Detection of multiple circles based on adaptive Hough transforms

X. Cao, M. G. Rodd, F. Deravi and Q. M. Wu
Image Processing Laboratory, Department of Electrical and Electronic Engineering, University College of Swansea, Singleton Park, Swansea SA2 8PP, UK
(*Received April 1988*)

ABSTRACT

The time and storage requirements for the general Hough transform rise exponentially with increases in the resolution of the image and the parameter space. A recent attempt at decreasing these requirements is the adaptive Hough transform, AHT[1]. But when used to detect images containing more than one shape, the AHT is inaccurate in the determination of the 'best peak' locations in the parameter space, and is also vulnerable to clusters of noise. Because all the features in the image space are mapped into the parameter space, the algorithm is computationally intensive. This paper shows how a 'labelling' method can be combined with the AHT to solve those problems. Its parallel implementation allows for successful detection of multiple circles, such as those found in many practical industrial applications.

INTRODUCTION

The original Hough transform was proposed by Hough[2] for detecting straight lines in an input image. It has since been improved by Duda and Hart, to detect more general classes of curves, such as circles[3], and then generalized by Ballard[4] to include arbitrary shapes.

The Hough transform, as a method of detecting specific structural relationships between pixels in an image, using a parameter space, has important advantages in resisting many kinds of noise such as missing, false, or misplaced image points, and has excellent recognition ability.

The earlier versions of the generalized Hough transform map features in an image space (x, y) into a parameter space (a, b) which is finely quantized into so-called 'accumulator cells', and identify the significant local maxima of mapped points (e.g. the intersections of many lines) among the accumulator cells. The accuracy of individual parameters depends on the fineness of the quantization where a high degree of accuracy is required. The resulting drawbacks of long computation times and large memory requirements make this method unsuitable for real-time applications.

Recent literature on the Hough transform (HT) includes work related to the reduction of the storage and computational requirements. For example, for the AHT, Illingworth *et al.* suggest an intelligent iterative 'coarse-to-fine' accumulation-and-search strategy. The method consists of accumulating the basic HT in a small fixed-size accumulator, and then identifying significant peaks. The main steps are:

1) Initially resolve each parameter only very coarsely.
2) Accumulate the basic HT in a small fixed-size accumulator.
3) Identify significant peaks in the accumulator.
4) Use the information on the 'best peak' as a basis on which to redefine the parameter range.

As a result, by moving from coarser levels to finer levels, parameters are determined to a prespecified accuracy. Approximate calculation shows that the AHT is potentially 5 times faster than the standard HT, using an accumulator $(9*9)$ which is some 3000 times smaller. The spatially-random-point noise has little effect on the identification of parameters[1].

The performance of the AHT was tested using both digital line and circle segments, but not both in the same image. If an image does contain multiple objects, Illingworth and Kittler suggest the following procedure:

0952–1976/88/020097–05$2.00
© 1988 Pineridge Ltd

1) Use AHT to identify points belonging to an instance of an object.
2) Remove this instance of an object from the image space.
3) Reinitialize the search at the coarsest resolution and search for another object. This process continues until the parameter space contains no significant structures.

We have identified the following disadvantages of this method:

When an image contains multiple objects, the mapped points in the parameter space will affect each other, causing shifts in the locations of the best peaks, and making it difficult to select an appropriate threshold to determine the location of the best peak of the other object. In addition, this sequential process of determining each object in turn is time-consuming and may not meet practical industrial needs. In AHT, correct identification of high-density regions in the parameter space may be influenced by spurious data. If the spurious data results from random noise, it has little effect on the parameters, but if the spurious data occurs as a cluster of noise, it may heavily affect the parameters, and may even be mistaken for an expected object. This will be discussed in detail in the next section.

In this paper, we discuss the AHT method from the viewpoint of an industrial application, and combine it with a 'labelling' method (LAHT) to detect multiple circles in an image. We attempt to solve the problems arising from the interactions between circles, and the disturbances caused by noise. Furthermore, because we can use this method to detect each circle in parallel with all the others, each circle can be handled in isolation, unaffected by the others.

The LAHT method and its parallel implementation will be described in detail in the later sections. A two-circle detection problem will be given as an example of parallel processing.

USE OF THE ORIGINAL AHT IN DETECTING MULTIPLE CIRCLES

For the AHT algorithm, the parameters selected are the coordinates of the centre of each circle. When an image contains only one circle (see *Figure 1*), the parameter space tends to be symmetrically populated about the true point of intersection (*Figure 2*). Therefore, it is comparatively easy to find the thresholded peak.

