

# Crater Detection Using a New Circle Detection Method

Keywords: circle detection, crater detection, constrained gradient, feature extraction

Abstract: A new circle detection algorithm based on the gradient of the intensity image is proposed. The method relies on two conditions. The “gradient angle compatibility condition” constrains the gradient of a given percentage of the pixels belonging to some digital circles of radius in the range of radii to detect to point towards the centre of the circle or in the opposite direction. The “curvature compatibility condition” constrains the variation of the gradient angle of the same pixels in a range depending on the radius of the circle. These two conditions are sufficient to detect the core of circular shapes. The best fitting circle is then identified. The method is applied to artificial and reference images and compared to two state-of-the-art methods. It is also applied to water-filled crater detection in Cambodia: these craters which might indicate the presence of Unexploded Ordnance (UXO) dating from the US bombing produce dark circles on satellite panchromatic images.

## 1 INTRODUCTION

The presence of Unexploded Ordnance (UXO) resulting from the US bombing during the late sixties and seventies is still preventing the use of the land in Cambodia. When dropped, the bombs produced craters that may still exist today. Many of them are filled with water so that they may appear as circular objects on panchromatic satellite images. The purpose of this study is to extract those craters as they might indicate the presence of UXOs. Similar work was made by Hatfield Consultants (Hatfield-Consultants, 2014) for Laos. The authors used historic Corona satellite images; they computed differences between the original image and its smoothed version, and use these differences in an unsupervised K-means fuzzy classifier. One class contained the impacted areas which were then identified based on geometrical characteristics. The geometry (i.e. the fact that craters are almost circle) is coming at the end of the process. Our approach is rather to start with geometry, i.e. dark circle detection.

Circle detection has been a challenge since the early days of Pattern Recognition and is still arousing interest as recent publications show ((Chung et al., 2012), (Akinlar and Topal, 2013), (Marco et al., 2014)). Exhaustive review of circle detection methods can be found in the introduction of these articles. “Circle” may designate the border of “sphere”, “disc” or “ring”, or, very thin ring, as shown on Figure 1. In this publication, we are interested in “disc” detection, although the proposed method can be used as such for sphere detection and for detecting inner ring circumference. The method may be adapted for the detection of the other types of “circle” but such an adaptation is

not described here.

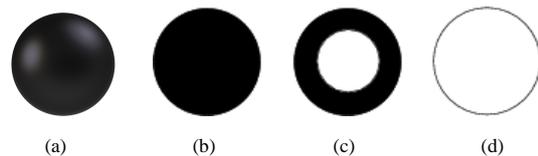


Figure 1: (a) Sphere; (b) disc; (c) ring; (d) thin ring.

Hough transforms and their randomized versions are very popular (see for example (Xu et al., 1990), (Yip et al., 1992)) but are still very time consuming as they rely only on the hypothesis of three edge pixels (i.e. edgels) belonging to a circle and may require complex structures for storing the votes. Some computational time may however be saved using the gradient orientation to constraint edgels belonging to the same circle (see for example (Atherton and Kerbyson, 1999)) or using LUT method ((Chung et al., 2009)).

Most existing approaches use the fact that some basic elements (pixels, edgels, or connected segments) are part of the circumference and combine some of them to generate a centre-radius pair hypothesis. To our knowledge, none of them uses a “blind” centre-radius hypothesis as detecting circles of various radii at each pixel seems time-consuming.

However, performing a first test on the *gradient angle* on a few pre-defined digital circles of radius spanning the radius range to detect allows already to isolate the core of shapes (circle, ellipses, squares, etc.) of corresponding size. This test requires that gradient angle of pixels located on the digital circles are *similar* to the angle of the line joining the pixel to

the circle centre. The second test consists in checking if the *gradient angle variation* of pixels located on the same circle are compatible with the one associated with the considered circle. The second test enables to keep circular shapes only. A counter is set at each pixel considered as a potential centre and is incremented if both tests are positive for the considered digital circle. The percentage of compatible pixels is stored and if the counter represents a significant part of the digital circle, the centre is considered as potential candidate.

A second phase is however necessary to identify the best centres/radius pair among the candidates. In order to ease the second phase, in this publication, we assume that the circles present in the image do not overlap, and thus, if there is a circle at some pixel, it is unique. This assumption is valid for the crater detection application we are concerned with but might not be true for other applications.

