THE JOURNAL OF NAVIGATION (2019), **72**, 1070–1088. © The Royal Institute of Navigation 2019
doi:10.1017/S0373463318001169

A new Concentric Circles Detection method for Object Detection applied to Radar Images

José Miguel Guerrero[1], Andreas Muñoz[2], Matilde Santos[1] and Gonzalo Pajares[1]

[1](*Complutense University of Madrid, Computer Science Faculty, Spain*)
[2](*CIRCE Foundation (Research Centre for Energy Resources and Consumption)*)
(E-mail: jmguerre@ucm.es)

In this work, a new concentric circles detection method for object detection is proposed. It has been applied to the images of a commercial radar, captured with a Charge-Coupled Device (CCD) camera. The processing includes the detection of centres and concentric circles in the images and the identification of the radar scale. Several methods found in the literature have been applied and compared with our novel proposal for multiple concentric circles detection, called "Propagation Method based on Circular Regression". This methodology has been validated with real radar images, proving its efficiency in obtaining the distance of any object to a marine vessel, with high accuracy and low computational cost, in real time. This system can not only be applied to most existing radars in the market by adjusting the parameters of each model but our proposal for concentric circle detection can be also applied to other sensing applications.

KEYWORDS

1. Obstacle detection. 2. Radar. 3. Image processing. 4. Circles detection. 5. Real-time alarm system.

Submitted: 30 April 2018. Accepted: 28 December 2018. First published online: 27 February 2019.

1. INTRODUCTION. In recent years there has been a great growth in navigation for various reasons: commercial, tourism, research, military, environmental monitoring, etc (Menoyo and Santos, 2016). In all of them it is important to have systems that allow obstacle detection and collision avoidance (Bonin-Font et al., 2017; Polvara et al., 2018; Yang et al., 2018). In some cases, object detection in marine environments can be crucial, for example, for rescue missions.

Different types of range sensors are used for obstacle detection on water surfaces, including laser range finders (Blaich et al., 2015), sonar (Heidarsson and Sukhatme, 2011), vision (Bovcon et al., 2018), scanning Light Directions and Ranging (LIDAR) (Halterman and Brush, 2010) and radar (Zhuang et al., 2016). Radar is the most used ranging-based sensor for computing distances from a few tens of metres to several kilometres. Radar information is very significant for carrying out some missions and can be crucial for obstacle avoidance.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Radar images provide important and necessary information for navigation. This information must be analysed, classified and interpreted quickly and reliably, but there is a lack of efficient radar image processing algorithms due to the intrinsic characteristics of these images.

Commercial off-the-shelf marine radars are sold with a display, but it is usually quite expensive to obtain a digital network interface to generate a map for obstacle detection. Cheap commercial radar systems only provide information on a display without standard interfaces. These economical commercial radars do not usually provide a warning system to send alarms wirelessly to some common electronic devices, such as mobile phones or tablets, although some modern small boat radars come with an inbuilt tracking system (Automatic Radar Plotting Aid, ARPA) that provides automatic collision alert information (Bole et al., 2014).

In this paper we propose a new object detection method based on radar images that can be seen as an aid to navigation. This novel image processing algorithm includes both the detection and location of centres and concentric circles in images and, in the particular case of radar signals, the identification of the scale at which the radar is operating.

It has been applied to the detection of concentric circles in the images taken from the screen of a commercial radar (Furuno 1715, 2018), captured with a Charge-Coupled Device (CCD) camera. Several methods found in the literature have been also applied and compared with our novel proposal, called "Propagation Method based on Circular Regression". This methodology is validated with real images, proving its efficiency.

Another contribution of this paper is the development of a low-cost alarm system to be installed on a radar monitor in a small boat or even in autonomous vessels or robots. This system can determine the distance of any object to the marine vessel in real time, with high accuracy and low computational cost, by processing the radar images. It sends a warning to the people responsible for the safe navigation of the vessel when a close obstacle is detected. This could help the operator to take appropriate actions in a faster and more effective way.

It is based on using the existing vessel's radar, a webcam and a Raspberry Pi. This very cost-effective system can be applied to most existing commercial radars by simply adjusting the parameters that characterise each model. Even more, it can be readily expanded with different software modules depending on the user and their final purpose.

The structure of the paper is as follows. Section 2 presents related works on circular shape detection in images. Section 3 describes the new proposed method for multiple concentric circle detection and radius estimation. It also describes the radar scale recognition used for the radar signals. Results are discussed in Section 4. Section 5 presents a real application with radar images, describing the experimental set up and the obstacle detection system. The main conclusions end the paper.

2. BACKGROUND.

There are many works that use vision techniques for obstacle detection and collision avoidance, path planning and navigation (Chaves, 2018). Nevertheless, only a few have focused on the marine environment and even fewer use radar images, despite radar being one of the most common sensors in vessels (Gupta and Singh, 2014).

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Some notable exceptions are found in Almeida et al. (2009), where the authors used a marine radar and other sensors to detect obstacles and achieve collision avoidance in an Unmanned Surface Vessel (USV). In Zhuang et al. (2016), a radar-based collision avoidance system was proposed to solve the problems related to USV autonomous navigation, applying vision techniques. The method described in Lee et al. (2004) is based on fuzzy logic combined with a virtual force field method, and it can be used for either track-keeping or collision avoidance, both with stationary and moving obstacles. In the paper by Onunka and Bright (2010), techniques for obstacle detection in autonomous marine craft navigation using a continuous wave radar system were discussed. The path planning algorithm proposed in Oleynikova et al. (2010) performed computer vision morphological operations on the radar image to find a suitable path around a coastline and choose an appropriate next waypoint.

This scarcity of papers may be due to the particular idiosyncrasy of the radar sensor. Radar signal processing demands circle detection to obtain the range of the circles displayed on the radar. The most widely used method for detecting circles in digital images is the Circle Hough Transform (CHT) (Muammar and Nixon, 1989), with multiple different implementations. This range of techniques are very robust, but they usually require large memory space and have a high computational cost. In addition, noise significantly affects the precision of the final result. To minimise these limitations, there are variants such as the randomised Hough Transform (HT) (Xu et al., 1990), probabilistic HT (Shaked et al., 1996) or fuzzy HT (Han et al., 1993). Although these developments help to reduce the processing time, weakness related to the noise remains an important factor.

