Detection and Characterization of Stone Circles in Permafrost Zones in Antarctica

Francisco Tomás Teotónio Ferreira Soares Pereira
francisco.s.pereira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2019

Abstract

In polar regions, seasonal freezing and thawing of the soils lead to the creation of stone circles, a natural type of patterned ground whose study provides information about the past climatic conditions that motivated its genesis. These patterns, that occur in clusters of thousands of circular elements, can be advantageously characterized if automated methods are used. The automated methods for the detection and delineation of stone circles are addressed for the first time in this thesis and are based on Template Matching, the Sliding Band Filter and Dynamic Programming. The detection methods are tested in a dataset of 20 Digital Elevation Models (DEM) of centimetric resolution obtained after surveys with Unmanned Aerial Vehicles (UAV) in Barton Peninsula in King George Island (Antarctica), while a set of 24 individual stone circles from different sites were selected to test the delineation methods. For the detection problem, the best compromise between performance and complexity was achieved by the Dynamic Programming technique with an average F-score of 84.3%. The Dynamic Programming method reached the smallest average percentage of gross errors in contour delineation (7.1%), a reduction of approximately 10% compared to the Sliding Band Filter's performance. Despite the variability in the shape and size of the unique structures analyzed and given the high-performance achieved, one can consider the proposed problem was appropriately addressed.
Keywords: Antarctica, Stone circles, Elevation, Circle detection, Circle delineation.

1. Introduction

The exceptional circular geometry of a natural type of patterned ground intrigues researchers in polar regions. Scattered across the periglacial landscape, these stone circles are characterized by a unique sorting phenomenon where the soil material is arranged according to its size. Scientists have relied on long studies and field campaigns to understand the physical properties and processes involved in the cyclic freezing and thawing of the soils. Understanding the stone circles' genetic process can provide information about the past climate, as they are considered potential paleoclimatic indicators. However, due to the lack of systematic quantitative measures of temperature and motion of the soils, the dynamics of the genetic process of the stones are not being properly characterized [Uxa et al., 2017]. Despite the development of microprocessor-based data acquisition systems to monitor such properties, the number of circles and large areas where they occur requires the development of automated detection methods. This work concerns the detection and delineation of stone circles and the methods proposed analyze elevation maps built from image data acquired by UAVs.

2. Background

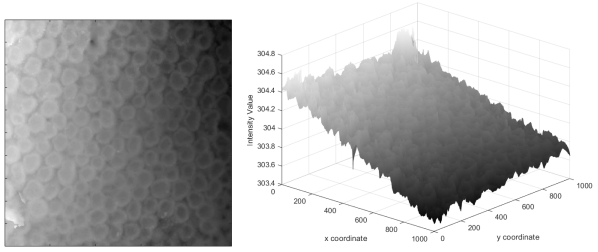
The automated detection and delineation of stone circles is an unexplored problem yet, the segmentation of circular shapes in images has been widely explored, whether in the detection of cells in microscopy imaging [Quelhas et al., 2010], the components of the eye for biometric identification [Hansen and Ji, 2009] or in the study of impact craters in planetary images [Marques and Pina, 2015]. These include diverse techniques such as the ones based on Hough transform [Ballard, 1981], template matching [Bandeira et al., 2007] and deformable templates [Yuille et al., 1992] or the recent Deep Learning methods [Falk et al., 2019].

3. Stone circle detection and delineation

This chapter describes the methods used to detect and delineate stone circles in digital elevation maps. It also includes a description of a preprocessing stage to enhance the stone circles.

3.1. Pre-Processing

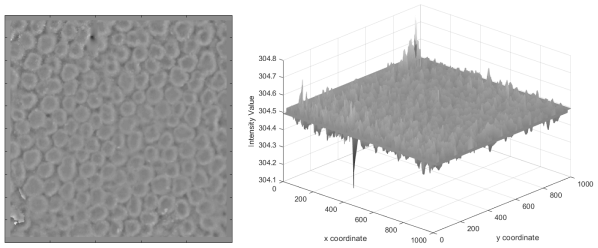
The input image, I, is a digital elevation image where brighter pixels, with higher intensity values, correspond to regions with a higher elevation while darker, low-valued pixels, correspond to low elevation regions. Figure 1(a) shows the input image, I, and Figure 1(b) the corresponding 3D model, where the vertical axis expresses the elevation measured from the stones' surface. The natural slope visible



(a) Input Image, I. (b) 3D model of I.

Figure 1: Digital elevation image, I, and the corresponding 3D surface.

in Figure1(b) was removed by applying a high-pass filter to the input image. First the slope was captured as the output of a low-pass filter, using the convolution operation $I * H$, where the filter mask H is an averaging filter. Then the low-pass filter output was subtracted from I, resulting in the filtered image, I_f, shown in Figure 2(a). Figure 2(b) shows the corresponding surface image where the stone circles to detect are displayed across an approximate horizontal plane. The filtering operation preserves the shape of the stone circles and removes the brightness difference across the image.



(a) Filtered Image, I_f. (b) 3D model of I_f.

Figure 2: High-pass filter output: slope removal.

For the sake of simplicity, the filtered image will be denoted as I and used as the input image of the methods.

