

A Circle Detection Task For Performance Evaluation of Edge Detection Algorithms

Victor Ayala-Ramirez, Gerardo Trejo-Caballero, Raul E. Sanchez-Yanez,
Carlos H. Garcia-Capulin and Francisco J. Montecillo-Puente

Universidad de Guanajuato FIMEE
Tampico 912, Col. Bellavista
36730 Salamanca, Mexico

e-mail: {ayalav, trejocg, sanchez, carlos, frajavimopu} @laviria.org

Abstract. In this paper, we propose to use a circle detection task to evaluate the performance of several edge detection algorithms. The purpose of using this kind of task is to measure the algorithm performance when edges are present in any possible orientation. We use a GA based circle detector to ensure sub-pixellic accuracy in the detection of the circle. Edge detection algorithms are applied to synthetic images with a single circle with known parameters on them. We analyze the edge operators response according to three criteria: spurious pixel detection, fitness value criterion and the center position criterion. These criteria are equivalent in some sense to Canny criteria for edge detection optimality. We present the tests and results performed with Canny, Roberts, Sobel, Marr-Hildreth, Prewitt edge detectors and on a custom developed NN-based edge detector.

1 Introduction

Robotic tasks like navigation, localization and reconstruction require computer vision methods [1]. Examples of these methods are object recognition [2], tracking objects and image understanding [3]. Some of these methods require the extraction of features from images like color blobs, texture images or edge information. This paper deals with methods where the main information is detected from edges. There are some widely used standard edge detectors like the ones by Canny, Roberts, Marr-Hildreth, Sobel and Prewitt [4].

In particular, to get a good performance in an object recognition task when using edges as information, we need to select an appropriate edge detector [5]. But, how to select it? We know that the goals of the object recognition task are the detection and localization of the object. In this order, the ideal detector is the one that optimizes the performance of the edge detector based on the following criteria: good detection, good localization and a unique response to a single edge [6]. Even if Canny proposes mathematical definitions for these criteria, we propose to use a specific task that includes implicitly all of these criteria. In this work, we present the evaluation of several edge detection algorithms based on a circle detection task. This task uses edges to detect circles and gives us the circle

center coordinates in the image with sub-pixellic accuracy. The position of the center varies if there are more or less edges and also varies according to where they are located. The comparison is made between the standard algorithms enumerated above and a neural network-based edge detector proposed in [7].

The paper is organized as follows. A description of the edge detectors under comparison is given in Section 2. Our proposed procedure to evaluate the performance of edge detectors algorithms is described in Section 3. A description of the tests and of the results are in Section 4. A discussion of the evaluation and the results is presented in Section 5.

2 Edge Detection Algorithms

Edges are used to represent objects in tracking tasks or object recognition tasks. Among the classical edge detectors, we have chosen the following ones to compare their performance for the circle detection task: Canny, Sobel, Prewitt, Roberts and the Marr-Hildreth edge detectors. Our choices were selected because we can test optimal implementations available in software packages like Matlab. We have also included in our tests a neural-network based edge detector [7]. The purpose of this inclusion is to compare performance when the edge detector is designed by using training samples taken under conditions where most of the classical edge detectors face difficulties to satisfy the criteria specified above.

2.1 Neural Network-based Edge Detector

We employed in this work the neural network-based edge detector that was previously introduced in [7]. For the sake of self-completeness, we describe here this edge detector. Neural networks are useful for systems where we have training examples of the required performance of a system [8], specially when there is an underlying complex model as in the edge detection problem [9] [10]. The neural network-based edge detector uses the intensities of the neighborhood pixels for each pixel of the image (a 3×3 pixel window) as inputs for the neural network to determine if the window center pixel has to be classified as an edge pixel. The architecture of the neural network is a multi-layer perceptron with 9 neurons in the input layer, 6 neurons in the hidden layer and only one neuron in the output layer. The neural network-based edge detector does not need tuning of any parameter because this step is performed during the neural network training phase. As a result of the training step, synaptic weights of the neural networks converge toward the optimal coefficients for the edge detection problem.

3 Circle Detection for Performance Evaluation

In this section we present our proposed procedure to evaluate the performance of edge detectors algorithms based on a circle detection task. Firstly, we present the circle detection algorithm and then we describe the procedure and criteria that give us the information used to select the best algorithm for such a task.

3.1 GA-based circle detector

The circle detection task consists in detecting a circle with known parameters on an input image that is the result of the application of one of the edge detection algorithms under test. The algorithm chosen for the circle detection is a method using genetic algorithms to detect circles [11]. This algorithm features a subpixel accuracy in the determination of circle parameters. We consider this task specially suited for performance evaluation of edge detection algorithms because present edges have orientations in the full range from 0 to 2π radians.