As an alternative to HT-based methods, different optimisation strategies, mainly evolutive algorithms, have emerged and have been extensively used as an efficient tool for shape identification. An example is the Roth and Levine (1994) heuristic algorithm for curves and surface extraction that can be parametrised in images. A further example is a circle detector based on Genetic Algorithms (GA) proposed by Ayala-Ramirez et al. (2006), whose proposal can detect several circles, but not imperfect circles. A third example was proposed by Cuevas et al. (2010), who applied the heuristic Discrete Differential Evolution (DDE) technique. These algorithms produce good results even with noise, but require long processing times, which would prevent their use in real-time applications. A summary on evolutionary techniques applied to multi-circle detection on images is presented in Cuevas et al. (2017).

Recently, new algorithms for circle detection in real time have been developed to improve results in environments where the response time is critical (Frosio and Borghese, 2008; Akinlar and Topal, 2013; Liu et al., 2014).

However, despite the generalised use of concentric circles in industry, their study and the development of specific solutions for their detection is not very widespread. Some solutions have been proposed to solve industrial problems such as the calibration of cameras (Jiang and Quan, 2005) or the identification of these circles in electronic plates (Nao et al., 2010). Generally, most of these algorithms first estimate the centre by parametrising a circumference, and then obtain the rest of the circumferences. In Chen et al. (2012), the authors solve the problems caused by noise and discontinuities with the CHT. Silveira (2005) also presents a method for the detection of multiple concentric circles based on HT. The procedure is separated into two stages, one for centre detection and another for radius determination, in order to reduce computation time and memory space.

In view of the results found in the literature, this paper proposes a methodology for concentric circle detection that can be applied to any images in real time.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

3. CONCENTRIC CIRCLE DETECTION. The detection of concentric circles in images is a specific case of pattern recognition and, more specifically, of the detection of generic circles. It is not only about finding all the circles in the image, but also to detect a common centre for all of them. Although there are several approaches for the general case of detecting circumferences, for this particular case of concentric circle detection the studies are scarcer.

In this section the methodology for concentric circles detection will be explained. It is divided in three steps: edge detection (Section 3.1), centres estimation and the radius of concentric circles determination (Section 3.2). In each step, several methods are tested, some of them based on well-known techniques, and also a new proposal, which has been compared with the existing ones to evaluate its performance. Previously, a pre-processing of real images has been necessary to improve the performance of the subsequent procedures.

3.1. *Edge detection.* The techniques that will be used in this paper to detect the circles work with edge points. These points of an image are characterised by a quite different value from its neighbours. To highlight these pixels, edge detection techniques are applied (Pajares and de la Cruz, 2007). First the original coloured image must be transformed into a greyscale one (Kanan and Cottrell, 2012). The result obtained after edge detection is a greyscale image with a black background and edges highlighted in white. After that, this image is transformed into a black and white (binary) image applying Otsu's method (Otsu, 1979). Finally, a morphological operation of dilation to get thicker and crisper lines is applied.

3.2. *Circle detection techniques.* In this paper, three circle detection techniques have been applied: the well-known Circle Hough Transform (CHT); the method proposed by Silveira (2005) based on three-edge points, and a new Propagation Method for multiple concentric circles detection based on Circular Regression (PMCR) designed for this purpose.

3.2.1. *Circle Hough Transform (CHT).* The Circle Hough Transform (CHT) is a further development of the Hough Transform (Hough, 1962), designed for noisy environments.

The Hough transform is a technique originally used on image processing to detect patterns and complex forms in binary images (Illingworth and Kittler, 1988). The basic idea of this method is to transform the data from the image space to a parametric space.

The HT can be applied to any geometric form, including circles. In a two-dimensional space, the circles are described by Equation (1):

$$(x - a)^2 + (y - b)^2 = r^2 \tag{1}$$

where (a, b) is the centre of the circle and r the radius. The parametric space is now defined by three dimensions, (a, b, r), and the crossing point will be in that three-dimensional space. There are multiple variants of the CHT (Yuen et al., 1990) but not all are focused on concentric circles.

3.2.2. *Detection of concentric circles based on three edge points.* This solution is based on the method described by Cao and Deravi (1992), in which centre detection is carried out by using three points that belong to the same edge. For the selection of these points, the direction of the edge gradient and a series of searching rules are used. Due to the use of the gradient, this method is quite sensitive to noise. To improve this proposal,

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Silveira (2005) used the same methodology but used geometric rules and constraints to reduce the effect of noise. In addition, it can also detect partially hidden or irregular circles.

3.3. *New Propagation Method for multiple concentric circles detection based on Circular Regression.* The main problem of the above methods is the computational load and the complexity. Therefore, we propose a new method for multiple concentric circles detection that we have called the Propagation Method, and it is based on circular regression. The process is divided into two steps: detection of the centre and determination of the radius of each circumference.

(1) For centre detection, after the determination of the edges of the image, the points of each edge are used to adjust the circumference that better approximates the cloud of points. This adjustment is performed using a linear square minimum. That is, the problem consists of finding the best circumference that matches a series of points, or finding the values of $x \in R^2$ and $r \in R^+$ that belong to a set of points $a_j \in R^2, j = 1, 2, \ldots, m$, that minimises Equation (2):

$$\min_{x,r} \sum_{j=1}^{m} \{F_j(x, r)\}^2 \tag{2}$$

where $F_j(x, r)$ is the distance from point a_j to the adjusted circle and x is the centre and r the radius of the circle. The centres beyond the boundaries of the image are discarded, and the others are stored in the array of possible candidates to be used as the input of the algorithm developed to find the real concentric circle centre.

(2) The new concentric circle detection centre works with the coordinates of the candidate centre vector previously obtained. Starting with this data vector, each possible point is evaluated using the general equation of the circumference. The Propagation Method works as follows.

Starting from the candidate centre (a,b), the radar image is scanned for a specific range of radii. For each radius, the corresponding circumference is calculated using Equation (1). The inputs of the function that estimates the radius are the amount of points of the circumference we work with, the interval between the radii that we want to check, and the precision required to identify a circle. The configuration of these input parameters will allow us to detect even incomplete or deformed circles.

Once a circle is drawn on the binarized image, the values of the pixels are evaluated (whether it belongs to an edge or not). Then, if the percentage of white pixels (edges) that match the circles exceeds a previously fixed precision threshold, a circumference of radius r has been found. As the circles must fit entirely in the image, the maximum radius will be half of the shortest side of the image, and the minimum radius is zero. Finally, it is checked that no circle has been counted twice.

The thickness of the circles is not known in advance, therefore the interval between radii is variable. If the distance between two circles that have been detected is smaller than a fixed threshold, only one is considered. The radius of that circle will be the one that has the highest percentage of edge points, because its precision is higher.