3.2. Template Matching

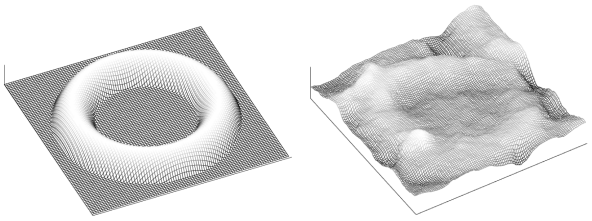
The Template Matching (TM) algorithm searches across the input image for regions that most resemble the template image T. The similarity between the image and the template is expressed by a score image Y, obtained by the cross-correlation operation given by

$$Y(x,y) = \sum_{i=-N}^{N} \sum_{j=-N}^{N} T(i,j) I(x+i, y+j). \quad (1)$$

The shape of a stone circle can be modelled by a half torus and is characterized by two parameters: the radius R i.e an integer representing the distance from the torus center to the highest point in the ring, and the width δR, which defines the concave shape of the template ($\delta \in [0,1]$). The intensity level of pixel $X = [x_1, x_2]^T$ of the template T depends on the Euclidean distance $d = \sqrt{(x_{c_1} - x_1)^2 + (x_{c_2} - x_2)^2}$ between X and the torus' geometrical center, $X_c = [x_{c_1}, x_{c_2}]^T$, according to

$$T(R,\delta) = \begin{cases} (\delta R)^2 - (d-R)^2, & d > \mid (1-\delta)R \mid \vee \\ & d < \mid (1+\delta)R \mid \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

In order to mitigate the influence of brightness variations across the image, the mean value of the template was subtracted from each pixel. Figure 3(a) shows the 3D normalized template and Figure 3(b) displays the elevation map of a real stone circle.



(a) Template's 3D representation. (b) 3D detail of a real stone circle.

Figure 3: Template and example of a stone circle: the real stone circle is a deformed version of the template.

Figure 4 shows the score image Y obtained for a template characterized by $R = 30$ and $\delta = 0.4$.

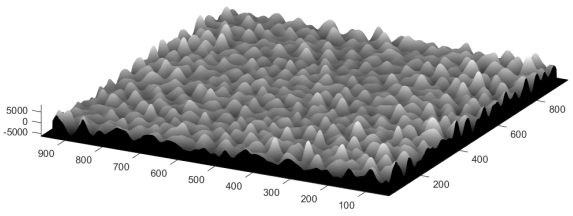


Figure 4: Score image obtained by correlation of I with the template T.

The bell-shaped peaks of Figure 4 correspond to the candidate center locations of the stone circles and their intensity reflects the respective confidence level: higher peaks are more likely to being stone centers than the lower amplitude ones.

In order to obtain the peaks of the score image, the non-maximum suppression algorithm is applied. From this set of local maxima, it is possible to select those with a sufficiently high confidence level, by performing a comparison between their score image value and a threshold Th, corresponding to a percentage of the maximum value of Y. The green pixels superimposed with the score image in Figure 5 are those whose score is greater than the threshold, represented by the red horizontal plane.

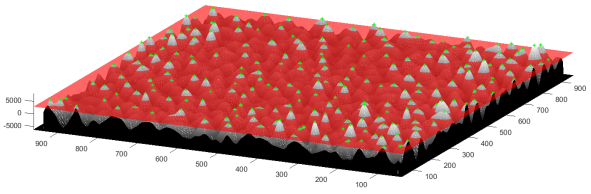


Figure 5: Thresholding output: the green marked pixels have a score value above the minimum required, represented by the red horizontal plane $z = Th \times \max\{Y\}$.

Although the thresholding operation eliminated many false candidates, there are still local maxima that are too close together to be the center of stone circles. The peak validation stage preserves the candidates whose score is the highest inside a circular region with a given radius, d_{min}. Any candidate with a lower score inside that region is discarded. Figure 6 shows the output of this spatial filter: the accepted candidates are represented as green markers while the red ones are the eliminated candidates, most of which are located in the intersection of three adjacent stone circles.

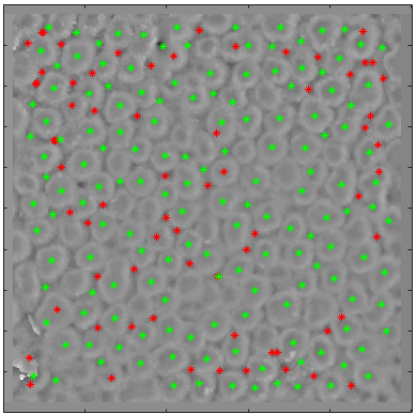


Figure 6: Detections (green) obtained with TM method.

The value of d_{min} should be greater than the average stone circle's radius, otherwise there could be accepted detections located near the stone's boundary. However, d_{min} must be smaller than twice the stone's radius, or adjacent detections would be discarded incorrectly. The detections obtained after the peak validation stage correspond to the output of the Template Matching algorithm.

3.3. Sliding Band Filter

The Sliding Band Filter (SBF) is a method developed for the detection of circular cells on microscopy images, through an analysis of the gradient vector's orientation [Quelhas et al., 2010]. It assumes the image gradient has a radial structure in the vicinity of a circle's center and measures the alignment between the gradient vector's direction and the direction of auxiliary radial lines: the center of the circle corresponds to the point where this alignment is maximized. By assigning each pixel a value according to the alignment, the score image Y is created.

3.3.1 Alignment score definition

In order to detect a circle's center, all image pixels $p = [x, y]^T$ are considered. Let us define N radial lines emerging from a given center point p, and let $\theta_i = 2\pi i/N$ be the direction of the i-th line, $i = 0, ..., N-1$. The alignment between the i-th line's direction and the image gradient direction computed at point

$$q = p + r[\cos(\theta_i), \sin(\theta_i)]^T \tag{3}$$

is given by

$$A_i(r) = \cos(\theta_i - \phi(r)), \tag{4}$$

where $\phi(r)$ denotes the direction of the gradient vector at point q , with $R_{min} < r \leq R_{max}$. Figure 7 displays a schematic representation of the alignment computation. The alignment $A_i(r)$ is computed for M points along direction i and the distance r from the evaluated point p is defined as $r = R_{min} + j\Delta r$, $j = 1, ..., M$, with the interval $\Delta r = (R_{max} - R_{min})/M$.

