ROBUST CIRCLE DETECTION USING A WEIGHTED MSE ESTIMATOR

Guido M. Schuster

Abteilung Elektrotechnik
Hochschule Rapperswil, Switzerland
guido.schuster@hsr.ch

Aggelos K. Katsaggelos

Department of Electrical and Computer Engineering
Northwestern University, Evanston, Illinois, USA
aggk@ece.northwestern.edu

ABSTRACT

In this paper we introduce a novel circle detection algorithm based on a weighted minimum mean square error (MSE) formulation. Traditional approaches to circle detections consist of two stages, an edge detection stage and a circle detection stage using the edge detection result. There are several problems with this approach. First, the initial edge detection stage is sensitive to noise. Second, the second stage does not use all the information available in the image and therefore incorrect decisions made by the first stage cannot be corrected in the second stage. The proposed algorithm achieves its robustness by operating in one step, using all pixels of the image (correctly weighted) and not using any thresholds. The detected circle is the solution of several weighted MSE problems. Experimental results demonstrate the performance of the algorithm in noiseless and noisy conditions.

1. INTRODUCTION

The goal of a circle detection algorithm is to find in a given grayscale image I a particular circle of interest, as shown, for example, in figure 1. One powerful approach for circle detection is the Hough transform and its variants [1]. One well known drawback of such approaches is the computational complexity, as basically every possible circle needs to be tested. Another approach is to use an edge image E to find the circle, since the circle creates two strong edges as can be seen, for example, in figure 2 which shows the squared norm of the gradient of a Gaussian (σ=1) lowpass filtered image of figure 1. It is common to first find all the edges in the image, using an edge detection algorithm and then analyze the edge image to find the particular circle. The main problems with this approach are that circle detection algorithms are sensitive to noise and the detection of the circle is now a two step process. If an error is made during the edge detection stage, it cannot be corrected in the circle detection stage. Furthermore, most edge detection algorithms require one or more thresholds for selecting which points belong to edges and which do not. Whenever a threshold is required, finding a robust value which works under all circumstances is a major problem which typically cannot be solved in a satisfactory way. The proposed algorithm does not have these problems; instead it finds the parameters of a circle model directly as the solutions of weighted MSE problems.

The paper is organized as follows. In section 2 we introduce notation and assumptions necessary for the remainder of the paper and a parametric model for a circle and present a circle detection method for noiseless images. In section 3 we derive the weighted MSE solutions to the circle detection problem in noisy images. In section 4 we propose the weighting matrices which are meaningful in our application. In section 5 we show results of the proposed algorithm and section 6 concludes the paper.

2. CIRCLE MODEL AND NOISELESS DETECTION

The fundamental assumption in this paper is that there is only one circle in the image. Since the algorithm is based on a weighted MSE approach, it would select a circle as the solution that is the weighted average of multiple circles, which would be incorrect. We further assume that the grayscale image I is of dimensions $R \times C$, where R is the maximum number of rows and C the maximum number of columns. Furthermore we use the notation (row, column) throughout the paper and the origin $(0, 0)$ is in the middle of the image.

The main purpose of the circle model is to incorporate our knowledge about the existence of a circle into the algorithm. Furthermore, such a mathematical description of the circle is a more useful abstraction for a higher level vision algorithm than a simple pixel based description. Figure 3 shows the circle model used in this paper. We assume a ray originating at the middle point of the image at an angle α. The angle is varied from 0 to 2π and hence the ray is covering the entire image. Given a particular circle, which is defined by a center $(z_r = z * \sin(\gamma), z_c = z * \cos(\gamma))$ and a radius r, the length of the ray which is tracing the circle is $a(\alpha)$.