3.4. *Methods for pattern recognition in radar images.* There are different solutions for pattern recognition. For the particular case of radar images, a template matching technique has been selected due to the requirements of designing a fast, reliable and scalable system. The final goal is to develop a strategy that can be used for different radar models

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Table 1. FURUNO 1715 radar scales. Range rings interval (km).

Range	0·125	0·25	0·5	0·75	1	1·5	2	3	4	6	8	12	16	24	36
Rings Interval	0·0625	0·125	0·125	0·25	0·25	0·5	0·5	1	1	2	2	3	4	6	12
Number of rings	2	2	4	3	4	3	4	3	4	3	4	4	4	4	3

in which the scales are previously fixed, and templates can be easily obtained. That is, the possible values to be identified must be limited.

In our case, the FURUNO 1715 radar has the possible values shown in Table 1.

Different algorithms can be applied to calculate the similarity between the original image and the template through the function F. We have selected two, the Sum of Squared Differences (SSD) (Kapadia et al., 2017) and the Normalised Cross Correlation (NCC) (Briechle and Hanebeck, 2001), to compare their efficiency for this application.

4. APPLICATION TO REAL RADAR IMAGES. In this section, the methodology proposed in the previous section is applied to real images obtained with the Furuno radar in a real scenario. The two steps, edges (Section 3.1) and circle detection (Section 3.2 and 3.3) will be applied to these images. In addition, results of the application of template matching to radar scale recognition is also shown and discussed.

4.1. *Edge detection in radar images.* Some of the most common edge detection techniques (Pajares and Cruz, 2007) have been used for image pre-processing. Table 2 summarises the algorithms selected and applied in this study, showing the number of edges detected in each case. Small edges have been discarded because in order to apply the following step of the procedure, at least three points of the circle with enough distance between them are necessary.

A vertical line appeared in most of the images that made it difficult to detect multiple circles due to the continuity of the stroke that generates only a group of lines. Only a Horizontal Sobel filter does not generate that vertical line. According to Table 2, this technique obtains the smallest number of valid edges, thus meaning that it will require fewer edges to analyse, decreasing the processing time.

4.2. *Circles detection.* The second step is the concentric circles detection. The three methods presented in the previous section have been applied: Circular Hough Transform; three-edge points, and the novel method proposed based on Circular Regression.

The number of detected circles will depend on the edges, and the computational time will depend on the candidate centres found that have to be checked.

4.2.1. *Circular Hough Transform.* Different algorithms were initially chosen to implement the CHT, but only the solution proposed by Bone (2010) gave good results. This solution has some parameters that need to be adjusted.

In the case here, the whole image has to be analysed, and the range of radii is only limited by the fact that the circles must fit completely within the image, establishing the

Table 2. Average number of valid edges for circle detection with each operator.

	Canny	Sobel	Horizontal Sobel	Roberts	Prewitt	Log	Zerocross
Number of valid edges	92	48	30	46	51	50	50

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

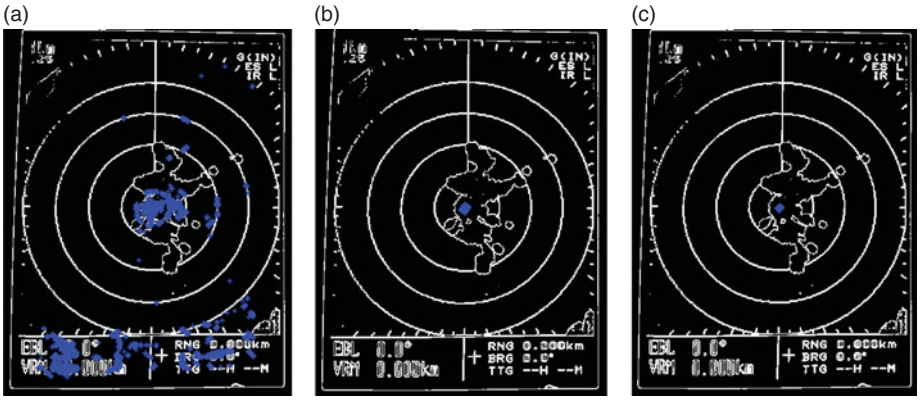


Figure 1. Centres obtained with a decision threshold of a) 50%, b) 80%, c) 100%.

maximum radius as half of the shortest side of the image. To select the decision threshold, several simulations were run to see how this parameter affects the final results (Figure 1). Although its determination is done by trial and error, once a fixed value is selected, the percentage of circumference that this value represents will depend on its length. That is, the threshold can be defined as a function of the radius. For example, a threshold given by $0{\cdot}8\pi r$ means that 80% of the points of the circumference must belong to an edge.

As can be observed in Figure 1, when increasing the threshold, the number of detected centres decreases and the precision increases. On the other hand, the computational time is practically the same with any threshold. Taking all of this into account, the selected value was $0{\cdot}8\pi r$. Using this threshold, the number of detected centres is similar to the one obtained with other more restrictive thresholds, and the accuracy is good.

4.2.2. *Detection of concentric circles based on three edge points.* The second method used to detect concentric circles is the one based on three edge points. The algorithm goes through each edge searching for three points that meet the condition of minimum distance between them. Thus, the higher the number of points calculated, the larger the computational time.

The thickness of the edges generates a lot of points that are very close to the already found ones, which causes oversampling and slows down the process. To avoid this problem, every time an initial point is evaluated and there are other two that meet a minimum distance condition d_{min}, the next initial point is chosen at a distance $\delta_{initial}$. This distance has been determined empirically. The aim is to find enough possible centres to find the real one, but without having too many, which would increase the searching time.

Figure 2 shows the whole process of centre detection.

Several simulations with different combinations of $\delta_{initial}$ and d_{min} were run. The results are shown in Table 3, where the relevant parameters are: centres detected, average number of valid centres, and the average execution time of the circle detection process.

The fastest case ($\delta_{initial} = 1,500$) was discarded because only two valid centres were detected, which means that in noisier images it could not detect any centre. Therefore, the chosen configuration for the final tests was $\delta_{initial} = 800$, $\delta_{initial} = 200$.

4.2.3. *New Propagation Method for multiple concentric circles detection based on Circular Regression.* Finally, the third method proposed to detect concentric circles was

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press



Figure 2. (a) Original image, (b) detected centres, (c) valid centres, (d) detected circles using the valid centres.

Table 3. Comparison between different tested configurations.