However, when an image contains more than one circle (see *Figure 3*), because of the interactions between these circles, the character of the symmetrical populations about the true points of intersection is not clear. *Figure 4* illustrates the

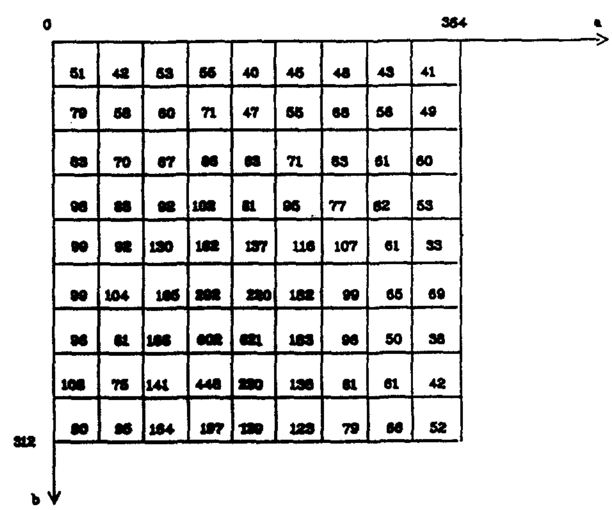

Figure 2 Accumulation in the parameter space in the first iteration in detecting one circle

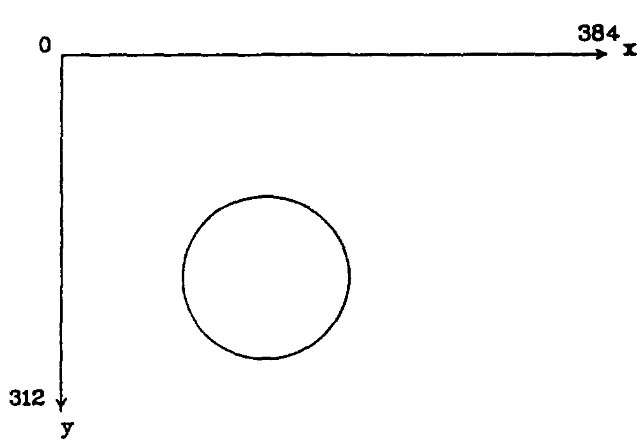

Figure 1 Detection of a circle in an image

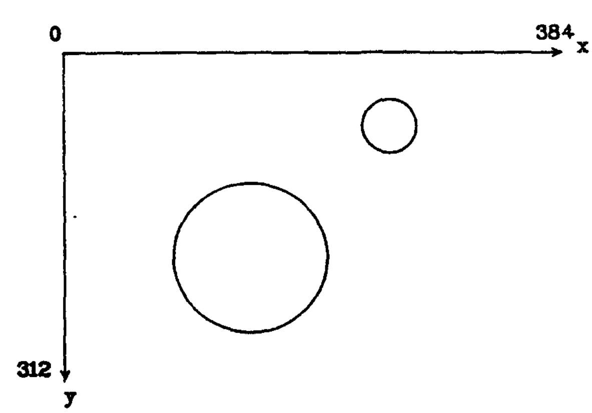

Figure 3 Detection of two circles in an image

0 — 384 a

60	72	57	98	65	60	75	60	77
101	88	80	127	79	96	74	106	47
137	109	96	151	171	135	116	82	79
133	152	129	171	245	142	148	99	83
137	181	251	236	262	325	133	88	71
85	75	267	292	308	181	110	74	54
120	84	236	253	324	103	69	56	51
91	81	263	331	388	179	70	65	43
92	92	401	102	187	213	106	67	51

312 b

Figure 4 Accumulation in the parameter space in the first iteration in detecting two circles

accumulation resulting from applying AHT to *Figure 3* in the first iteration.

The number 401 is the highest count, but the counts around it are not symmetrically populated. If the threshold is taken as 0.9 of the highest count, both 401 and 388 are greater than 306.9, and are therefore both 'significant peaks'. However, because they are not compact, there are two separate significant peaks. It is therefore difficult to locate intelligently the true point of intersection. The threshold may be increased to leave only the count 401 as the best peak, but in fact, by simply analysing the original image in *Figure 3*, we can see that the best peak should be around the count 388. It is the interactions that have shifted the apparent location of the best peak away from its true position.

In AHT, all the features in the image space are mapped to the parameter space to allow 'voting' in the accumulator, so that when an image contains spatially-random noise, these noises are also regarded as features. It is clear that these 'false' features will affect the results of accumulation in the parameter space to a greater or lesser degree, and will consume some time. Another time-consuming problem arises from the sequential nature of the process employed.

METHOD FOR DETECTING MULTIPLE CIRCLES

A specific application of computer vision lies in the inspection of sheet metal punching, usually involving many holes in any one sheet. Once the desired patterns have been designed on a CAD package and the sheet metal has been processed, it is desirable to check the quality of the resulting material automatically, using visual techniques.