If we assume that we can measure $a(\alpha) * \cos(\alpha)$ and/or $a(\alpha) * \sin(\alpha)$ within pixel accuracy, then we can directly estimate the circle parameters z_c, z_r and r from these measurements. If we consider the horizontal dimension first, the following holds, $a(\alpha) * \cos(\alpha) = z_c + r * \cos(\beta(\alpha))$. Note that while $\beta(\alpha)$ depends on α, z_c does not. We remove the $r * \cos(\beta(\alpha))$ term by taking the average over all pixels that belong to the circle (we indicate this operator by a line above the expression). Since for every pixel on the circle there is a pixel diametrically opposite to it where the $\cos(\beta(\alpha))$ term has the same magnitude but opposite sign, the following holds: $\overline{a(\alpha) * \cos(\alpha)} = z_c$. A similar expression holds in the vertical dimension, that is, $\overline{a(\alpha) * \sin(\alpha)} = z_r$.

If the image is noiseless as shown in figure 1, then we can identify the pixels that belong to the circle using a threshold. Having done that, the above averages which result in an estimate of the circle center, (z_r, z_c), can be calculated. In the next step we estimate the radius of the circle by calculating the average distance between the estimated circle center and all circle pixels.

While this approach is much faster than a Hough transform, in this form it requires that the pixels belonging to the circle can be easily detected, which is only possible in a noiseless situation. Nevertheless, this approach only requires the calculation of averages to estimate the circle parameters. In the next section we address the case of noisy data and hence the solution will not be based on the fact that the circle pixels have been selected by a threshold, but the detection of the circle pixels will be implicitly performed using a weighted MSE solution similar to the line detection approach proposed in [2]. Another way to combine the pixel detection and the shape estimation is by following a snake based approach [3] and its more recent variants, such as deformable templates [4]. These approaches have the problem that they are not globally optimal but depend on the initial conditions. As we will show in section 3 the proposed solution does not require any initial conditions.

3. WEIGHTED MSE SOLUTION

We use the well known concept of MSE estimation via the generalized inverse (also known as pseudo-inverse or Moore-Penrose-inverse) of a matrix M which we denote by $(M)^{\dagger}$. Assume that we have the equation $y = Ax$ where y is an $N \times 1$ vector, $x = [x_1, x_2, \ldots, x_{o-1}, x_o]^T$ is the model parameter vector of dimensions $o \times 1$ and A is a matrix of dimensions $N \times o$ (Note that we will later show that for the problem at hand the order o is equal to 1, but we keep this discussion general.) Then the MSE solution to this problem is: $\hat{x} = (A^T A)^{\dagger} A^T y$. In other words, the total quadratic error, $e^2 = (y - A\hat{x})^T (y - A\hat{x})$, is as small as possible for the selected $\hat{x}$.

Since we would like to be able to weight the individual components of the quadratic error, we introduce an invertible diagonal weight matrix W of dimensions $N \times N$. Hence we are interested in the solution that minimizes the weighted quadratic error $e_w^2 = (y - A\hat{x})^T W^T W (y - A\hat{x}) = (W(y - A\hat{x}))^T (W(y - A\hat{x}))$. Note that this is an equivalent problem to minimizing the mean squared error of the equation: $y_w = A_w x_w$, where $y_w = Wy$, $A_w = WA$ and x_w is the parameter vector that needs to be estimated. The solution to this problem is now $\hat{x}_w = (A_w^T A_w)^{\dagger} A_w^T y_w = (A^T W^T W A)^{\dagger} A^T W^T W y$. Now that we have a circle model and a method for finding the optimal solution to a weighted MSE problem, we need to set up three estimation problems, one for each parameter.

4. WEIGHTING SELECTION

The basic idea of this algorithm is to use all available pixels for detecting a circle that is given by a circle model. In other words, no pixels are eliminated *à priori* but all of them are involved in finding the proper estimate. Clearly some pixels are more important than others and hence we use the weighted MSE scheme discussed in section 3 for estimating the parameters. In section 2 we have discussed the noiseless case, where the circle parameters are estimated using a two step approach. First the circle pixels are detected using a threshold, and then the parameters are estimated using averages over only those pixels.