This paper is organized as follows. Section 2 present the first phase aiming at extracting centres candidates. In that Section, the gradient angle and the curvature compatibility conditions are presented. The algorithm is then described and applied to an artificial image. In Section 3, a second phase to extract the best centre-candidate and the adequate circle is proposed. In Section 4 the results on the full process applied to three different images are presented and compared to the ones of State-Of-The-Art methods. Discussion and conclusions are provided in Section 5.

## 2 EXTRACTING CENTRES CANDIDATES

In natural images or in scanned graphics, pure step edges are not very probable; edges are rather spanning over a few pixels, which makes hills of gradient norm even broader; in this study we use the Gaussian gradient (Canny, 1986) with  $\sigma = 1$ . In the vicinity of the border of a circle, the gradient orientation is near the line passing through the centre, although the discretization process makes it dependent on the angle and position (see Figure 2). This property is exploited in the first phase of the process.

### 2.1 Compatibility Conditions

In order to have a circle of a radius  $r$  at centre  $c$ , *all* gradient vectors located on the digital version of the circle should point either towards  $c$  (bright circle) or in the opposite direction (dark circle); because of the discretization process, they will not point *exactly*

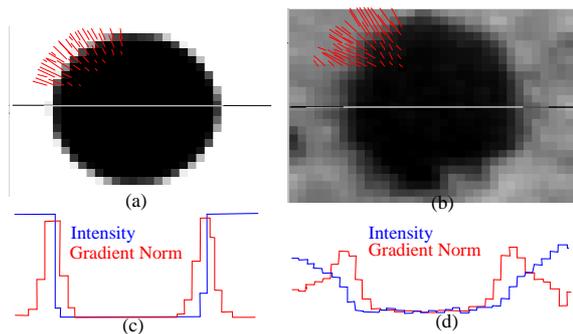


Figure 2: (a) Scanned graphic and (b) natural image with some scaled gradient vectors overlaid and in (c-d) their respective Intensity (in blue) and Gradient norm profiles (in red) at the centre line.

along this direction, but the angle should be small (see the examples of Figure 2).

Let angles be expressed in fraction of radian (i.e. unit= radian/ $\pi$ ) so that the angle range is  $[-1, 1]$ . Let  $p$  be a pixel located on a digital circle  $C$  of radius  $r$  at centre  $c$ , let  $\gamma$  be the gradient angle at  $p$ , and let  $\alpha$  be the angle of the line joining  $p$  to  $c$ . If there exists a circle of radius  $r$  at  $c$ , the **gradient angle compatibility condition at  $c$**  is defined as:  $\forall p \in C$ ,

$$\begin{aligned} \text{Diff}(\alpha - \gamma) &< \varepsilon_a & \text{for dark circle} \\ \text{Diff}(\alpha - (1 - \gamma)) &< \varepsilon_a & \text{for bright circle} \end{aligned} \quad (1)$$

where  $\text{Diff}$  denotes angle difference.

This condition is necessary but not sufficient; indeed, it is also true for “centres” of other shapes such as circles of radius  $r'$  close to  $r$  (depending on the edge profile and on the extent of the filter in the gradient computation), for ellipses of axis close to  $r$ , and for  $n$ -regular polygons (with  $n > 3$ ) fitting in circles of similar radius values. Moreover, depending on the tolerance on the angle difference  $\varepsilon_a$  the condition in (1) will not only hold for the centre of the shape (i.e circle, ellipse,  $n$ -regular polygon) but also for its neighbourhood.

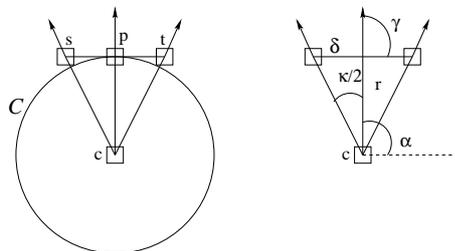


Figure 3: Constraint on gradient angle  $\gamma$  and on curvature  $\kappa$  at  $p$  located on circle  $C$  of radius  $r$ ;  $s$  and  $t$  are located at a distance  $\delta$  of  $p$  in the direction perpendicular to  $\gamma$ .

