) in a diversified dataset of 805 craters and clearly outperformed the best available algorithm.

We consider that the exploitation of the *a priori* knowledge about the problem, like the circular geometry and image intensity patterns of the craters, and its integration into an optimization procedure, are the key features for the robustness and high success achieved by this novel algorithm. In particular, the geometry of the craters permits to adequately define a region of interest around its rim and hugely constrain the space of search for edges of interest. Moreover, the improved detection of the crater edges synthesized on the Edge Map and the detection of the optimal path (the crater contour) with the Dynamic Programming algorithm are also strong points. Finally, converting and processing the crater images into polar coordinates also greatly simplifies the processing and turns it into an additional advantage of the approach.

Acknowledgments. This work was developed in the frame of the projects PTDC/CTE-SPA/110909/2009 and PEst-OE/EEI/LA0009/2013, both funded by FCT-Fundação para a Ciência e a Tecnologia, Portugal.

References

1. Michael, G.: Coordinate registration by automated crater recognition. Planetary and Space Science **51**, 563–568 (2003)
2. Bue, B.D., Stepinski, T.F.: Machine detection of Martian impact craters from digital topography data. IEEE Trans. Geoscience & Remote Sensing **45**, 265–274 (2007)
3. Bandeira, L.P.C., Saraiva, J., Pina, P.: Development of a methodology for automated crater detection on planetary images. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4477, pp. 193–200. Springer, Heidelberg (2007)
4. Bandeira, L., Saraiva, J., Pina, P.: Impact crater recognition on Mars based on a probability volume created by template matching. IEEE Trans. Geoscience & Remote Sensing **45**, 4008–4015 (2007)

5. Martins, R., Pina, P., Marques, J.S., Silveira, M.: Crater detection by a boosting approach. IEEE Geoscience and Remote Sensing Letters **6**, 127–131 (2009)
6. Urbach, E.R., Stepinski, T.F.: Automatic detection of sub-km craters in high resolution planetary images. Planetary and Space Science **57**, 880–887 (2009)
7. Bandeira, L., Ding, W., Stepinski, T.F.: Detection of sub-kilometer craters in high resolution planetary images using shape and texture features. Advances in Space Research **49**, 64–74 (2012)
8. Vijayan, S., Vani, K., Sanjeevi, S.: Crater detection, classification and contextual information extraction in lunar images using a novel algorithm. Icarus **226**, 798–815 (2013)
9. Jin, S., Zhang, T.: Automatic detection of impact craters on Mars using a modified adaboosting method. Planetary and Space Science (in press, 2014)
10. Salamunićcar, G., Lončarić, S., Pina, P., Bandeira, L., Saraiva, J.: MA130301GT catalogue of Martian impact craters and advanced evaluation of crater detection algorithms using diverse topography and image datasets. Planetary and Space Science **59**, 111–131 (2011)
11. Salamunićcar, G., Lončarić, S., Mazarico, E.: LU60645GT and MA132843GT catalogues of Lunar and Martian impact craters developed using a Crater Shape-based interpolation crater detection algorithm for topography data. Planetary and Space Science **60**, 236–247 (2012)
12. Salamunićcar, G., Lončarić, S., Pina, P., Bandeira, L., Saraiva, J.: Integrated method for crater detection from topography and optical images and the new PH9224GT catalogue of Phobos impact craters. Advances in Space Research **53**, 1798–1809 (2014)
13. Hartmann, W.K., Neukum, G.: Cratering chronology and the evolution of Mars. Space Science Reviews **96**, 165–194 (2001)
14. Boyce, J.M., Garbeil, H.: Geometric relationships of pristine Martian complex impact craters, and their implications to Mars geologic history. Geophysical Research Letters **34**, L16201 (2007)
15. Marques, J.S., Pina, P.: An algorithm for the delineation of craters in very high resolution images of mars surface. In: Sanches, J.M., Micó, L., Cardoso, J.S. (eds.) IbPRIA 2013. LNCS, vol. 7887, pp. 213–220. Springer, Heidelberg (2013)
16. Pina, P., Marques, J.S.: Delineation of impact craters by a mathematical morphology based approach. In: Kamel, M., Campilho, A. (eds.) ICIAR 2013. LNCS, vol. 7950, pp. 717–725. Springer, Heidelberg (2013)
17. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**, 321–331 (1988)
18. Blake, A., Isard, M.: Active Contours. Springer (1998)
19. Szeliski, R.: Computer vision: algorithms and applications. Springer (2011)
20. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision (2002)
21. Bellman, R.E.: The Bellman Continuum. A collection of the works of Richard E. Bellman, Robert S. Roth (ed.) World Scientific (1986)
22. Bertsekas, D.: Dynamic Programming and optimal control. Athena Scientific (2005)