If we consider the horizontal dimension first, then we would like to calculate the average over all circle pixels $\overline{a(\alpha) * \cos(\alpha)} = z_c$ for finding the horizontal coordinate of the circle center. This equation must now hold for all circle pixels in the image. Since the circle pixels are not known in advance (they are the objective of the estimation process), the above equation is applied to all pixels in the image after they are appropriately weighted. Pixels that have a high intensity value are more likely to belong to the circle, hence their weights should be high. In other words, the weight for a pixel (r_i, c_i) should be a monotonically increasing function of its intensity value $I(r_i, c_i)$. Furthermore, in the edge image E, which is in our case the norm of the gradient of a Gaussian ($\sigma = 1$) lowpass filtered image, high values belong to pixels that are on edges. Clearly the circle creates two edges, but the weighted MSE solution results in a point located in the middle in the horizontal direction of these two edges, since at this point the average squared error is kept the smallest. Hence the weight of a pixel at (r_i, c_i) should also be a monotonically increasing function of the value in the edge image $E(r_i, c_i)$. To make sure that all these weights are of the same magnitude, we normalize the intensity image, denoted by $\tilde{I}$, such that its standard deviation is 1 and its minimum value equal to 0. We do the same for the edge image and call it $\tilde{E}$. We experimentally selected the quartic function ($F(g) = g^4$) as the monotonic function since it performed slightly better than the classic quadratic function.

Now one row i of the set of equations $Wy = WAx$ (there are $N = RC$ rows of them) has the form: $w_i * c_i = w_i * x_1$, where $w_i = \tilde{I}(r_i, c_i) * \tilde{E}(r_i, c_i)$. In other words, the diagonal matrix W consists of the above weights, the column vector y consists of the column indices of all points in the image I, the A matrix is simply a column vector filled with ones and the parameter vector x contains only one parameter x_1 and the resulting $\hat{x}_1$ is the best estimate for z_c. Note that the error for a particular parameter vector $\hat{x}$ is now equal to $e_i(\hat{x}) = w_i * (c_i - \hat{x}_1)$, which illustrates the idea behind weighting the error of the pixels differently.

After the horizontal coordinate z_c of the circle center has been estimated, we do the same for the vertical coordinate of the circle center. Here the role of the rows and the columns in the above equations are swapped. After these two steps, an estimate of the circle center is obtained. This estimate is now used for estimating the radius of the circle. For every point in the image (r_i, c_i) we can now calculate the distance to the estimated circle center. Again, for each point in the image we weight the error as described above but in addition we also divide the weight by this distance. This results in the fact that points with large radii count proportionally less than points with small radii. Since there are proportionally more points with large radii (the circumference grows linear with the radius) than with small radii, this weighting removes a systematic estimation bias towards large radii.

With the proposed weighting of the intensity image and the edge image a solution is favored where the circle parameters result in circles that are attracted to bright areas and strong edges. Furthermore, the inverse weighting of the radii allows for the removal of an implicit bias in the estimation process. It is important to note that the proposed algorithm is quite fast, since every pixel is only used once. This is in contrast to a Hough transform based solution or a correlation based solution.

5. EXPERIMENTAL RESULTS

We have performed a series of experiments where we visually compare the performance of an automatic Canny edge detector [5] (the first step of a two step line detection process) and our proposed algorithm. First we show the performance in a noise-free environment and then with additive white Gaussian noise. Figure 4 shows the result of the automatic Canny edge detector implemented in the MATLAB image processing toolbox. Figure 5 shows the same image processed by the proposed algorithm. Clearly the result is at least as good as the Canny edge detector and in addition the proposed algorithm results in a parametric description of the circle and not only in an edge image. Furthermore, in the MATLAB implementation of both algorithms our algorithm is about an order of magnitude faster. For the next experiment we added zero mean additive white Gaussian noise (AWGN) to the image resulting in a signal to noise ratio (SNR) of -6dB. The resulting grayscale image can be seen in figure 6, the squared norm of the gradient of a Gaussian ($\sigma = 1$) lowpass filtered version of this noisy image can be seen in figure 7 and figure 8 shows the result of the Canny edge detector. Figure 9 shows the image of figure 6 processed by the proposed algorithm. Clearly this is still a good result even though the SNR is so low that the Canny edge detector basically fails.