If  $p$  is located on a digital circle, the local variation

of the angle of the gradient is also constrained. Let  $s$  and  $t$  the two points located at a distance  $\delta$  in the direction perpendicular to the gradient at  $p$ . Let  $\kappa$  be defined according to the Equation 2.

$$\kappa = \text{Diff}(\gamma_s - \gamma_t) \quad (2)$$

where  $\gamma_s$  and  $\gamma_t$  denote the gradient angle at  $s$  and  $t$ .  $\kappa$  is an approximation of the local curvature at  $p$  and will abusively called ‘‘curvature’’ in the following. If  $r$  is the radius of the circle,  $\kappa$  is such that

$$\kappa = 2 * \arcsin(\delta/r) \quad (3)$$

Some tolerance on the curvature should also be allowed to deal with the discretization process. Thus the **curvature compatibility condition at  $c$**  is defined as:  $\forall p \in C$ ,

$$\begin{aligned} 2 * \arcsin(\delta/r) - \varepsilon_k &< \kappa_p < 2 * \arcsin(\delta/r) + \varepsilon_k \\ \kappa_r - \varepsilon_k &< \kappa_p < \kappa_r + \varepsilon_k \end{aligned} \quad (4)$$

where  $\varepsilon_k$  designates the tolerance on the curvature and  $\kappa_p$  the curvature at  $p$ .

Figure 3 illustrates the two constraints for a pixel  $p$  located at angle  $\alpha$  on the circle  $C$  of radius  $r = 6$ .

In the case of an imperfect or partial circle of radius  $r$ , only a fraction  $f$  of the gradient angles located on  $C$  will satisfy condition (1) and (4). The number of pixels  $n$  that should satisfy the angle and the curvature compatibility condition, and  $b$ , the maximum number of non-valid pixels are thus defined by

$$n = f \# C \text{ and } b = (1 - f) \# C \quad (5)$$

where  $\#$  denotes the cardinality.

If the circles to detect have a radius between  $rmin$  and  $rmax$ , thanks to the smooth variation of the gradient direction in the direction perpendicular to the border (see Figure 2), not all corresponding digital circles need to be tested for angle and curvature compatibility. In this publication we use all integer values of  $r$  between  $rmin$  and  $rmax$  by step of two.

There exist several implementations of digital circles (Blinn, 1996). In this study, the pixels of the digital circle  $C$  of radius  $r$  are found by starting at the pixel  $p(i, j) = (0, r)$ , incrementing  $i$  by one, computing  $j$  using the circle equation, and computing the angle using  $\arcsin$  until 45 is reached; the other pixels are found using the circle symmetries (see some resulting circles in Figure 4). The chosen implementation — also known as Bresenham’s circle algorithm — is the one that has the smallest number of pixels while preserving the connectivity.

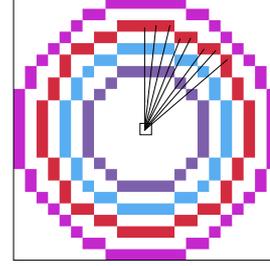


Figure 4: Digital circles used in this study for a radius range from 5 to 12 and a few vectors joining the centre to the pixels lying on the circle of radius  $r = 9$ ; each digital circle is displayed in a different color;

## 2.2 Phase one: algorithm description

### 2.2.1 Prerequisite

1. Compute Gaussian Gradient  $G_x$ ,  $G_y$  and norm  $N$ .
2. Compute Angle of Gradient  $A$ , and curvature  $K$  according to Equation 2.

### 2.2.2 Center Candidates

Let  $\varepsilon_a$ ,  $\varepsilon_k$ ,  $f$  and  $t$  be the tolerance on gradient angle, the tolerance on curvature, the fraction of circle to detect and the threshold value under which the norm of gradient is considered as noise respectively.