This kind of image containing many holes can be considered as a binary image, the holes represented by black colour and the metal sheet by white. Before attempting to determine the centre and radius of the circle which defines the boundary of each hole, we make use of both edge and region data in the image to label all holes. Each apparent 'hole' (including any noise) is given a different label.

The labelling algorithm can be used to segment the image and assign labels to holes[5]. The highest-value label number represents the biggest hole, the second highest-valued label number the second biggest hole, and so on. Therefore, we can identify the multiplicity of holes by the various labels. When the image contains some spatially-random noise, each instance of noise also has its own label, with a label-number-value much smaller than those of the holes. If the image contains a cluster of noise, each component of the cluster may have the same label, but the value of that label number is still less than those of the holes. As a result, we can distinguish between holes and noise, and also between different holes, according to the values of their labels. From the highest label number to the lowest label number, we can, in turn, identify each hole, from the biggest to the smallest.

By making use of the labels of the different holes, the whole image can be coarsely divided into different ranges, each different location range containing only one circle. Instead of mapping all the features in the image space to the parameter space, we map just the features within each location range in turn. Once the coarse location ranges of the circles have been determined, the coordinates of the circle centres must lie within those ranges. So, instead of setting the parameter ranges arbitrarily at the first iteration, the ranges of the parameter space are set to match the location ranges of the circles in the image space. In this manner, the interactions between the different circles, and the effects of noise, can be decreased. The processing time will also be reduced.

Moreover, an important advantage of this labelling method is that each location range can be used for detecting its circle in parallel with the others, because of the independence from each other of the different circle ranges. We can distribute the coarse location range of each circle to a separate processor, and the different processors can detect the individual circles simultaneously. As we know, the effectiveness of a multiprocessor system is highly dependent on the ratio of computing time to communication time. Computation is the function that contributes to problem-solving, while communication is only a means to an end (allowing the computation to be distributed). One must therefore keep each processor doing as much computing as possible, in order to utilize all the processors in a system to the fullest extent. In this method, once the different circle ranges have been distributed to different processors, computation of AHT can proceed without much necessity for communication between processors. Thus, the parallel implementation of multiple-circle detection can achieve high efficiency levels.

EXPERIMENTS

In the first experiment, the program is executed in sequence. At the first stage, we input a binary image (as in *Figure 5*) and scan the whole image, labelling the components by tracking runs of 1's (component interior points) rather than by finding the borders. Connected components are given the same label. Thus the bigger the hole is, the higher the value of its label number. As a result, the label number of the bigger hole is 8532, that of the smaller one 640, and that of the cluster of noise 61. And the label of the bigger hole is 39, that of the smaller one 9, and that of the cluster of noise 1. We select only the highest and the second highest label numbers, and can easily get rid of the noise.

Next, the coordinates of points A, B, C, D and E, F, G, H can be calculated by using the labels 39 and 9. The coordinates of these points are shown in *Table 1*. We can add a small number, for instance 5, to the

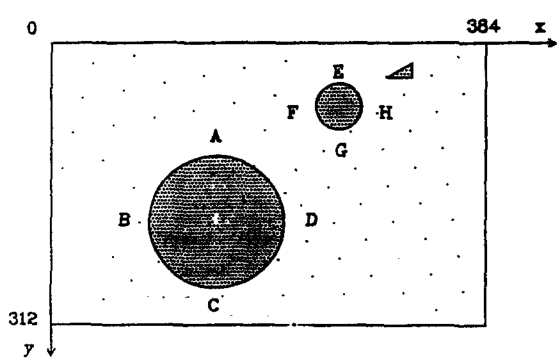

Figure 5 A sheet of metal containing two holes, a cluster of noise, and some spatially-random noise. AC, BD, EG and FH are the coarse ranges of the circles

Table 1 The coordinates of the points in *Figure 5*

	x-coordinate	y-coordinate
A	164	170
B	109	217
C	164	273
D	212	217
E	182	101
F	169	112
G	182	129
H	197	112

y-coordinates of C and G and the x-coordinates of D and H, and at the same time subtract the same number from the y-coordinates of A and E and the x-coordinates of B and F. So the coarse ranges of these two circles are: circle 1: $(x, y) = (104 \sim 217, 165 \sim 278)$ and circle 2: $(x, y) = (164 \sim 202, 96 \sim 134)$, respectively.