$\delta_{initial}$	200			300	400	500	800	1000	1500
d_{min}	100	200	300	200					
Execution time (s)	2,619	2,059	6,692	1,871	1,661	1,514	1,222	1,206	857
Detected centres	2,125	1,775	48	1,775	1,775	1,775	1,775	1,775	1,572
Valid centres	29	12	0	12	12	12	12	12	2

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

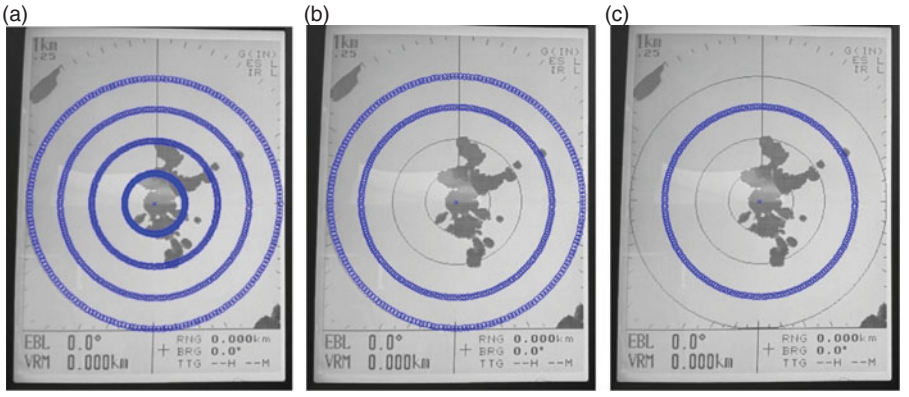


Figure 3. Circles detected with a precision of a) 35%, b) 40%, c) 65%.

applied. There are three parameters to be adjusted: (1) accuracy, which indicates the percentage of the detected circle necessary to consider it so; (2) the radius increment, which determines the minimum distance between one radius and the next one in pixels; and (3) the number of points that define each circumference.

These have been analysed independently, starting with the following initial configuration: 35% precision, 0·5 pixel radius increment, and 200 points per circle.

4.2.3.1. *Accuracy.* Figure 3 shows the results obtained with different precision values. With the initial value, precision = 35%, all the circles were detected. When this value increases, information is missing, until the precision value reaches 65% where the detection is only occasional. This is due to the fact that the edge extraction method does not obtain perfect circumferences, but sections of them. Therefore, in order not to discard incomplete but valid circles, the value of 35% accuracy was selected.

4.2.3.2. *Radius increment.* The different values of radius that have been tested are: 0·5, 2 and 5 pixels (Figure 4). It can be observed that, as the radius increases, it is more likely that some circumferences are missing due to the distance between them. Radius increment strongly depends on the thickness of the circles. As in our case the circles are quite thin, it is better to choose the smallest value, that is, 0·5.

4.2.3.3. *Number of points to define a circumference.* Figure 5 shows the results for 40, 100 and 200 pixels. The first one is too low, and it fails to detect the smallest circumference of the image because it is partially covered, and the circles are poorly represented. As the number of points increases, all the circumferences are detected and the circles are easier to identify, even when they are partly covered or incomplete, showing a continuous circumference. For the tests, this parameter is set to 200.

4.3. *Radar scale recognition results.* Template matching was used to identify the radar scale of the real images. Therefore, as patterns are needed, a group of images with good resolution, high contrast and homogeneous luminosity were selected.

Although using individual digits as patterns was an option, results were not good enough. We then used all the possible values of scales as the pattern set, taking advantage of the low number of scales to be recognised (Table 1). The set of patterns is formed by the six different scales values that could be found in the radar image database (Figure 6, top images).

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

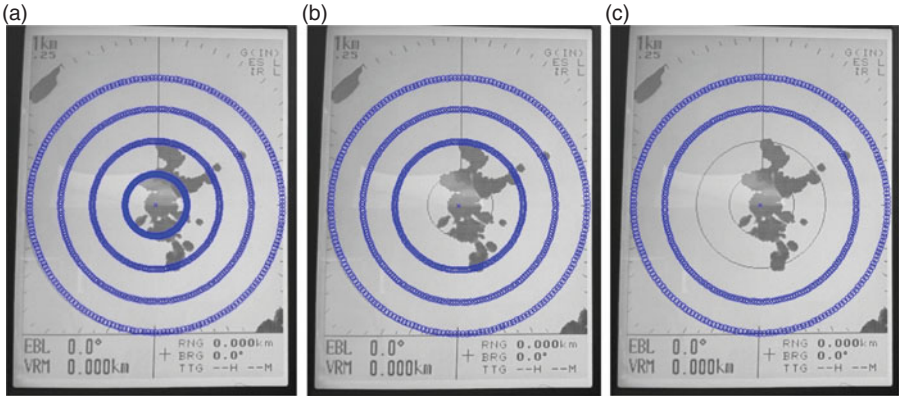


Figure 4. Circles detected with a radius increment of (a) 0·5, (b) 2, (c) 5.

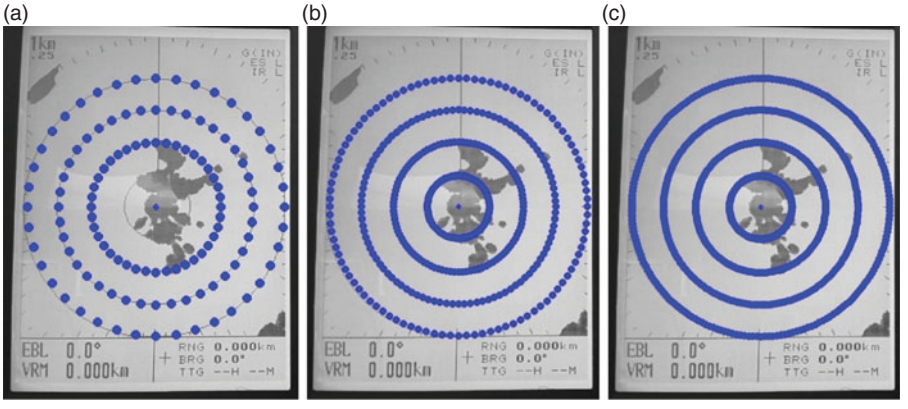


Figure 5. Circles detected for a number of pixels of (a) 40, (b) 100, (c) 200.

Two template matching techniques were initially applied in order to compare the results, Normalised Cross-Correlation (NCC) and Sum of Squared Differences (SSD).