6. SUMMARY AND CONCLUSIONS

In this paper we have presented a robust circle detection algorithm based on a weighted MSE formulation. One of the features of the algorithm is that it does not require any thresholds, but finds a solution as the optimal tradeoff between many different sources of information and constraints. The main constraint is the circle model which represents our prior knowledge about the shape. The information sources are the intensity of the image and the edge image (but other ones can be added easily, depending on the particular application). The fusion of this information is achieved via a weighted MSE formulation. The experimental results show that the proposed algorithm is quite robust and significantly faster than the Canny edge detector. Most notably in a noisy environment the proposed algorithm can still reliably estimate the location and size of the circle.

7. REFERENCES

[1] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Computer Vision, Graphics and Image Processing*, vol. 44, pp. 87–116, 1988.

[2] G. M. Schuster and A. K. Katsaggelos, "Robust line detection using a weighted MSE estimator," in *Proceedings of the International Conference on Image Processing*, Barcelona, Spain, September 2003.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.

[4] A.K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 3, pp. 267–278, 1996.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, November 1986.

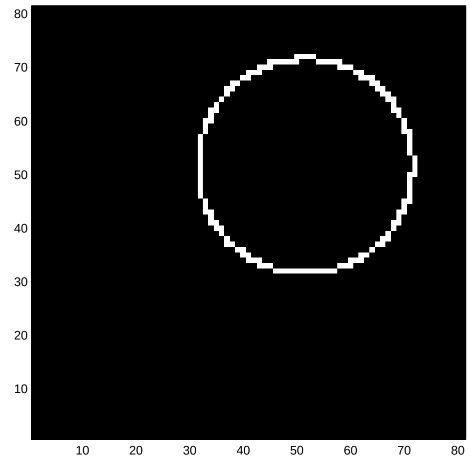

Fig. 1. Original grayscale image

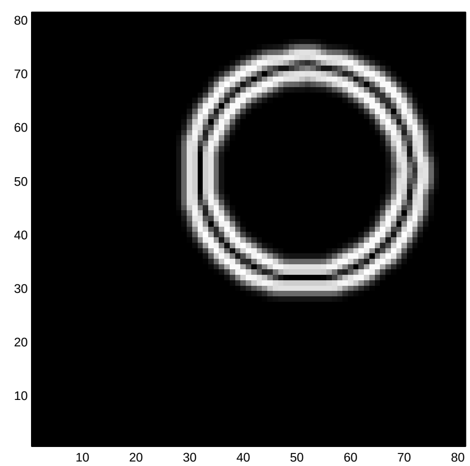

Fig. 2. Squared norm of the gradient of a Gaussian ($\sigma = 1$) lowpass filtered image shown in figure 1