The detected circle is represented by the triad (x_0, y_0, r) with (x_0, y_0) being the center of the circle in image coordinates and r is the radius of the circle also in pixels. Ground truth images used for input to edge detectors were constructed by software, that is, pixels were drawn for a circular region by considering if they satisfy the locus of a circle.

$$I(x, y) = \begin{cases} 1 & \text{if } \sqrt{(x - x_0)^2 + (y - y_0)^2} \leq R \\ 0 & \text{elsewhere} \end{cases}$$

The fitness function is a measure that relates the number of expected edge pixels and the pixels that actually are present in the edge image. A fitness value for the best circle is provided at each run of the GA-based circle detector. Good detection and good localization criteria are satisfied when the actual circle coordinates obtained using the edge image from the edge detection phase result are very close to the ground truth parameters of the synthetic images. Uniqueness of the response to edges is correlated to the fitness value of the detected circle. A greater value for the fitness function of the detected circle implies that edge points are unique in the detected circle.

The GA-based circle detector was evolved until there were no changes in fitness above a 0.01 % of the total number of edge points of the ideal circle.

3.2 Procedure to Evaluate Edge Detection Algorithms Performance Using a Circle Detection Task

For the procedure, we need a synthetic image including a circle with known center coordinates and radius. In Fig. 1, we show four of the test images. We apply then the edge detector under test to each of these images and save the center coordinates of the best circle detected and also the corresponding fitness function value. These results are then compared to find the optimal edge detector for the circle detection task.

For all the tests, the parameters employed for the circle detector algorithm are shown in Table 1.

4 Tests and Results

To evaluate performance of edge detection algorithms, we use three criteria. The first criterion concerns spurious edge pixels, that is, the detected edge pixels not

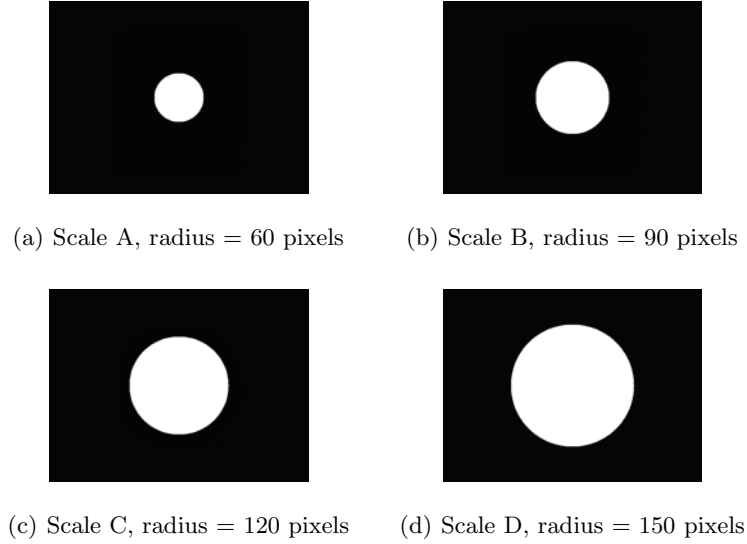


Fig. 1. Four test images, they present the same center, to know (318.5, 238.5) but different radii (60, 90, 120 and 150 pixels respectively).

Table 1. Parameters of the circle detector

Generations	Population size	Crossover probability	Mutation probability	Elite individuals
50	1000	0.80	0.10	2

belonging to the detected circle. The second criterion is the fitness value of the best detected circle. The last criterion concerns accuracy of the position of the center of the detected circle.

In Fig. 2, we show the edges extracted by the edge detection algorithms under test. By using only visual inspection, it is difficult to find differences in their outputs.

4.1 Spurious Pixels Criterion

Ground truth data for the spurious pixels criterion is obtained by generating a circle using the parameters found by the GA circle detector. We define spurious pixels as the ones that are present in an edge image but they are not part of the edge ground truth image. Table 2 shows the total number of edge points in the image, the number of detected edges not belonging to the ground truth image and the ratio of these quantities. Values in this table are the average of 30 tests. The lower the spurious pixel ratio, the best we can rate the performance