1. Let  $C = \{C_0, \dots, C_j, \dots, C_n\}$  be the list of digital circles located at the origin, corresponding to the increasing radii  $r_0, \dots, r_j, \dots, r_n$  with  $rmin = r_0$  and  $r_n \leq rmax$ . For each  $C_i$ , get the list of vectors  $p_{ij}$  joining the center to each pixel located on the digital circle (i.e. using Bresenham’s algorithm) and their angle  $a_{ij}$ ; compute  $z_i = \# C_i$ ,  $b_i$  and  $n_i$  according to Equation 5 and the ideal curvature  $\kappa_i$  according to Equation 3.
2. Initialize two rasters  $R$  and  $V$  for storing the radius and the fraction of valid pixels.
3. Scan the image; for each pixel  $c$ , compute the fraction of valid pixels lying on the digital circles  $\{C_0, \dots, C_j, \dots, C_n\}$  starting with the smallest circle  $C_0$ , as follows. For each vector  $p_{ij}$  of  $C_i$ , get the pixel  $q_{ij} = c + p_{ij}$ , lying on circle  $C_i$  centred at  $c$  and test three conditions:
  - the minimum norm condition:  $N_{q_{ij}} > t$ ;
  - the angle difference compatibility condition according to Equation 1, where  $\alpha = a_{ij}$  (angle of  $p_{ij}$ ) and  $\gamma = A_{q_{ij}}$  (gradient angle at  $q_{ij}$ );
  - the curvature compatibility condition according to Equation 4 where  $\kappa$  is the curvature at  $q_{ij}$  and  $\kappa_r = \kappa_i$ .

If all conditions are satisfied increment the “good” counter at  $c$  (count separately dark and bright circle according to the gradient angle at  $q_{ij}$ ), otherwise increment the “bad counter”. As soon as the bad counter has reached  $b_i$ , the next circle  $C_{i+1}$  is tested. Otherwise test the full circle  $C_i$ , save  $r$  (radius)  $r$  and  $v$  (fraction of valid pixel) in  $R$  and  $V$  respectively. For convenience purpose, a negative value in  $V$  indicates a dark circle. Store  $c$  in a list of potential centre candidates.

### 2.2.3 Core Shapes

Get the connected components of all center candidates. In the case of non-overlapping circles, each connected component will make the core of a circular shape. For each connected components, compute  $c_a$ , the center of gravity of the pixels with the lowest radius  $r$  (it should be a center candidate, but if it is not, consider the 8-neighbours of similar radius with the highest counter).  $c_a - r$  is a first centre-radius approximation of the shape.

## 2.3 Application

The parameter of the proposed method are:

- $\sigma = 1$  and  $\delta = 2$  for the prerequisite Gaussian Gradient and curvature computation,
- $t = 10$  for ignoring pixels with a too low gradient norm,
- $\epsilon_a, \epsilon_k$ , for setting the tolerance on gradient angle and curvature; expressed in fraction of radian (i.e. unit= radian/ $\pi$ ), a reasonable range is ]0.05 0.15[. For simplicity  $\epsilon_a = \epsilon_k$ .
- $f$ , for the fraction of circle to detect. A reasonable range is ]0.7 1[;

The parameter  $f$  and  $\epsilon_a/\epsilon_k$  are not independent. Indeed, if the tolerance rises, the portion of detected circle will rise.

Experiments on geometrical figures on a uniform background (see Figure 5) provide some insight on the method (used with  $rmin = 5$  and  $rmax = 15$ ) and enable to analyse the effect of the parameters  $\epsilon_a, \epsilon_k$ , and  $f$  on the production of candidates. The shapes are identified by numbers on Figure 5 (up). The raster  $V$  displaying the fraction of the smallest valid circle at each centre candidate have been analysed for values of  $\epsilon = \epsilon_a = \epsilon_k$ , ranging from 0.06 to 0.14, and  $f = 0.6$ . An example of such a raster is shown on Figure 5 (down). Although 0.6 is below the recommended value, it enables to see when false alarm occur for low  $f$  values. Connected sets of non-zero values correspond to the core of each shape, except for

the ellipse (26), which generates two connected sets when  $\epsilon = 0.14$ . The results are shown in Table 1. None of the triangle generates a connected component, they are thus ignored in the table.

Type	$\epsilon \rightarrow$ Id	0.06	0.08	0.10	0.12	0.14
Circle	14					
	15	60-100	63-100	77-100	60-100	60-100
Circular Ellipse	10		65-95	60-100	61-100	60-100
	11	60-98	59-100	59-100	60-100	60-100
	18	60-85	61-95	63-100	63-100	60-100
	23	60-70	60-85	60-96	60-100	60-100
Elipse	1	61-85	59-69	60-88	60-95	60-100
	4		90	71-85	60-92	60-92
	12	59-70	60-90	59-95	59-95	60-100
	26					60
						60
Square	2					61
	3				66	59-84
	5				60	60-63
	6					59-67
	7					61-71
	17	60-85	63-95	59-97	60	60-63
	19					61
	21					65-84
	29					61-63