Figure 6 shows an example of the results of applying both techniques, NCC and SSD, to each of the available templates of the same radar scale image. The dots, blue for NCC and red for SSD, represent the position of the maximum value obtained by the function F described in Section 3.3. The SSD method was discarded because it was not able to discriminate between patterns and it gave many false positives, obtaining the maximum similarity outside the limits of the radar scale image (this can be seen in Figure 6(a), 6(c) and 6(f), where the red dot does not appear). So, the maximum value of F is obtained by the NCC method (blue dots), and this value is analysed in order to select the best threshold to match the template and the radar scale image.

Several thresholds were tried in applying the NCC method. First it was set to 0·8, high enough to reduce the number of false positives and at the same time significant information is not missed. Using this, 87% of the images were correctly identified, eight were not classified and five were wrongly classified; the average execution time was 1·2 seconds.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

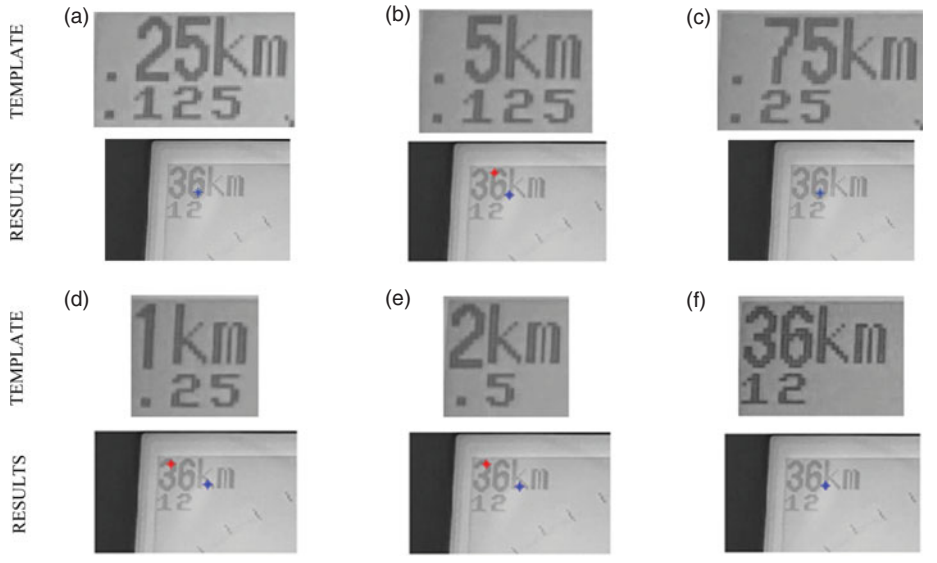


Figure 6. Maximum value obtained with SSD (red) and NCC (blue) on the same image with all the templates (bottom). At top, the templates.

The number of hits increases to 95% if the threshold is reduced to 0·65, which is the average value of the maximum NCC for the eight images that generate false negatives. In any case, the small execution time allows this technique to be used in real time applications.

4.4. *Circle detection methods comparison and discussion of the results.* In this section the results obtained for the whole detection process using the three methods are compared. That is: (1) the Circle Hough Transform; (2) the Three-edge points method; and (3) the new Propagation Method for multiple concentric circles detection based on Circular Regression.

Five representative images from the database out of 100 available images were selected (Figure 7).

From left to right and top to bottom, the images contain: (a) four circles with relevant obstructions in the smallest ones; (b) two circles with some occlusion and different degrees of luminosity; (c) two circles and some noise; (d) four circles with a lot of noise; (e) two circles with half of the image covered by a foldable screen, with the aim of showing that the radar could be manipulated by the user and the algorithm still should be able to recognise the circles. This selection covers all the possible situations in terms of luminosity, noise and obstruction.

The criteria to evaluate and compare the proposal are: the execution time in seconds, the number of detected centres, and the number of valid centres of concentric circles. The same computer was used for all of them: an Intel Core i7-3770k 3·5 GHz with 32 Gb RAM, using Windows 7 Professional and the code was developed using the Matlab image processing toolbox.

Tables 4 to 6 show the results obtained with each one of the proposals. For our novel method (Table 6), different configurations were tested changing some input parameters, from the optimum configuration to a suboptimum one. Only the best results are shown in Table 6 to be compared with the other two algorithms.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press



Figure 7. The five images used to evaluate the algorithms.

Table 4. Results obtained with the Circle Hough Transform.

Circle Hough Transform	Image 1	Image 2	Image 3	Image 4	Image 5
Execution time (s)	1,070	1,069	728	889	909
Candidate centres	44	12	14	18	15
Valid centres	11	8	14	12	15

Table 5. Results obtained with the three-edge points method.

Three-edge Points	Image 1	Image 2	Image 3	Image 4	Image 5
Execution time (s)	1,222	1,038	257	226	335
Candidate centres	1,775	1,477	445	348	303
Valid centres	12	32	87	0	91

Table 6. Results obtained with the new propagation method based on circular regression.

Circular Regression	Image 1	Image 2	Image 3	Image 4	Image 5
Execution time (s)	32	40	32	49	24
Candidate centres	62	80	64	109	52
Valid centres	1	1	2	4	3

Compared with the three-edge point method (Table 5), the Circle Hough Transform (Table 4) is slower, although it reduces the number of valid centres without losing precision in the detection process. Nevertheless, the time both methods require is too high to be considered for real-time solutions.

The new propagation method for multiple concentric circles detection based on Circular Regression presents a significant reduction of the execution time, and it also reduces the number of valid centres to the minimum without losing precision. Therefore, this proposed solution (Table 6) could be used in real time applications.

5. OBSTACLES DETECTION ON RADAR IMAGES: REAL APPLICATION.

The whole procedure has been applied to a real case in order to test it and to calculate the computational time required. This consists of the design and implementation of an alarm system located on a vessel to detect obstacles during navigation. These obstacles are detected in real time with the processing of the images taken from the radar.

5.1. *Materials (radar images).* The experimental setup of this application is as follows. A FURUNO 1715 radar (Furuno, 2018) was used, with frequency 9,410 MHz (X-band), horizontal beam width 5·2°, vertical beam width 25°, range scales 0·125-24 nm, peak output power of 2·2 kW and with a 7" monochrome Liquid Crystal Display (LCD) display unit.

The final system was implemented in a Raspberry Pi 2 model B (Raspberry Pi, 2018) with a Logitech C920 USB webcam (Logitech, 2018) connected (Figure 8, left).