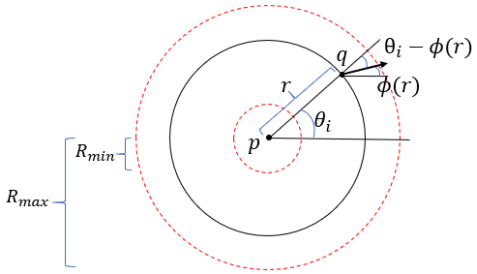
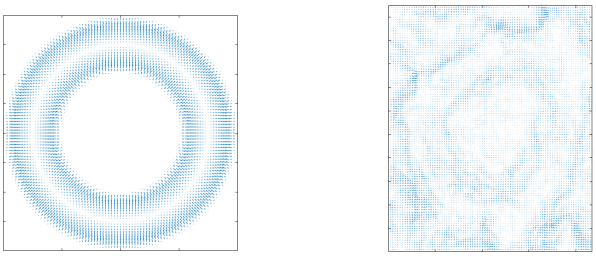


Figure 7: Alignment computation at point q, given the i-th line's orientation, θ_i, and the gradient direction, $\phi(r)$.

3.3.2 Score image computation

By computing the alignment score for M points along a given radial direction, the gradient noise gives rise to a noisy alignment vector $A_i(r)$. Ideally this vector would correspond to the template $T(r) = -u(r+\delta) + 2u(r) - u(r-\delta)$, where $u(r)$ denotes the unit step signal, as the gradient vector's direction points outwards and then inwards as r increases. This is true for a synthetic stone circle whose gradient field is shown in Figure 8(a). Figure 8(b) shows the noisy gradient field corresponding to a real stone circle.



(a) Synthetic stone circle.

(b) Actual stone circle.

Figure 8: Gradient field distribution: inwards and outwards radial field.

The stone circle's boundary along the direction i can be computed by a Matched Filtering operation i.e the convolution of the alignment vector $A_i(r)$ with an impulse response that is a reversed and time delayed version of the template $T(r)$. The maximum value of the filter's output is used to quantify the similarity between $A_i(r)$ and $T(r)$,

$$A_i = \max_{r_0}\Big\{\sum_r T(r_0 - r)A_i(r)\Big\}, \tag{5}$$

where r_0 is the sample corresponding to the output's highest peak. Since there are N radial directions, by computing an average of the alignments A_i, $i = 1, ..., N$ a global alignment score is obtained to rank the center point p. The SBF's score image value for the input pixel $p = [x, y]^T$ is then defined as

$$Y(x, y) = \frac{1}{N}\sum_{i=1}^{N} A_i. \tag{6}$$

In order to remove the noise of the score image, a smoothing operation is performed. By convolving Y with the Gaussian kernel $H(x, y) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{x^2+y^2}{2\sigma^2}}$, the sharp variations are eliminated locally yet, the score alignment information is preserved. This filtering process produces bell-shaped surfaces with brighter rounded peaks, corresponding to possible center candidates of the stone circles. The local maxima of the filtered score image are computed using the non-maximum suppression and only the resulting peaks whose score exceeds the threshold $Th \times \max\{Y\}$ are accepted as detections.

Similarly to the Template Matching's output, some detections obtained with the SBF are too close together to be considered stone circles. The spatial filter described in Section 3.2 eliminates many false detections, represented as red stars in the example of Figure 9. Most of the final detections (green) correspond to an accurate location of the stone circles.

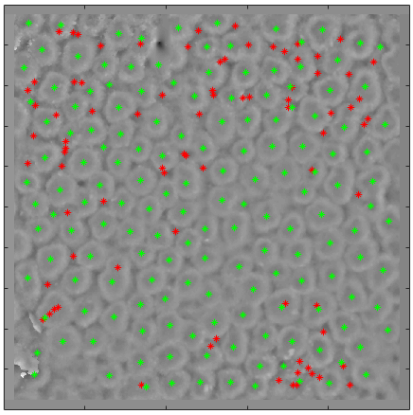


Figure 9: Detected peaks after spatial filtering (green marks). The red marks correspond to eliminated peaks.

3.4. Dynamic Programming

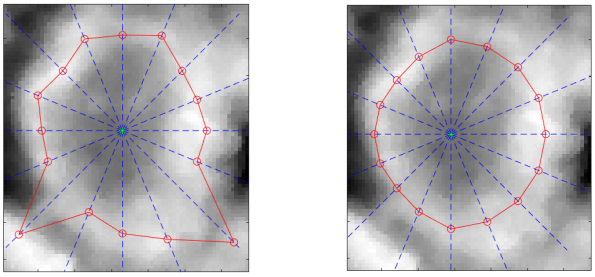
The method proposed in this section is based on a Dynamic Programming (DP) technique [Marques and Pina, 2015] and on the same criteria used by the SBF for detecting and delineating the stone circles. Both methods rely on the contour information to evaluate if a given pixel is a valid stone circle center but the main difference is in the additional smoothing criteria used in DP. As opposed to the SBF that estimates the contour independently for each direction, the DP-based method computes the contour globally, by enforcing the smoothness associated with the circular geometry of the stone circles.

Figure 10 shows the effect of the global constraint in both the detection and delineation of the circles: SBF is not able to avoid sharp variations hence the outliers detected (see Figure 10(a)) while DP forces the global dependency of the contour points (see Figure 10(b)).