Table 1: Percentage of compatible pixels inside the connected component corresponding to the shapes displayed in Figure 5

The image contains two perfect circles (14, 15), three ellipses almost circular (10, 11, 18, 23), other ellipses (1, 4, 12, 26) and several squares (2, 3, 5, 6, 7, 17, 19, 21, 29). The analysis of the table suggests to use the method with  $\epsilon \simeq 0.12$ . Then, for any value of  $f > 66$ , all circles and almost circular ellipses will be detected, and so will be the ellipses (1, 4, 12) which might be considered as false alarms in some applications. Thus with  $\epsilon = 0.12$ , only the nine shapes 14, 15, 10, 11, 18, 23, 1, 4, 12, are detected by our method provided that  $f > 66$ .

The same image has been processed using the isophote (Marco et al., 2014) and the EDCircles (Akinlar and Topal, 2013) method providing seven shapes and nine shapes respectively (see Figure 6). The results of the three methods are similar except that:

- The isophote method does not detect shapes 23 and 4;
- The EDCircles method does not detect shape 23 but detects 26.

## 3 BEST CENTER-CIRCLE CANDIDATES

The first phase enables to detect the core of each circular shape. The second phase needs to find the best center-radius pair; a first center approximation ( $c_a$ ) is obtained taking the center of gravity of pixels

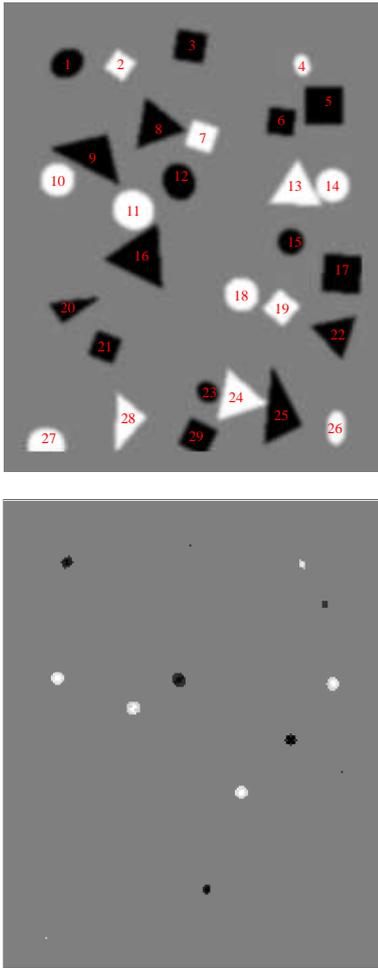


Figure 5: (up) Test image made of random geometrical dark and bright shapes; (down) Example of Raster  $V$  for using  $\epsilon_a = \epsilon_k = 0.12$  and  $f = 0.6$ ; white= 100% detection for bright circle, grey=0% , black= 100% detection for dark circle;

of the lowest radius value inside the connected component. The exact center will be in the vicinity of this point, and provided that there is a unique circle near  $c_a$  (working hypothesis), the exact radius should be in the range of  $[r \ r + 2]$ .

One could use any existing circle detection method in the neighborhood of  $c_a$ , or use active contours with the circle of radius  $r_{min}$  as initial contour or consider the estimation of the parameter of the potential circles in these areas as a least square estimation problem such as in (Zelniker, 2006).

The problem thus becomes an optimization problem: find the best circle given a center and find the best center within the set associated with the same label. Again, the best optimization function will depend on the application: e.g., is a partial circle perfectly fit-

ting the border of the shape better than a more complete circle further away from the border at some location? Different choices will lead to different optimization functions.

Nevertheless, three factors are important in the process: the gradient angle and curvature compatibility as already identified in the first phase, and the location of the shape border which should be made of the local maxima of the gradient in the direction of the gradient.

The current implementation of the second phase is described as follows. For each shape label, each digital circle of radius in the range of  $r$  to  $r + 2$  at center  $c_a$  (identified by phase one) are tested as the potential circle. For each of these circles, pixels satisfying the three compatibility conditions (Norm, angle and curvature) are considered: the projection of the gradient along the line joining the pixel to the center is computed and the average  $n_a$  is performed on the circle. The fraction of valid pixel  $v$  is computed. The circle with the highest value of  $n_a * v$  is considered as the best fit. The second phase is thus very similar to the the first one, excepted that:

- more precise circles are used ( $r$  by step of one),
- only center of gravity of shapes are tested
- an additional computation involving the gradient is performed at each pixel

An integer value is thus obtained for the center and for the radius.