Matlab software was used for the process design but the final system was implemented in the Raspberry Pi using C++ with the OpenCV library (OpenCV, 2018) for image processing. The camera system was mounted in front of the radar display and both were inside

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press



Figure 8. Raspberry Pi 2 model B and Logitech C920 USB webcam.

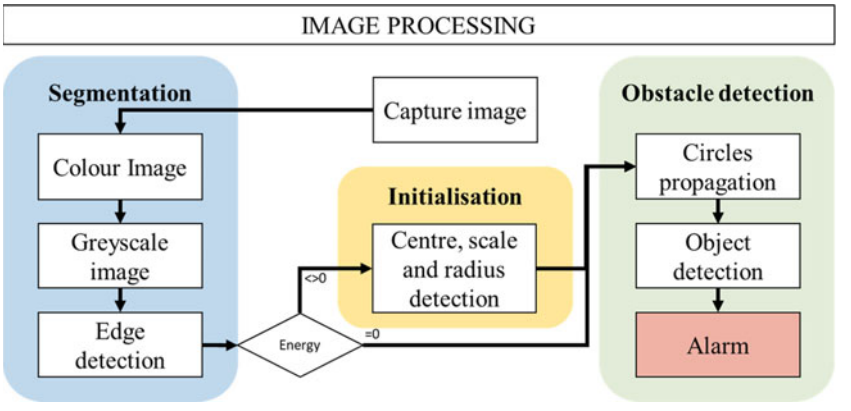


Figure 9. Alarm system processing flow.

a closed (black) box so that the luminosity of the display was kept constant, avoiding shadows and other spurious effects. This box can be readily and quickly moved out of the way when necessary in order not to limit the human interface with the radar display, allowing the operator to use its own interface through the buttons. Due to necessity of visualise the radar display continuously, both the original image captured by the camera and the processed image, are sent to an external display connected to the Raspberry Pi.

The information is captured by the camera, processed, and a hazard map is generated providing information of the possible obstacles. The final set up is shown in Figure 8, right.

5.2. *Image Processing.* The processing is based on three modules: (1) Initialisation; (2) Segmentation, and (3) Obstacles detection. Figure 9 shows the procedure.

The initialisation module calculates the centre (C) and radius of the concentric circles (r) according to the new Propagation Method proposed in this paper. The initialisation process runs until an Energy state = 0 is reached, that is, the difference between consecutive centres is not significant. After that, the scale (S) of the rings is calculated applying the NCC method.

The segmentation module is implemented as explained in Section 3.1.

In the obstacle detection module, the rings and other marks that appear in the edge images are removed according to the data obtained in the initialisation module. Now, only the unidentified objects appear in the image (Figure 10, left). Then, we apply the new propagation technique, using the found centre (red dot) and propagating the circles up to the maximum radius (Figure 10, right).

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

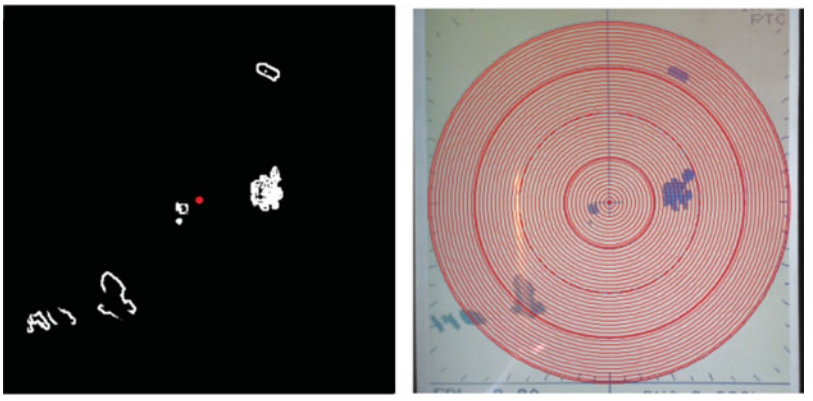


Figure 10. Final image with the unidentified objects and without circles and marks (left); radius propagation with the new technique (right).

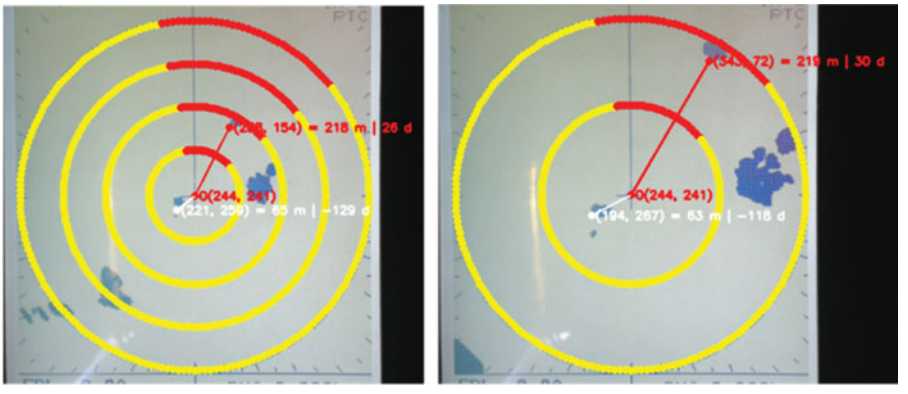


Figure 11. Object detection with different scales. The red dot identifies the nearest object in the cone, and the white dot the nearest object in the whole image. (a) outer radius 1 km (b) outer radius 250 metres.

When a circle's point detects an edge, we calculate the Euclidean distance in pixels and the angle in radians between this point (Px, Py) and the centre of the circles (Cx, Cy). Now, knowing the metric radar scale that has been calculated in the initialisation module and the radius of the outer concentric circle, we can convert the distance from pixels to the metric scale.

Our algorithm has been designed to identify the nearest object both in the whole image and also within a selected cone, so the human operator can select a particular area to get alarms when the closest object appears in that zone (Figure 11).

If the objects are closer than a pre-defined distance, the alarm system is activated, and it can send a message to a mobile phone or tablet connected via Wi-Fi, or by broadcasting a sound.