3.4.1 Score image computation

In order to compute the score image value for each candidate p, let us define a matrix $V \in \mathbb{R}^{M\times N}$ whose rows and columns designate the coordinates of points in polar form. For instance, the entry $V(j, i)$ refers to the point

$$q = p + r_j[\cos\theta_i, \sin\theta_i]^T, \tag{7}$$



(a) (b)

Figure 10: Stone detection and contour extraction using SBF (a) and the DP (b) method, with $N = 16$ directions.

and stores the symmetrical of the filtered alignment value computed at point q,

$$s_i = -\sum_r T(r_0 - r)A_i(r), \quad V(j,i) = s_i(j). \quad (8)$$

The index i refers to the angle $\theta_i = \Delta\theta(i-1)$, with $i = 1, ..., N$ and the index j determines the radius $r_j = R_{min} + j\Delta r$, with $j = 1, ..., M$, where the angle and radius steps are defined by $\Delta\theta = 2\pi/N$ and $\Delta r = (R_{max} - R_{min})/M$, respectively.

The stone circle's boundary is defined by a sequence of row indices $r = [r_1, ..., r_N], r_t \in \{1, ..., M\}$, that represent the stone's radius for the first up to the N-th direction. The contour r is chosen to minimize the energy functional [Marques and Pina, 2015]

$$E(r) = V(r_1 = k, 1) + \sum_{t=2}^{N} V(r_t, t) + c(r_{t-1}, r_t), \quad (9)$$

where $c(r_{t-1}, r_t)$ defines the cost associated to the displacement between radius r_{t-1} and r_t,

$$c(r_{t-1}, r_t) = \begin{cases} m \mid r_t - r_{t-1} \mid, & \text{if } \mid r_t - r_{t-1} \mid \leq \beta \\ \infty, & \text{otherwise} \end{cases}. \quad (10)$$

The parameters m and β control the smoothness of the transition in the radius obtained for consecutive directions: the greater the slope m, the more restrictive are the allowed jumps since the cost increases faster and by choosing a smaller β, bigger deviations are avoided.

Although r is a closed contour, given the definition of θ_i, the first and last contour points do not coincide. By adding a column to V whose values are those computed for the first direction ($\theta_1 = \theta_{N+1}$), this issue is fixed and the contour computation is subjected to $r_1 = r_{N+1}$.

The minimization of the contour's energy $E(r)$ implies the alignment for each r_t should be high - minimum $V(r_t, t)$ - and the deviations between adjacent radius should be small, as the cost is a positive quantity that increases the energy. The optimal path leading to the optimal energy value is obtained in two stages: the forward recursion and the backward recovery step.

The forward recursion computes the optimal energy to reach the element in row j and column t, from the first path element located in row k and column 1, according to

$$E_t(j) = \min_{r_2,...,r_t; r_t = j} \Big[V(r_1 = k, 1) + \sum_{p=2}^{t} V(r_p, p) + c(r_{p-1}, r_p) \Big], \quad (11)$$

which can be written recursively as

$$E_t(j) = V(j,t) + \min_i \Big[E_{t-1}(i) + c(i,j) \Big]. \quad (12)$$

It is also important to store the sequence of indexes i representing the state that minimized $[E_{t-1}(i) + c(i,j)]$. This set of pointers is obtained from

$$\phi_t(j) = \arg\min[E_{t-1}(i) + c(i,j)]. \quad (13)$$

The optimal path $r^* = [r_1^*, r_2^*, ..., r_{N+1}^*]$ is obtained by the backward recovery step: the last contour point is given by the constraint $r_{N+1}^* = k$ and the remaining ones are computed according to

$$r_{t-1}^* = \phi_t(r_t^*), \quad t = N+1, ..., 2. \quad (14)$$

The unknown initial condition $r_1 = r_{N+1} = k$ can be computed assuming that, regardless of the initial radius k, halfway of the optimal path the radius value is always the same. The procedure used to determine the boundary condition is similar to the one performed in [Santiago et al., 2018].

Figure 11 (left) shows an example of the contour r in yellow and a horizontal curve corresponding to a circle with radius $R = 30$, superimposed with matrix V. A contour point in Cartesian coordinates, X_c, can be obtained from the radius r_t and the stone circle's center, p, by the expression $X_c = p + \rho[\cos\theta, \sin\theta]^T$, with $\rho = R_{min} + r_t\Delta r$ and $\theta = \Delta\theta(t-1)$. Figure 11 (right) shows the corresponding contour in Cartesian coordinates superimposed with the stone's image.

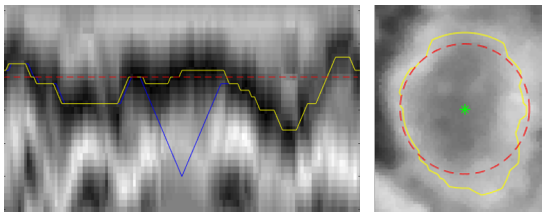


Figure 11: Optimal contour r (yellow) in polar (left) and Cartesian coordinates (right).

The contour shown in Figure 11 is composed of $N = 128$ points and the parameters used ($m = 0.3$,

$\beta = 1$) restrict the variations between consecutive radius, as visible through the constant segments of r and the absence of sharp transitions.

By computing the optimal contour centered at each image pixel, its energy can be used to rank its likelihood of being a stone's center: a candidate whose contour energy is small is more likely to be a valid stone circle's center as the alignment is higher according to (9). Therefore, given the optimal contour $r^* = [r_1^*, ..., r_{N+1}^*]$ for the center pixel $p = [x, y]^T$, the DP's score image value is $Y(x, y) = E(r^*)/(N + 1)$.