## 4 RESULTS

The full detection method has been applied to the artificial image displayed in Figure 5, on a reference image displayed in Figure 7 and on a satellite image and compared to the results obtained by the using the isophote (Marco et al., 2014) and the EDCircles (Akinlar and Topal, 2013) method.

### 4.1 Application on an Artificial Image

The results of phase one on the image displayed in Figure 5 has already been described in Section 2.3. The method is used with  $\sigma = 1$  and  $\delta = 2$  for the norm and curvature computation,  $\epsilon_a = \epsilon_k = 0.12$ ,  $t = 10$ ,  $f = 0.80$ ,  $r_{min} = 5$ ,  $r_{max} = 15$ . The best-fitting circles are shown on Figure 6.

### 4.2 Application on an Reference Image

The method with the same parameter value has been use to detect the circular shapes on the image shown in Figure 7 found at

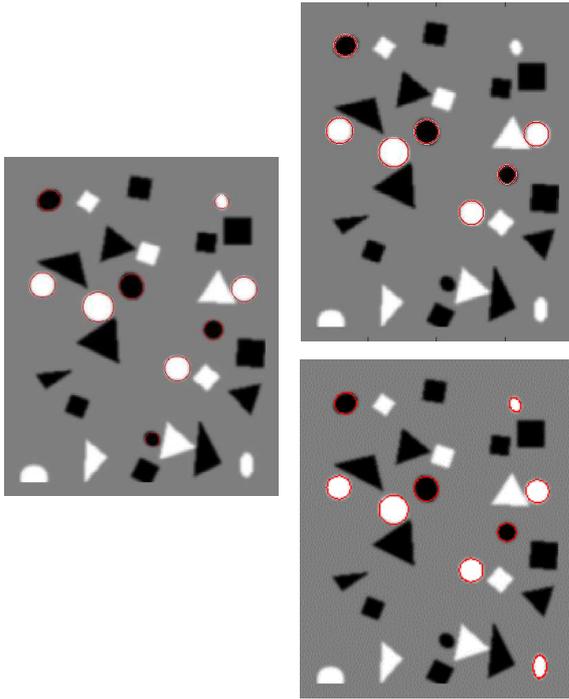


Figure 6: Detected circles superimposed on Test image made of random geometrical dark and bright shapes (200×250); left: our method; right up: isophote; right down: EDCircle

<http://ceng.anadolu.edu.tr/cv/EDCircles/except>

except for  $rmax = 50$ . As the working hypothesis assumes only one disc per centre, only the inner rings of the signs are detected. The result of the EDCircle method can be found at the same dress.

### 4.3 Application on a Satellite Image

The study area is located in the eastern part of Cambodia near the border with Vietnam, in a rural zone (Choam Kravien) that was heavily bombed during the Vietnam War. The terrain is quite flat and the landscape consists mainly of agricultural land. The panchromatic image used was acquired by the Pleiades 1A satellite on 6 March 2014. It covers 100 km, with a spatial resolution of 0.5 meter and a pixel depth at acquisition of 12 bits, but 8 bits image have been used. The method is used with  $\sigma = 1$  and  $\delta = 2$  for the norm and curvature computation,  $\epsilon_a = \epsilon_k = 0.12$ ,  $t = 10$ ,  $f = 0.80$ ,  $rmin = 5$ ,  $rmax = 15$ ; only dark circles are detected and an additional threshold on the circle average norm  $n_a$  is performed with  $t = 50$ .

The results using our method, isophote and EDCircles are respectively shown in Figure 8, 9 and 10.