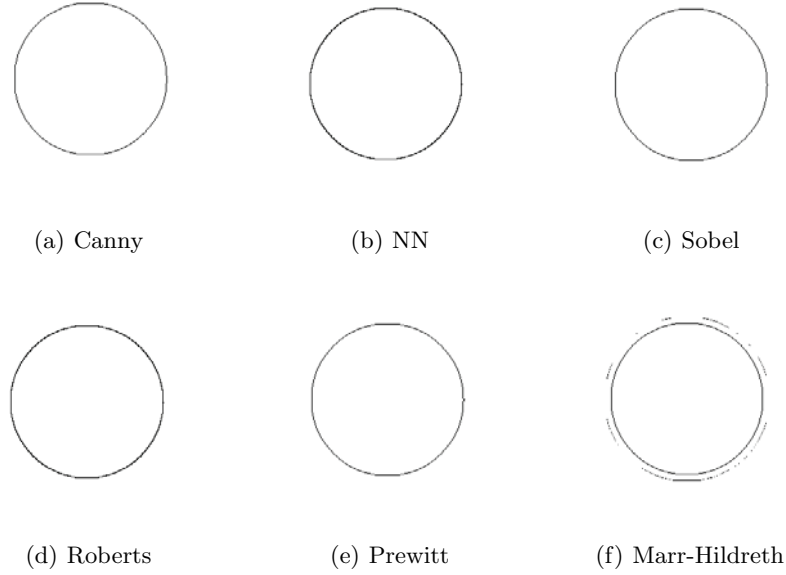


Fig. 2. Output edge images for the test image Scale B (radius = 90 pixels) obtained by the application of six edge detectors under test.

of the edge detector. In our tests, we have found that Canny is the best detector according to this criterion, followed in the second place by the NN detector.

Table 2. Spurious and detected pixels in the edge image.

Edge Detector	Detected Pixels	Spurious Pixels	Ratio
Sobel	282	20	0.070
Canny	284	6	0.021
NN	404	17	0.042
Roberts	404	18	0.044

4.2 Fitness Value Criterion

The fitness function value gives the fraction of pixels present both in the output edge image and in the ground truth image. It is used as a indicator of the correct response and the good localization of the edge points. This value will tend to 1 when the edge detector provides a good localized response.

We show the behavior of the fitness function value for all the detectors under test at scale B ($r = 90$ pixels) in Fig. 3. Best performance is obtained by the Roberts edge detector. The second place is for the NN based edge detector.

We also show the behavior of the fitness function value through the different scales for each edge detection algorithm under test. For all of them, we have run 30 tests at each scale. We can see the behavior for Canny, Roberts and the NN based edge detectors in Fig. 4 (a), (b) and (c) respectively. We can see that Roberts edge detector decreases its performance for greater scales (C and D). In the other hand, NN based detector shows a good performance at all scales. We conclude that for this criterion, the NN based edge detector is the best one. Canny detector has shown to be the worst edge detector for all scales That is explained by the fact that it detects a lower number of edges than the other methods.

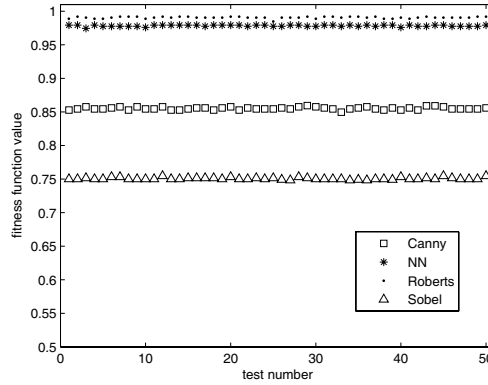
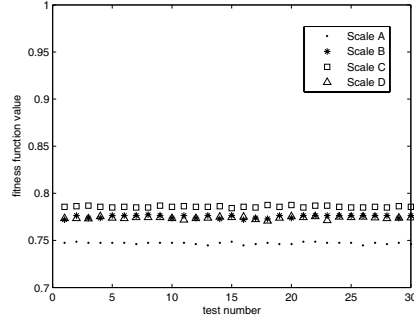


Fig. 3. The fitness function value of the detected circles shown for scale B.

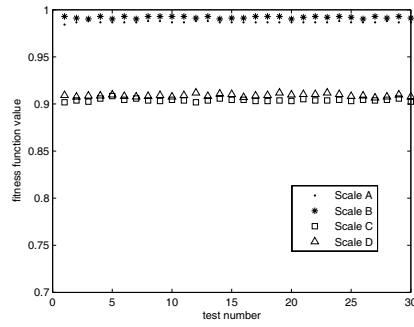
4.3 Circle Center Criterion

Another test we have performed to evaluate edge detectors was to compare the center position of the detected circle in the output edge image against the known center of the circle. The ground truth center is (318.5, 238.5) for test images at all scales.