An auxiliary version of Y given by $Y_o = \max\{Y\} - Y$ is created such that the peaks of Y_o correspond to the most likely center locations i.e the low values of Y. After obtaining Y_o, the stone's locations are computed by the non-maximum suppression, thresholding and peak validation methods. Figure 12 shows the accepted detections (green) and the ones discarded by the peak validation method (red).

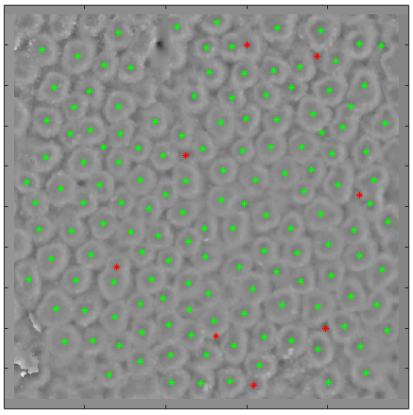


Figure 12: Detections (green) obtained with DP method.

3.5. Contour Delineation

In addition to an accurate location of the stone circles, the geologists also wish to determine their boundary to study the circles' dimensions and shapes. The SBF and DP methods are also used to produce estimates of the contour of the circles.

3.5.1 Sliding Band Filter

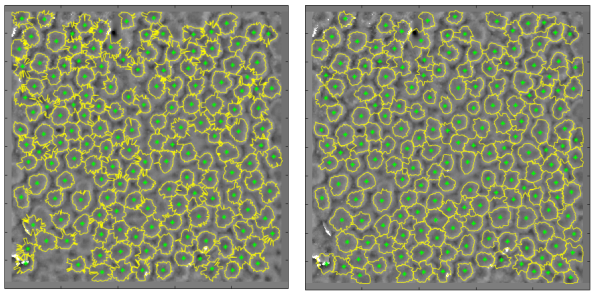
In order to determine the N radius values along each θ_i direction, $i = 1, ..., N$, that form the contour, the alignment is evaluated. By computing the sample $r_0^{'}(i)$ that maximizes the filtered alignment

$$r_0^{'}(i) = \arg\max_{r_0} \Big\{ \sum_r T(r_0 - r) A_i(r) \Big\}, \qquad (15)$$

for each direction, the circle's boundary is identified. However, a shift operation is required since $r_0^{'}(i)$ does not correspond to the transition from the negative to the positive pulse of $T(r)$ (see Section 3.3.2). The i-th contour point is then computed as $r_0^*(i) = r_0^{'}(i) - d$, where d is half the number of samples of $T(r)$. Figure 13(a) shows the contour estimation output for the SBF obtained with $N = 128$ directions. The contour points obtained are connected, creating a closed curve for each detected stone. The contours exhibit some noise as there are some contour points that can lie farther from the stone circle, and be instead in the proximity of a neighbor circle.

3.5.2 Dynamic Programming

The DP method presents a solution for the contour delineation problem by minimizing the contour's energy, a value that considers both the alignment score (4) and the deviation between consecutive radius values. The optimal contour r is given by the forward recursion (12) and backward recovery (14) equations. Figure 13(b) shows the contour obtained with $N = 128$ directions and cost parameters $m = 0.03$ and $\beta = 10$: the outliers are reduced, in comparison with the contour points obtained with the SBF method yet, given the small cost penalization there are still observed some oscillations.



(a) Output of SBF. (b) Output of DP.

Figure 13: Contour delineation with $N = 128$ directions.

4. Results

4.1. Evaluation metrics

The performance of the detection and contour extraction methods is defined by the similarity between their output and the ground-truth (GT) information. For each one of the 20 elevation maps used to test the detection methods, the GT corresponds to the center coordinates of the stone circles. For the contour delineation problem, the GT consists of the coordinates of the stones' boundary obtained for 24 stone circles. The complexity and uncertainty of the manual marking task impede testing the delineation methods in a larger set of circles.

4.1.1 Stone circle detection

In order to evaluate the quality of the stone's location, a one-to-one correspondence between the

ground-truth centers and the automated detections is implemented [Pina et al., 2019]. Since the matching criteria takes into account the displacement error between the GT points and the detections, the first step consists in building a matrix of distances from all the ground-truth centers to all the detected points. The distance that separates the closest ground-truth and detected elements is evaluated: if the localization error is smaller than λ, the elements form a valid pair and are considered a true positive (TP). The distance matrix is updated by removing the corresponding line and column of the elements paired. This process is repeated until there are no distance matrix entries smaller than λ. The ground-truth points and detections that were not matched correspond to the false negatives (FN) and false positives (FP), respectively. The value chosen for λ corresponds to the average stone's radius $R = 30$ because a larger tolerance would produce unreliable results as the classifier may be associating points that belong to different circles.

From the number of TP, FP and FN, the metrics Precision and Recall are computed to assess the algorithm's performance according to

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}. \tag{16}$$

Precision measures the percentage of the algorithms's detections correctly located while Recall indicates the percentage of the actual centers correctly identified. The balance between these metrics is quantified by the harmonic mean F-score

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{17}$$

4.1.2 Contour Delineation

The quality of the contour depends on the magnitude of the localization error between the ground-truth contour (B) and the detected contour (A) [Marques and Pina, 2013]. For each detected contour point, p, the displacement from the closest ground-truth point

$$d(p, B) = \min_{q \in B} \|p - q\| \tag{18}$$

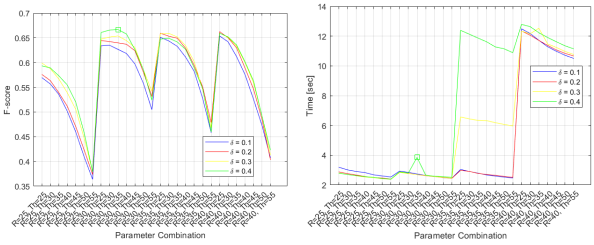
dictates its classification: if $d(p, B) = 0$, point p is a correct point (CP); if $0 < d(p, B) < \Delta$, the error in p is considered a small error (SE) and if $d(p, B) \geq \Delta$, p is considered a gross error (GE). The performance of the delineation task is entirely determined by the GE percentage as the displacement corresponding to a CP and SE is assumed to be irrelevant. The tolerance Δ is fixed at $0.15R$, with $R = 30$.