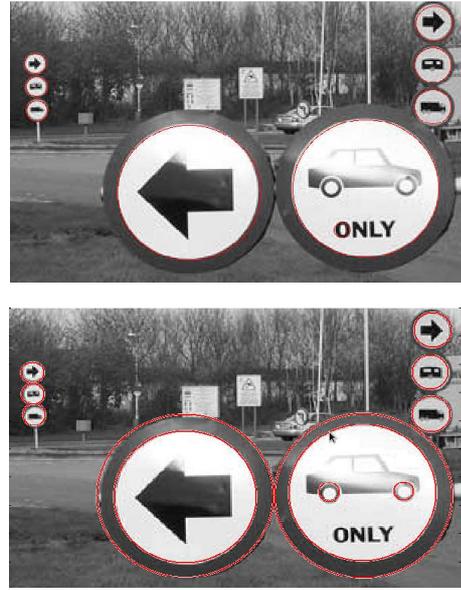


Figure 7: Detected circles superimposed on Sign Image (300×498); (left) our method (right) isophote

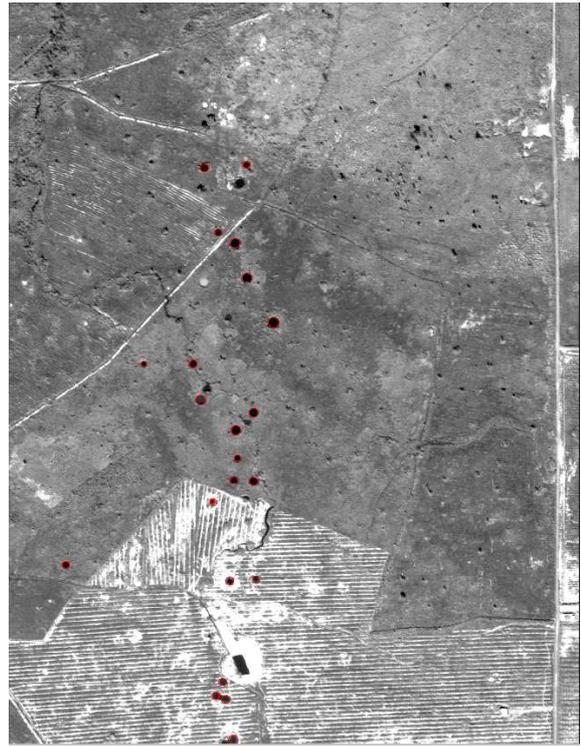


Figure 8: Detected circles (our method) superimposed on 1309×1855 part of the panchromatic image;

The comparison of the results shows that many circle shapes are missed by the EDCircle; many are detected by the isophote method but the current method provide better results. A comparison with vi-

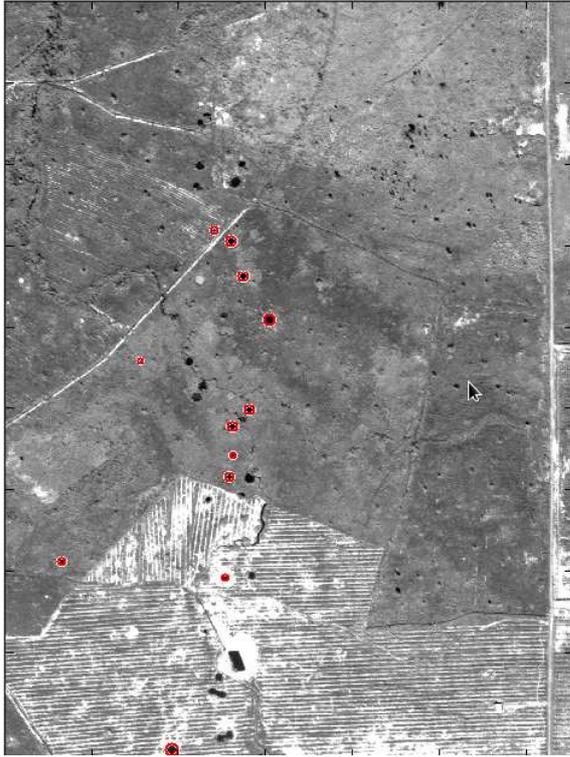


Figure 9: Detected circles (isophote method) superimposed on 1309×1855 part of the panchromatic image;

sual inspection should still be performed to have a quantitative view of the performance of the method in this context.