At the second stage, the AHT method is applied to detecting each circle's centre and radius, where the circles form the boundaries of the holes (see *Figure 3*). First we map the points in the range of $(x, y) = (104 \sim 217,\ 165 \sim 278)$ into the (a, b) parameter space. This time the parameter a varies only from 104 to 217, rather than from 0 to 384, and b from 165 to 278, rather than from 0 to 312. The computational burden is therefore decreased. After 3 iterations, the bigger circle is determined with its centre at $a = 159.96$, $b = 222.43$, and a radius of 53. Then the small circle is detected using a similar process. Its parameters are found to be: $a = 181.91$, $b = 115.63$ and radius is 15.

In the second experiment, the algorithm is executed in parallel, using a Transputer Network (a Meiko Computing Surface) composed of 8 Transputers, IMS T800, where the 64-bit floating-point unit is able to perform floating-point arithmetic operations concurrently with the processor[6]. For the image shown in *Figure 3*, three transputers are used. The block diagram is shown in *Figure 6*.

One Transputer executes the labelling process, and the other two detect the two circles, respectively. After Transputer 1 has finished labelling the image in *Figure 5*, the whole image is divided into two ranges, each range containing a circle. Then Transputer 1 sends the two coarse ranges down to Transputers 2 and 3 respectively, using hard channels. Transputers 2 and 3 receive the information, and perform AHT simultaneously.

Because the information received by Transputers 2 and 3 is independent, there is no need for

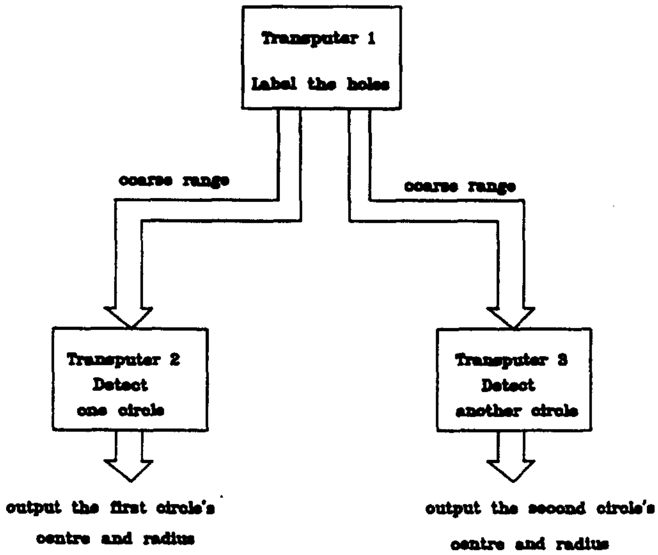

Figure 6 Block diagram of three transputers used in detecting two circles

communication between them. As a result, the ratio of computing time to communication time is increased. The processing time is 3 seconds.

CONCLUSIONS

We have presented a modification of the AHT method, the combination of AHT and the labelling method, to detect two circles in an image. By using LAHT, different circle ranges can be distinguished. It is this 'labelling' that has the advantages in both sequential and parallel processes:

1) When using a sequential process to detect two circles, it is necessary to map only the features in a specific range of the image space onto the parameter space to detect the specific circle. Thus, both the interactions between the different circles and the effects of clusters of noise, and the processing time, will be decreased.
2) When using a parallel process, the two different location ranges of circles can be distributed to two processors, respectively. These two processors can then detect each circle simultaneously without any need for communication between them. The experiments show that when using the AHT method, it takes 11.97 seconds for detecting two circles, but only 3 seconds for detecting two circles when using LAHT in parallel. The processing time is therefore greatly reduced.

If an image contains more circles, we can use more transputers. Because there is little need for communication between transputers, a linear increase in processing speed can theoretically be obtained by using an appropriate number of processors. The proposed modification is also impervious to the existence of clusters of noise. This not only increases the accuracy of circle detection, but also reduces the processing time, especially when an image contains spatially-random noise or clusters of noise. Finally, we point out that the labelling process can also be parallelized[7].

REFERENCES

1 Illingworth, J. and Kittler, J. The adaptive Hough transform, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-9**(5), pp. 690–698 (September 1987)
2 Hough, P. V. C. *Method and means for recognizing complex patterns*, U.S. Patent 3 069 654 (1962)
3 Duda, R. O. and Hart, P. E. Use of the Hough transform to detect lines and curves in pictures, *Comm. ACM*, pp. 11–15 (March 15, 1975)
4 Ballard, D. H. Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition*, **13**(2), 111–122 (1981)
5 Rosenfeld, A. and Kak, A. C. *Digital Picture Processing*, second edition, **2**, p. 243 (1982)
6 INMOS product information, *The transputer family*, p. 16 (1987)
7 Lim, W. A fast parallel algorithm for labeling connected components in image arrays, *Proc. Int. Conf. on Parallel Processing for Computer Vision and Display*, **2**, University of Leeds, UK (1988)