4.2. Experiments - Stone circle detection

Each detection method was executed for different parameter combinations, for each input image, and average values of the evaluation metrics were obtained. The optimal parameter configuration corresponds to the one that maximizes the average F-score for all images tested.

4.2.1 Template Matching

The performance of TM depends on the template's radius, R, its width, δR, and the threshold Th. The following values were considered for each parameter: $R \in \{25, 30, 35, 40\}$, $\delta \in \{0.1, 0.2, 0.3, 0.4\}$, $Th \in \{25, 30, 35, 40, 45, 50, 55\}$. Figure 14(a) shows the average F-score computed per parameter combination and Figure 14(b) shows the effect of the parameters on the average computational time of TM.



(a) (b)

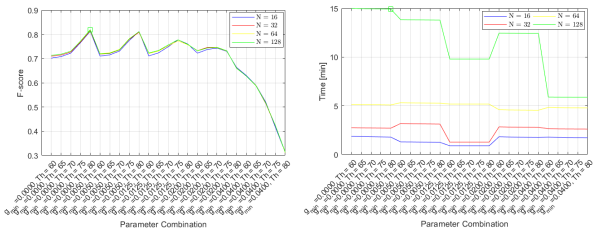
Figure 14: Template Matching: F-score (a) and Time (b) as a function of the parameters.

Figure 14(a) shows the F-score is reduced as Th increases, regardless of the values of R and δ. For $R = 25$ and $R = 30$, larger values of δ increase the method's performance while for larger R values, the F-score increases for intermediate values of δ. The maximum average F-score for TM (66.67%) is obtained with the parameters $R = 30$, $\delta = 0.4$ and $Th = 35\%$, although there are other combinations that achieved similar performances. The method's low complexity is confirmed by Figure 14(b): it takes no more than a few seconds to produce the detections. As R increases the average time also increases, given the larger amount of computations executed. It is interesting to note that for the same R, the time reduces as Th increases since there are less detections.

4.2.2 Sliding Band Filter

The SBF's performance is determined by the number of directions considered, N, the minimum gradient magnitude required g_{min} and the threshold Th. The method was executed for the following parameter values: $N \in \{16, 32, 64, 128\}$, $g_{min} \in \{0.0, 0.005, 0.0125, 0.02, 0.04\}$ and $Th \in \{60, 65, 70, 75, 80\}$.

Figure 15(a) illustrates the average F-score as a function of the parameters and the computation time evolution is shown in Figure 15(b). Figure



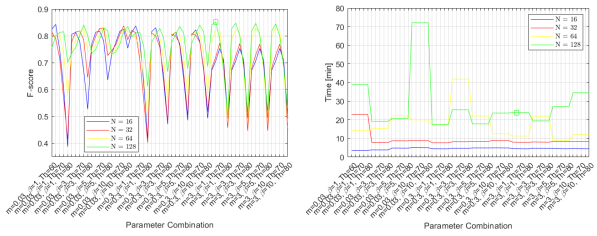
(a) (b)

Figure 15: Sliding Band Filter: F-score (a) and Time (b) as a function of the parameters.

15(a) shows the SBF achieves higher performance values compared to the TM, with some parameter configurations reaching F-scores above 80%. It can also be concluded that SBF is not sensitive to N as the 4 curves are similar. Moreover, the F-score is reduced as g_{min} increases because the required magnitude value is so excessive that the majority of pixel's orientation is ignored in the alignment computation. The maximum average F-score = 81.95% was achieved with the parameters $N = 128$, $g_{min} = 0$ and $Th = 80\%$ but, since combinations with a smaller N obtain a performance close to optimal, an analysis of Figure 15(b) is useful to understand if the reduction on N can be more efficient. The method analyzes each image in the order of minutes and for $N = \{16, 32, 64\}$ the time is approximately constant and independent of g_{min}. For $N = 128$, larger g_{min} values reduce the time given the less calculations performed.

4.2.3 Dynamic Programming

The performance of DP is influenced by N, Th and the transition cost parameters m and β. The algorithm was executed for all combinations with $m \in \{0.03, 0.3, 3\}$, $\beta \in \{1, 3, 5, 10\}$ and the same values for N and Th tested for SBF. Figures 16(a) and 16(b) show the effect of the parameters in the average F-score and computational time obtained with DP, respectively. Figure 16(a) shows peaks in the F-score for smaller values of Th, whichever m and β values. It is also observed that the oscillations in performance are less evident for $N = 128$ at least up to $m = 3$. Note the DP is not sensitive to changes in β for the bigger slope tested, $m = 3$. The maximum average F-score is 85.18% and is obtained with $N = 128, Th = 65\%$ and the most restrictive cost parameters $m = 3, \beta = 1$. As expected, Figure 16(b) shows the complexity increases with N. For $N = 128$ the method always takes above 10 min and up to more than 70 min while for $N = 16$ and



(a) (b)

Figure 16: Dynamic Programming: F-score (a) and Time (b) as a function of the parameters.