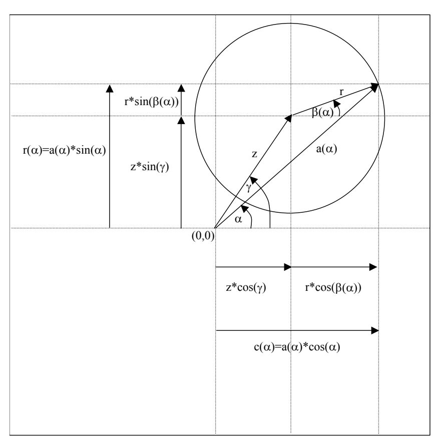

Fig. 3. Definition of the circle model

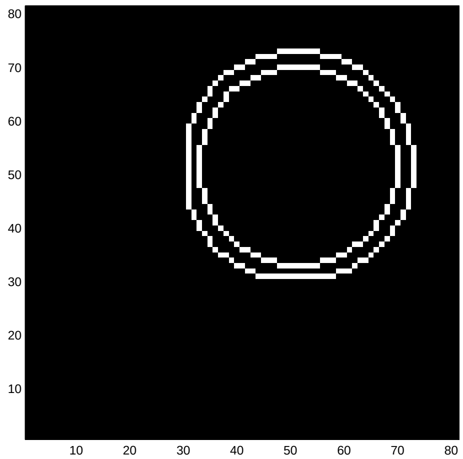

Fig. 4. Canny edge detector processing the image in figure 1

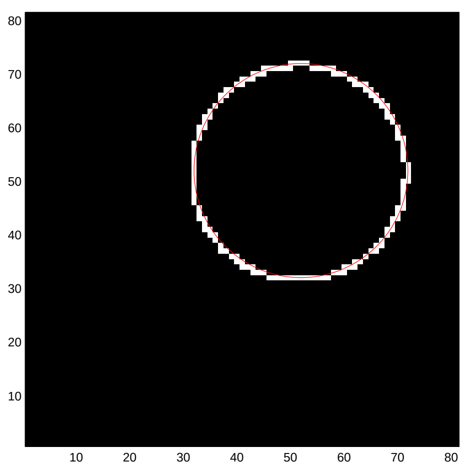

Fig. 5. Detected circle in figure 1 using the proposed algorithm

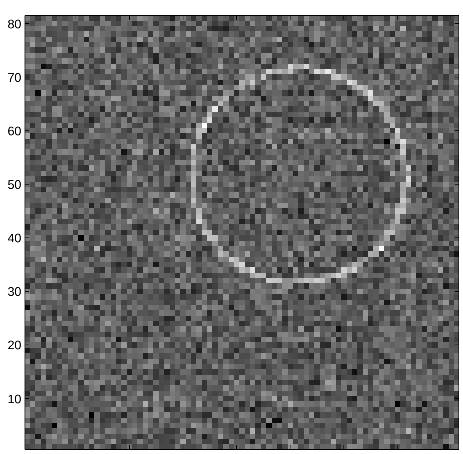

Fig. 6. Original grayscale image with AWGN @ SNR=-6dB

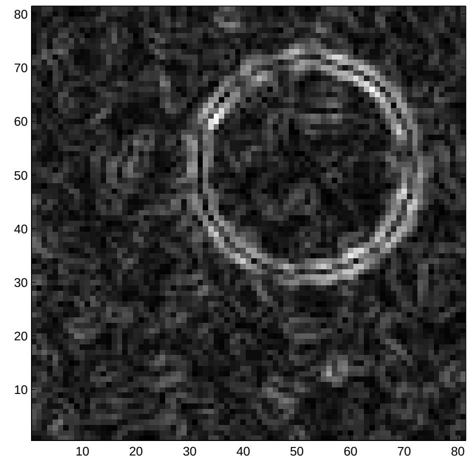

Fig. 7. Squared norm of the gradient of a Gaussian ($\sigma = 1$) low-pass filtered image shown in figure 6

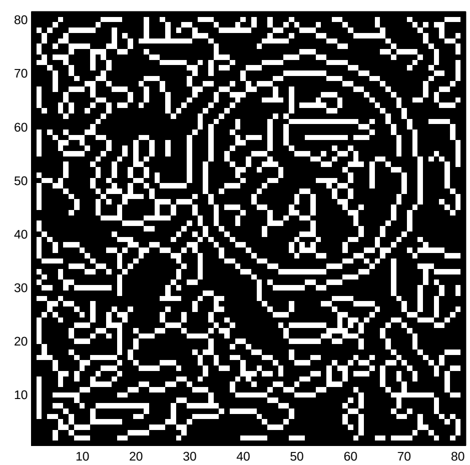

Fig. 8. Canny edge detector processing the image in figure 6

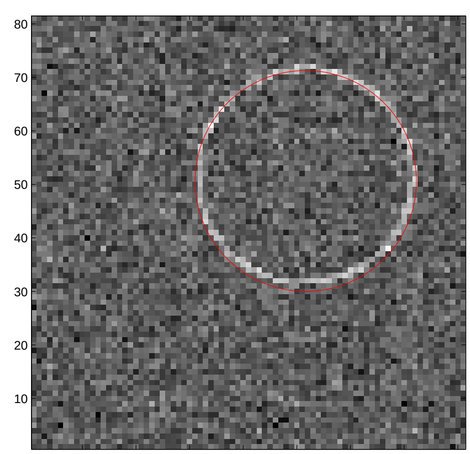

Fig. 9. Detected circle in figure 6 using the proposed algorithm