## 5 DISCUSSION AND CONCLUSIONS

The new proposed two-phase method to extract circles seems promising. The originality of the method resides in the first phase which uses a gradient angle and curvature compatibility constraint at pixels lying on various digital circles to produce sets of connected pixels belonging to potential circle candidates. The second phase consists in finding the best pair of circle-centre which optimizes the circle position. The method has been compared to State-of-the-Art methods on an artificial image, a reference image and a panchromatic image of Cambodia for crater detection. The results seem promising. The process could be performed in parallel, not only at each pixel, but also for each of the digital circles, which makes the method efficient. Although the method in the current form can deal with any radius range and various portions of circle, it is better suited to the detection of

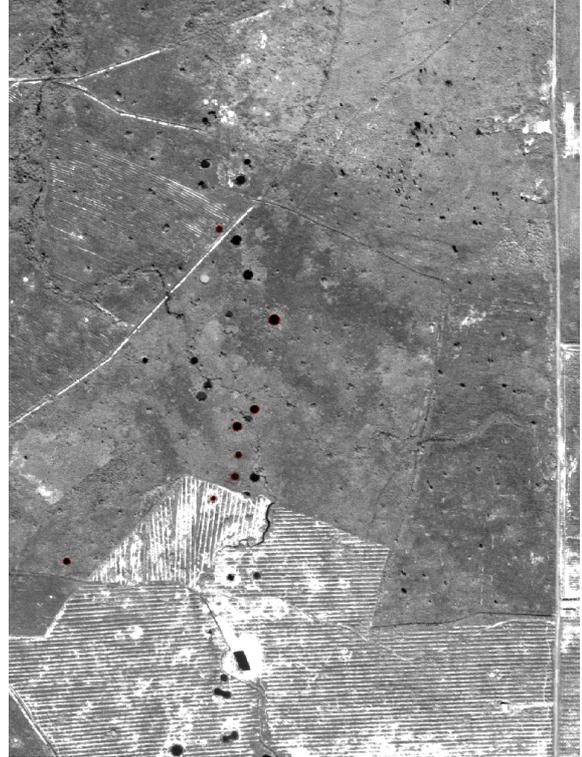


Figure 10: Detected circles (EDCircle method) superimposed on 1309×1855 part of the panchromatic image;

full disconnected circles whose radius is small compared to the image size. The method could be generalized in order to extract overlapping circles, rings and thin rings by the elaboration of another second phase (thin ring detection would require a further line detection step as prerequisite) and also to the detection of squares, by adapting the curvature compatibility constraint.

## ACKNOWLEDGEMENTS

Special thanks to Dr. De Marco who processed our data with his method, and to Dr. Akinlar and Dr. Topal for their site at

<http://ceng.anadolu.edu.tr/cv/EDCircles/except> which allows to process any data with their method.

## REFERENCES

- Akinlar, C. and Topal, C. (2013). Edcircles: A real-time circle detector with a false detection control. *Pattern Recognition*, 46.
- Atherton, T. J. and Kerbyson, D. J. (1999). Size invariant circle detection. *Image and Vision Computing*, 17:795803.
- Blinn, J. (1996). *Jim Blinn's Corner: A Trip Down the Graphics Pipeline*. Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann Publishers.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, pages 679–697.
- Chung, K.-L., Chen, P.-Z., and Pan, Y.-L. (2009). Speed up of the edge-based inverse halftoning algorithm using a finite state machine model approach. *Computers & Mathematics with Applications*, 58(3):484 – 497.
- Chung, K.-L., Huang, Y.-H., Shen, S.-M., Krylov, A. S., Yurin, D. V., and Semeikina, E. V. (2012). Efficient sampling strategy and refinement strategy for randomized circle detection. *Pattern Recognition*, pages 252–263.
- Hatfield-Consultants (2014). Uxo predictive modeling in mmg lxm sepon mine development area. Technical Report 1791.D6.1, Hatfield consultants, East Lansing, Michigan.
- Marco, T. D., Cazzato, D., Leo, M., and Distanto, C. (2014). Randomized circle detection with isophotes curvature analysis. *Pattern Recognition*.
- Xu, L., Oja, E., and Kultanen, P. (1990). A new curve detection method: Randomized hough transform (rht). *Pattern Recognition Letters*, 11:331338.
- Yip, R. K., Tam, P. K., and Leung, D. N. (1992). Modification of hough transform for circles and ellipses detection using a 2-dimensional array. *Pattern Recognition*, 25:10071022.
- Zelniker, E. E. (2006). Maximum-likelihood estimation of circle parameters via convolution. *IEEE Transaction on Image Processing*, 16:865–876.