$N = 32$, the time is approximately constant for all configurations, with values below 10 min.

4.2.4 Detection methods comparison

Table 1 summarizes the mean and standard deviation (in brackets) of the evaluation metrics obtained for the optimal parameters per method, in the columns named TM, SBF_1 and DP_1.

Table 1: Optimal and sub-optimal performance measures for TM, SBF and DP methods.

Method	TM	SBF_1	DP_1	SBF_2	DP_2
F1 [%]	66.7 (8.9)	82.0 (6.6)	85.2 (6.6)	81.3 (6.9)	84.3 (6.1)
Precision [%]	63.1 (11.7)	82.2 (14.5)	87.0 (8.1)	81.3 (14.4)	89.0 (8.1)
Recall [%]	72.7 (10.1)	84.2 (8.8)	85.5 (14.8)	83.9 (9.1)	80.7 (8.4)
Time [min]	0.1 (0.1)	14.9 (1.0)	23.9 (12.9)	1.8 (0.2)	3.6 (1.6)

Table 1 shows the best performance was obtained by the DP method with an average F-score of 85.2%. This method and SBF (82.0%) clearly outperform TM that only achieved an F-score of 66.7%. Given the excessive time obtained by the best methods (more than 150 times the TM's computational time), it is reasonable to discuss if a reduction in N decreases the complexity yet still leads to a sub-optimal F-score. The columns SBF_2 and DP_2 show the sub-optimal results obtained with the parameters $N = 16, g_{min} = 0, Th = 80\%$ and $N = 16, m = 0.03, \beta = 1, Th = 65\%$, respectively.

Note that maximizing the F-score for the DP method with $N = 128$ implies choosing a larger cost slope ($m = 3$), while for $N = 16$ it is preferable to reduce the transition penalizations by selecting $m = 0.03$. When choosing a larger angle interval ($N = 16$) there is more uncertainty given the great angular displacement between directions and it is not adequate to assume the radius for the next direction is close to the one for the previous one. On

the other hand, when the angle interval is smaller ($N = 128$), it is reasonable to accept that the distance from the stone's boundary to the center does not vary significantly hence the increased m.

Table 1 also shows that a reduction in N does not significantly affect the F-score for SBF and DP yet, the gain in computational time is around 8 and 7 times, respectively. Overall the DP method with $N = 16, m = 0.03, \beta = 1, Th = 65\%$ outperforms the other alternatives considering the trade-off between performance and time complexity. Figure 17 shows the performance of each method in a difficult image: SBF achieved the best result given the smaller number of false detections yet, the methods failed to detect some stones with ambiguous borders. Note the multiple false positives detected by TM in the intersection of existing stone circles.

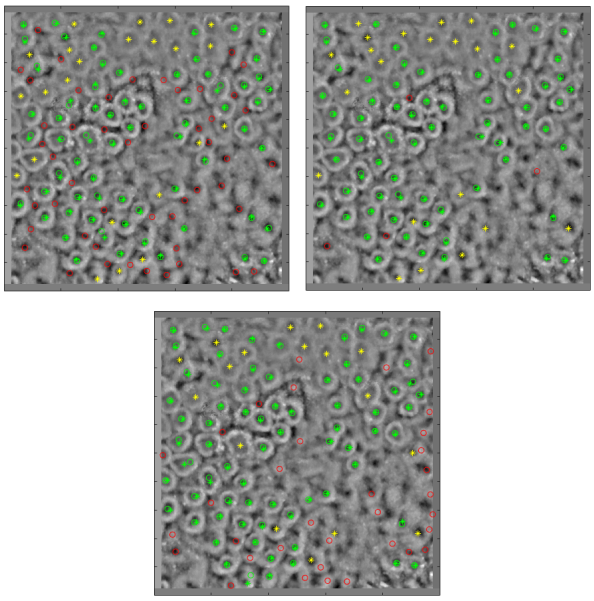


Figure 17: Matching output: TM (top left), SBF (top right) and DP (bottom). Colour code: true positives (green), false positives (red) and false negatives (yellow).

4.3. Experiments - Contour Delineation

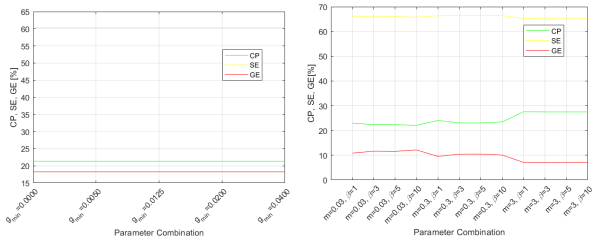
Each delineation method was executed for different parameter combinations, for each of the 24 circles tested, and average values of the evaluation metrics were obtained. The optimal parameter configuration must minimize the average GE percentage.

4.3.1 Sliding Band Filter

The SBF's performance in the delineation task was evaluated by testing different values of $g_{min} \in \{0.0, 0.005, 0.0125, 0.02, 0.04\}$, with a number of directions set to a high value ($N = 360$) so the contour is well defined. Figure 18(a) shows the method is not sensitive to the minimum gradient value.

4.3.2 Dynamic Programming

The parameter values tested for the contour delineation with DP are $m \in \{0.03, 0.3, 3\}$, $\beta \in \{1, 3, 5, 10\}$ and $N = 360$. Figure 18(b) shows that the GE percentage is reduced as m increases, while the CP has the opposite evolution. The influence of β is residual as the curves are approximately constant when m is fixed. The graph also shows that a larger m improves the contour detection results and the most permissive cost parameters ($m = 0.03, \beta = 10$) led to the greatest GE percentage.