5.3. *Distance measurement validation.* To evaluate the precision with static objects, we used a cylindrical buoy which was positioned by the Global Positioning System (GPS). Later, a kayak equipped with our vision system and a GPS was used to navigate around this buoy and to take pictures. The GPS coordinates and the images were stored in a Secure

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

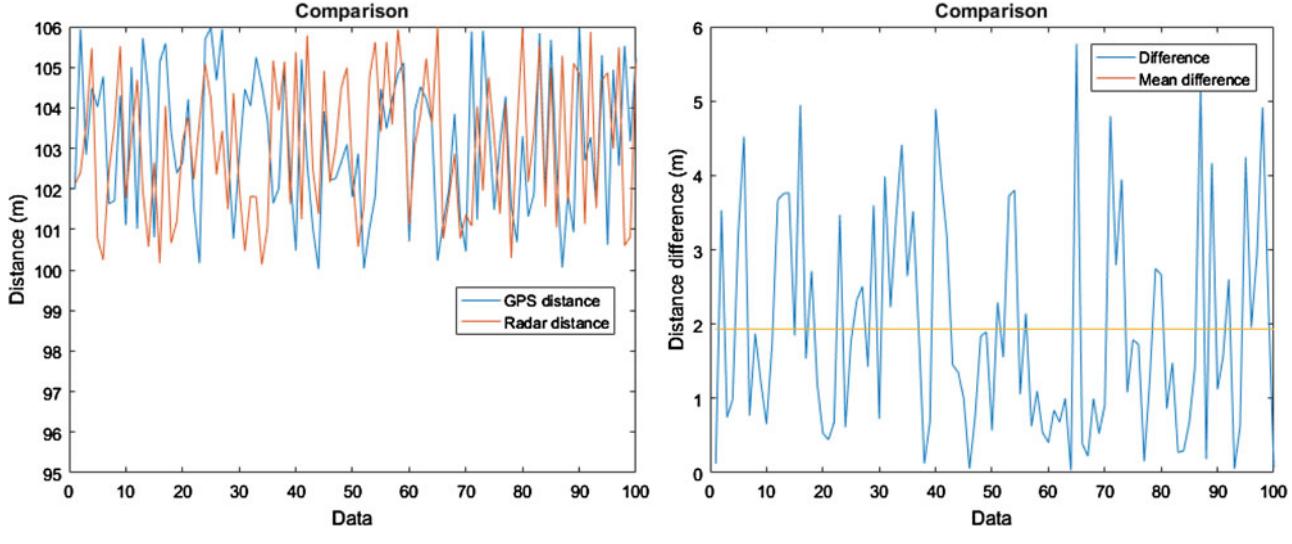


Figure 12. GPS-radar distance comparison.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Table 7. Computational detection time at difference distances.

Group	Outer circle distance	Detection distance	Computational time
1	250 m	100 m	80 ms
2	500 m	400 m	150 ms
3	16 km	8 km	100 ms
4	36 km	30 km	130 ms

Digital (SD) card to measure the distance and to compare the GPS and radar measurements (Figure 12, left). The error in 100 of the captured images can vary up to 5·76 m with an average difference of 1·93 m (Figure 12, right).

This difference is considered valid for navigation purposes. Objects were detected at long distances, further than 10 m. For lower distances, this radar system would not be appropriate.

This technique can be used for different scales. As is shown in Figure 11, the same object is identified within the cone with two different scales. Its distance and angle are similar, in fact they only vary 1 m and 4° respectively from one to another.

5.4. *Processing time.* The processing time varies with the distance of the object. The farther the object is, the longer it takes, because it is a circle propagation technique. In Table 7, objects belonging to four representative classes with different scale rings are shown. Objects in groups 2 and 4 are closer to the outer circle, and objects belonging to group 1 and 3 appear near the middle circles. The scale is selected manually by the operator and the initialisation module configures the radar to detect objects at different distances.

Computation time represents the average time that the object detection requires at that particular distance for 100 of the captured images. All objects belonging to a group are placed at approximately the same distance (detection distance), which differs with the object distances of another group.

The antenna speed for a short pulse has a rotation speed of 41 rpm, so for one complete scanning (360°) it takes 1·5 seconds. In the worst case, the processing time allows up to six frames per second (fps), so this system can detect an object in real-time and it is not necessary to wait for a complete scan.

6. CONCLUSIONS. In this work a novel algorithm for multiple concentric circles is developed. The so-called Propagation Method based on Circular Regression is compared to other traditional circle detection algorithms with good results in terms of efficiency and computational time. The influence of the configuration of the initial parameters has also been analysed.

The method has been applied to radar images. The images of this sensor consist of concentric circles where the detection of the common centre is important to obtain useful information. Indeed, distance to any objects can be obtained applying this new algorithm with high accuracy and in real time. In addition, radar scale recognition is carried out by applying template matching using Normalised Cross Correlation.

Finally, this methodology has been tested in an implemented low-cost alarm system located on a vessel, showing that it is able to detect the obstacles and to send an alarm in real time, as a useful aid to navigation.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

This technique can be applied to any commercial radar device and can also be adapted to other similar problems which require concentric circle detection such as rings' detection in industrial processes, iris recognition, camera calibration, or the identification of bacterial activity or diseases in medical images, among others.

ACKNOWLEDGEMENT

The research was partially funded as part of the SALACOM project (DPI2013-46665-C2-1-R), supported by the Ministry of Economy, Industry and Competitiveness of Spain.

REFERENCES

Akinlar, C., Topal, C. (2013). EDCircles: a real-time circle detector with a false detection control. *Pattern Recognition*, **46**, 725–740.

Almeida, C., Franco, T., Ferreira, H., Martins, A., Santos, R., Almeida, J.M., Carvalho, J., Silva, E. (2009). Radar based collision detection developments on USV ROAZ II. *OCEANS 2009-EUROPE*, Bremen, Germany, 1–6.

Ayala-Ramirez, V., Garcia-Capulin, C., Perez-Garcia, A., Sanchez-Yanez, R. (2006). Circle detection on images using genetic algorithms. *Pattern Recognition Letters*, **27**, 652–657.

Blaich, M., Köhler, S., Schuster, M., Schuchhardt, T., Reuter, J. and Tietz, T. (2015). Mission integrated collision avoidance for USVs using laser range finder. *OCEANS 2015-Genova*, 1–6.

Bole, A., Wall, A. and Norris, A. (2014). *Radar and ARPA Manual: Radar, AIS and Target Tracking for Marine Radar Users*, Butterworth-Heinemann.

Bone, P. (2010). *Hough Transform for circle detection*. http://www.mathworks.com/matlabcentral/fileexchange/9833-hough-transform-for-circle-detection. Accessed 21 December 2018.

Bonin-Font, F., Gomila, C. C. and Codina, G. O. (2017). Towards visual navigation of an autonomous underwater vehicle in areas with Posidonia Oceanica. *Revista Iberoamericana de Automática e Informática Industrial* RIAI, **15**(1), 24–35.