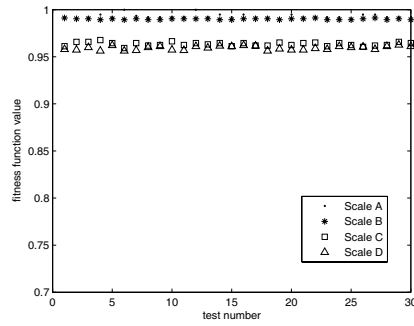
We show in Fig. 5 (a), (b) and (c) the position of the center of the detected circle for Canny, Roberts and NN edge detectors, respectively. We have run 30 tests at each scale. Graphs share the same axis limits to ease comparison. We can conclude that the NN edge detector presents a smaller bias from the ground truth circle center. According to this criterion, Canny detector comes in second place.



(a) Canny



(b) Roberts

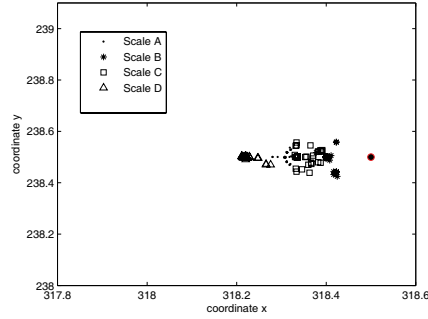


(c) NN

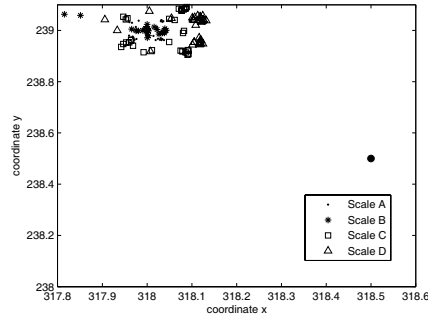
Fig. 4. Fitness function values found for the Canny, Roberts and NN algorithms at different scales.

5 Conclusions and Perspectives

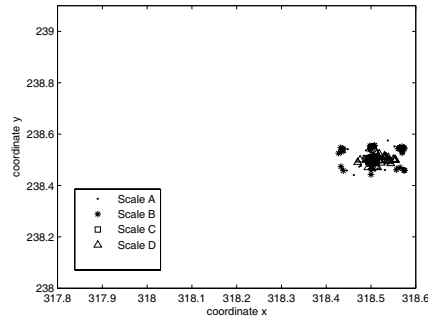
The results of our tests are only significant for a task like the one used here. Detecting circles and ellipses could be useful in some camera calibration procedures where an edge detection step is needed.



(a) Canny



(b) Roberts



(c) NN

Fig. 5. Bias of the center position for the circles detected on outputs from the Canny, Roberts and NN-based algorithms.

We have presented three criteria where a circle detector may be used to evaluate the performance of several edge detectors. There is no an only winner edge detector for all tests. Additionally we have compared a custom NN edge detector

and the results show that its performance is good enough as the classical ones. Tests show the dependence on scale of the classical edge detector performance. However, the custom NN edge detector maintains its performance through the different scales used for testing.

As a future work, we plan to evaluate the performance on real-time systems that uses edges as main information, in order to face real problems like illumination and noise.

Acknowledgements

This work has been partially funded by the University of Guanajuato project “Funcionalidades visuales para la navegación topológica de robots móviles” and the Concyteg project “Herramientas Mecatrónicas para la Implementación de Entornos Virtuales”.

References

1. Souza, G.D., Kak, A.C.: Vision for mobile robot navigation: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24** (2002) 237–267
2. Hunter, I., Soraghan, J.J., McDonagh, T.: Fully automatic left ventricular boundary extraction in echocardiographic images. In: *Proc. Computers in Cardiology’95*. (1995) 741–744
3. Tabb, M., Ahuja, N.: Multiscale image segmentation by integrated edge and region detection. *IEEE Transactions On Image Processing* **6** (1997) 642–655
4. Parker, J.R.: *Algorithms for image processing and computer vision*. John Wiley and Sons (1997)
5. Heath, M.D., Sarkar, S., Sanocki, T., W.Bowyer, K.: A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19** (1997) 1338–1359
6. Canny, J.: A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **8** (1986) 679–698
7. Trejo-Caballero, G., Ayala-Ramirez, V., Perez-Garcia, A., Sanchez-Yanez, R.E.: A neural network approach to edge detection in real images. In: *Proc. RFMPIM 2005*. (2005)
8. Lippmann, R.P.: An introduction to computing with neural nets. *IEEE ASSP Magazine* (1987) 4–25
9. Kulkarni, A.D.: *Computer Vision and Fuzzy-Neural Systems*. Prentice-Hall, Upper Saddle River, NJ (2001)
10. Perry, S.W., Wong, H.S., Guan, L.: *Adaptive Image Processing: A Computational Intelligence Perspective*. CRC PRESS, Boca Raton, Florida (2002)
11. Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E.: Circle detection on images using genetic algorithms. *Pattern Recognition Letters* **27** (2006) 652–657