(a) (b)

Figure 18: Average CP, SE and GE per parameter combination for SBF (a) and DP (b) methods.

4.3.3 Delineation methods comparison

Table 2 shows the average metrics obtained for the optimal parameters of SBF ($N = 360, g_{min} = 0$) and DP ($N = 360, m = 3, \beta = 1$) in the columns SBF_1 and DP_1, respectively. The superior performance of the DP method is confirmed by the small percentage of GE (7.12%), a reduction of about 10% in comparison with SBF's optimal result. This increase in performance comes at the expense of a complexity 10 times larger than the one taken by SBF.

Table 2: Performance measures for SBF and DP methods with $N = 64$ (columns with subscript 2) and $N = 360$ (columns with subscript 1).

Method	SBF_1	DP_1	SBF_2	DP_2
CP [%]	21.4 (9.4)	27.6 (9.8)	18.5 (8.5)	24.6 (10.3)
SE [%]	60.4 (11.7)	65.3 (10.6)	63.5 (13.0)	68.2 (13.0)
GE [%]	18.3 (18.8)	7.1 (12.3)	18.0 (18.8)	7.2 (13.7)
Time [s]	5.5 (1.8)	50.2 (14.3)	4.4 (1.3)	3.2 (0.8)

For visualization purposes, the methods were also executed for $N = 64$. The columns SBF_2 and DP_2 of Table 2 show the corresponding results. It is

interesting to note that the minimum GE percentage obtained with $N = 64$ was achieved with less restrictive cost parameters $m = 0.3, \beta = 1$ in comparison with the optimal m for $N = 360$, which is a consequence of the explanation given in Section 4.2.4. For the DP method, changing $N = 360$ to $N = 64$ led to a reduction of the average computational time of about 15 times, without compromising the GE percentage. The complexity reduction is not significant for the SBF and its performance is also maintained. Figure 19 shows an example of the methods' performance: the contour points (blue) obtained with SBF present a large variation while the dependency between consecutive radius imposed in the DP method reduced the number of outliers.

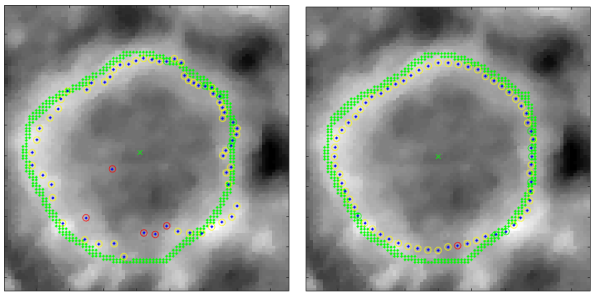


Figure 19: Matching output: SBF (left) and DP (right). Colour code: correct point (green), small (yellow) and gross (red) error.

5. Conclusions

This work focuses on the unexplored subject of automated detection and delineation of stone circles in periglacial terrains, by analysis of DEMs obtained after processing of images captured by a UAV in Antarctica. The algorithms developed are based on Template Matching, the Sliding Band Filter and Dynamic Programming.

In the dataset of 20 elevation maps, the DP approach achieved the best compromise between performance and complexity, reaching an average F-score of 84.3%, outperforming SBF (81.3%) and TM by a significant amount (66.7%). The rigid assumptions limit TM's performance while SBF accommodates deformations and scale changes. The DP's superior performance is due to additional smoothness constraints that improve the gradient noise robustness. This constraint also led to the top performance of DP in the contour delineation task: for the 24 circles evaluated, it reached an average percentage of gross errors of 7.1%, a significant reduction in comparison with the 18.3% obtained by the SBF.

References

[Ballard, 1981] Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122.

[Bandeira et al., 2007] Bandeira, L., Saraiva, J., and Pina, P. (2007). Impact crater recognition on mars based on a probability volume created by template matching. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4008–4015.

[Falk et al., 2019] Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., et al. (2019). U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 16(1):67.

[Hansen and Ji, 2009] Hansen, D. W. and Ji, Q. (2009). In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):478–500.

[Marques and Pina, 2013] Marques, J. S. and Pina, P. (2013). An algorithm for the delineation of craters in very high resolution images of mars surface. In *Iberian Conference on Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science, Vol. 7887, 213-220. Springer.

[Marques and Pina, 2015] Marques, J. S. and Pina, P. (2015). Crater delineation by dynamic programming. *IEEE Geoscience and Remote Sensing Letters*, 12(7):1581–1585.

[Pina et al., 2019] Pina, P., Pereira, F., Marques, J. S., and Heleno, S. (2019). Detection of stone circles in periglacial regions of Antarctica in UAV datasets. *Iberian Conference on Pattern Recognition and Image Analysis*, Lecture Notes in Computer Science, Vol. 11687, 279-288. Springer.

[Quelhas et al., 2010] Quelhas, P., Marcuzzo, M., Mendonça, A. M., and Campilho, A. (2010). Cell nuclei and cytoplasm joint segmentation using the sliding band filter. *IEEE Transactions on Medical Imaging*, 29(8):1463–1473.

[Santiago et al., 2018] Santiago, C., Nascimento, J. C., and Marques, J. S. (2018). Fast segmentation of the left ventricle in cardiac mri using dynamic programming. *Computer Methods and Programs in Biomedicine*, 154:9–23.

[Uxa et al., 2017] Uxa, T., Mida, P., and Křížek, M. (2017). Effect of climate on morphology and development of sorted circles and polygons. *Permafrost and Periglacial Processes*, 28(4):663–674.

[Yuille et al., 1992] Yuille, A. L., Hallinan, P. W., and Cohen, D. S. (1992). Feature extraction from faces using deformable templates. *International journal of computer vision*, 8(2):99–111.