Bovcon, B., Perš, J. and Kristan, M. (2018). Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. *Robotics and Autonomous Systems*, **104**, 1–13.

Briechle, K. and Hanebeck, U. D. (2001). Template matching using fast normalized cross correlation. *Proceedings of Optical Pattern Recognition XII Volume 4387*.

Cao, X. and Deravi, F. (1992). An efficient method for the detection of multiple concentric circles. *Proceedings of ICASSP-92:1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, Vol 3, 137–140.

Chaves, D., Saikia, S., Fernandez-Robles, L., Alegre, E. and Trujillo, M. (2018). A systematic review on object localisation methods in images. *Revista Iberoamericana de Automática e Informatica Industrial*, **15**(3), 231–242.

Chen, X., Lu, L. and Gao, Y. (2012). A new concentric circle detection method based on Hough transform. *Proceedings of 7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, VIC, 753–758.

Cuevas, E., Zaldiwar, D., Perez-Cisneros, M. and Ramirez-Ortegon, M. (2010). Circle detection using discrete differential evolution optimization. *Pattern Analysis Applications*, **14**, 93–107.

Cuevas, E., Osuna, V. and Oliva, D. (2017). *Evolutionary Computation Techniques: A Comparative Perspective*. Springer International Publishing. 35–65.

Frosio, I. and Borghese, N. (2008). Real-time accurate circle fitting with occlusions. *Pattern Recognition*, **41**, 1041–1055.

FURUNO 1715. (2018). https://www.furunousa.com/en/products/1715. Accessed 21 December 2018.

Gupta, S. and Singh, Y. J. (2014). Object detection using shape features. *IEEE International Conference Computational Intelligence and Computing Research*, Coimbatore, 1–4.

Halterman, R. and Brush, M. (2010). Velodyne HDL-64E LIDAR for unmanned surface vehicle obstacle detection. *Proceedings of SPIE–The International Society for Optical Engineering*, Orlando, United States, 76920D-1-8.

Han, J., Koczy, L. and Poston, T. (1993). Fuzzy Hough transform. *Proceedings of the 2nd International Conference on Fuzzy Systems*, San Francisco, CA, vol. 2, pp. 803–808.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press

Heidarsson, H. and Sukhatme, G. (2011). Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. *IEEE International Conference on Robotics and Automation*, Shanghai, pp. 731–736.
Hough, P. (1962). A method and means for recognizing complex patterns. U.S. Patent Office No 306954.
Illingworth, J. and Kittler, J. (1988). A survey of the Hough transform. *Computer vision, graphics, and image processing*, **44**(1), 87–116.
Jiang, G. and Quan, L. (2005). Detection of concentric circles for Camera Calibration. *Proceedings of Tenth IEEE International Conference on Computer Vision (ICCV'05)*, 1, 333–340.
Kanan, C. and Cottrell, G. W. (2012). Color-to-grayscale: does the method matter in image recognition? *PloS ONE*, **7**(1), e29740.
Kapadia, A. S., Chan, W. and Moyé, L. A. (2017). *Mathematical statistics with applications*. Boca Raton: CRC Press.
Lee, S., Kwon, K. and Joh, J. (2004). A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines. *International Journal of Control, Automation and Systems*, **2**(2), 171–181.
Liu, D., Wang, Y., Tang, Z. and Lu, X. (2014). A robust circle detection algorithm based on top-down least-square fitting analysis. *Computers and Electrical Engineering*, **40** 1415–1428.
Logitech. (2018). Logitech C920 Webcam. https://www.logitech.com/es-es/product/hd-pro-webcam-c920. Accessed 21 December 2018.
Muammar, H. and Nixon, M. (1989). Approaches to extending the Hough transform. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing ICASSP-89*, 1556–1559.
Menoyo, J. and Santos, M. (2016). Intelligent rudder control of an unmanned surface vessel. *Expert Systems with Applications*, **55**, 106–117.
Nao, Q., Ye, Y., Mo, C., Wu, Y. and Liu, L. (2010). Method for the Detection of Concentric Circles of Photoelectric Image of Circular Hole in Printed Circuit Board. *Acta Photonica Sinica*, **30**, 75–78.
Oleynikova, E., Lee, N. B., Barry, A. J. and Holler, J. (2010). Perimeter patrol on autonomous surface vehicles using marine radar. *OCEANS*, Sydney, Australia, 1–5.
Onunka, C. and Bright, G. (2010). Autonomous marine craft navigation: On the study of radar obstacle detection. *Proceedings of 11th International Conference on Control Automation Robotics & Vision*, Singapore, 567–572.
OpenCV. (2018). https://opencv.org/. Accessed 21 December 2018.
Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, **9**(1), 62–66.
Pajares, G. and de la Cruz, J.M. (2007). *Visión por Computador: Imágenes Digitales y Aplicaciones*, RA-MA, Madrid.
Polvara, R., Sharma, S., Wan, J., Manning, A. and Sutton, R. (2018). Obstacle Avoidance Approaches for Autonomous Navigation of Unmanned Surface Vehicles. *The Journal of Navigation*, **71**(1), 241–256.
Raspberry Pi Model B. (2018). https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. Accessed 21 December 2018.
Roth, G. and Levine, M. (1994). Geometric primitive extraction using a genetic algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**, 901–905.
Shaked, D., Yaron, O. and Kiryati, N. (1996). Deriving stopping rules for the probabilistic Hough transform by sequential analysis. *Computer Vision Image Understanding*, **63**, 512–526.
Silveira, M. (2005). An algorithm for the detection of multiple concentric circles. *Pattern Recognition and Image Analysis*, **3523**, 143–224.
Xu, L., Oja, E. and Kultanen, P. (1990). A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, **11**, 331–338.
Yang, C., Park, J. and Rashid, A. (2018). An improved method of land masking for synthetic aperture radar-based ship detection. *Journal of Navigation*, **71**(4), 788–804.
Yuen, H. K., Princen, J., Illingworth, J. and Kittler, J. (1990). Comparative study of Hough transform methods for circle finding. *Image and vision computing*, **8**(1), 71–77.
Zhuang, J., Zhang, L., Zhao, S., Cao, J., Wang, B. and Sun, H. (2016). Radar-based collision avoidance for unmanned surface vehicles. *China Ocean Engineering*, **30**(6), 867–883.

https://doi.org/10.1017/S0373463318001169 Published online by Cambridge University Press