An algorithm to determine corner points in an image

P. Chongstitvatana
Department of Computer Engineering
Chulalongkorn University
Bangkok 10300, Thailand
email: fengjps@chulkn.ac.th

Abstract

Corner points in an image are distinctive features such as discontinuity, points where two objects are overlapped or various intensity curvature maxima. They are useful low-level features in machine vision, with applications in object tracking, recognition and measurements. This work presents an algorithm to determine corner points in an image emphasising on robotic applications. The algorithm does not use spatial derivatives of the intensity of an image. It has been tested with real images. Comparison with other algorithms are presented.

Introduction

It is useful to find distinctive features in images. They can be used to locate, to measure some characteristic of objects in images, or to track the movement of objects. Corner points can be defined as interesting change in image intensity. The method should not use any geometrical model of objects, i.e. it is a low-level image processing.The necessary information are the change in intensity, some assumptions about the shape (in 2-D) and the continuity of intensity change of a surface.

There are fast algorithms to find the corner points on the boundary of objects. A method (Malcolm, 1982) uses the trace of boundary of objects and finds the corner points in these traces. Many methods differ only in the way to calculate the gradient along the boundary traces.

The difficulty concerning this method is that in order to trace the boundary of an object a threshold must be select to distinguish the object from its background. (This is equivalent to transforming a greyscale image to a binary image). The threshold selection is sensitive to the assumption about the uniformity of the background. Another limitation is that the corner points inside an object cannot be determined.

To find corner points which are not on the boundary of objects the whole image must be searched. Many works have been reported (Harris and Stephen, 1988; Kitchen and Rosenfeld, 1982; Medioni and Yasumoto, 1986; Singh and Shneier, 1990). Most methods are based on brightness spatial derivatives which use first and second derivatives. This can severe the problem of noise in the image. To reduce the effect of noise, the image or the derivatives can be smoothed.

The algorithm

The method presents in this work does not use any image brightness spatial derivatives. Instead, it uses a correlation of pixels and their neibours. Imagine a pixel being considered as a candidate to be a corner point. It is a centre of a circle which contains its neibours. The brightness of the centre is compared with other pixels in the circle. This will determine the area which has the same brightness as the centre. The ratio of this area to the circle area can be used to determine corners (the shape of this area can also be used). The idea can be understood easily from an example (fig. 1).

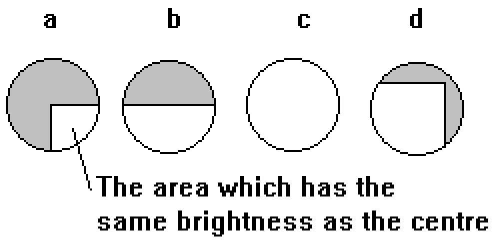


Figure 1 The mask for finding a corner at different situations.

Figure 1 depicts a circular mask at four different situations. The centre of the mask is the point that being examined. Assume that the object in an image is a rectangle. The mask at a) is at a corner of the rectangle, b) is at a side, c) is either completely on the background or inside the object and d) is near a corner. By inspection it is clear that if the area of the same brightness is less than half of the mask area then the centre is placed on or near a convex edge of a surface. This is the case for a) and d). Another observation is that the local minimum of the same brightness area will occur at the corner point (compare a and d). This is essentially the method in Smith (1992).

The detail of the algorithm is as follows. Comparison of the brightness of the centre and each pixel in the mask is defined as a function:

$$c(\vec{r},\vec{r}_0) = 100e^{-\left(\frac{I(\vec{r})-I(\vec{r}_0)}{t}\right)^6} \qquad (1)$$

where c is the result of comparison, $\vec{r}_0$ is the position of the centre, $\vec{r}$ is the position of other pixel in the mask, $I(\vec{r}_0)$ is the brightness of the centre, $I(\vec{r})$ is the brightness of any pixel, t is the brightness difference threshold. This function is due to Smith (1992). The form of equation was chosen to allow a pixel's brightness to vary slightly without having a large effect on c. To determine the area, a sum of c is made:

$$n = \sum c(\vec{r},\vec{r}_0) \qquad (2)$$

The main difference between this work and Smith's is the way in which this sum has been made. Smith counts all the pixel within the mask. This work counts only the pixel on the circumference of the mask. The main thrust is that it is faster to use the circumference than to use the area, in this case $2\pi r$ as opposed to πr^2 pixel comparison. The reason is that if the ratio of the area can be used to determine corner points so is the ratio of the arc. The only concern is that using the arc will be more prone to the noise than using the area. The experimental results (in the next section) show this to be acceptable.

The next step is to compare n with a threshold g (geometrical threshold) which should be set smaller than half of the maximum of possible n. If n is less than g, this pixel is counted as a candidate and g-n is the strength of that corner point otherwise is zero. An image is created with each pixel contains the strength value of a corner point. This image is searched for local maxima (which is above zero) within a window the same size as the mask to report corner points (this is called non-maximum suppression).

Experiments

Because the algorithm in this work is a derivative of Smith's work, it would be interesting to compare the speed and accuracy of the reported corner points of two algorithms. Both algorithms have not yet been reported with the actual working applications therefore to establish a baseline, the Malcolm's algorithm (1987) is used. Malcolm's algorithm has been used extensively in real robotic applications as reported in (Chongstitvatana and Conkie, 1992) and the detail of its implementation can be found in (Chongstitvatna, 1992).

The parameters used in the experiments are as follows. The mask is a 5 x 5 pixel square with 3 pixels added on the centre of each edge. It is a reasonable approximate of a circle. The difference threshold t is 25. The geometrical threshold g is 750. (The mask has 16 pixels as its circumference. Each can contribute maximum value of 100. Half of the sum will be 800. Given the noise allowance of 50 and the threshold figure will be 750). Test images are 256 x 256 pixels x 8 bits (256 grey levels).

All three algorithms were tested on a synthetic image with added noise (figure 2). The result is shown in table 1 (*MA* = Malcolm, *SM* = Smith, *CH* = this algorithm). For the speed results, all timings are normalised to Malcolm's time. The result shows that this algorithm is about 2.6 times faster than Smith (not much of a surprise as the number of pixels in the mask for *SM* is 37 and 17 for *CH*, the ratio is 37/17 = 2.16). The reason for *MA* to be much faster than *SM* and *CH* is that *MA* searches only the point on the boundary of objects while *SM* and *CH* must search all points in the image.

	t_{MA}	t_{SM}	t_{CH}	t_{SM} / t_{CH}
synthetic	1.0	48.5	18.3	2.6
gear	-	-	-	2.7
face	-	-	-	2.8

Table 1 : The speed test results

On the accuracy test between *SM* and *CH* (no comparison with *MA* is made as *MA* does not report the *internal points* that *SM* and *CH* can) three measures are made : the number of points that are matched m, the number of points in *SM* that do not matched n_{SM} and the number of points in *CH* that do not matched n_{CH} (therefore $m + n_{SM}$ = total number of points that *SM* reported and $m + n_{CH}$ = total number of points that *CH* reported). Two points are considered matched if they lie within 4 pixels of the other (the *s.d.* show radius of distance of the matches) (table 2).

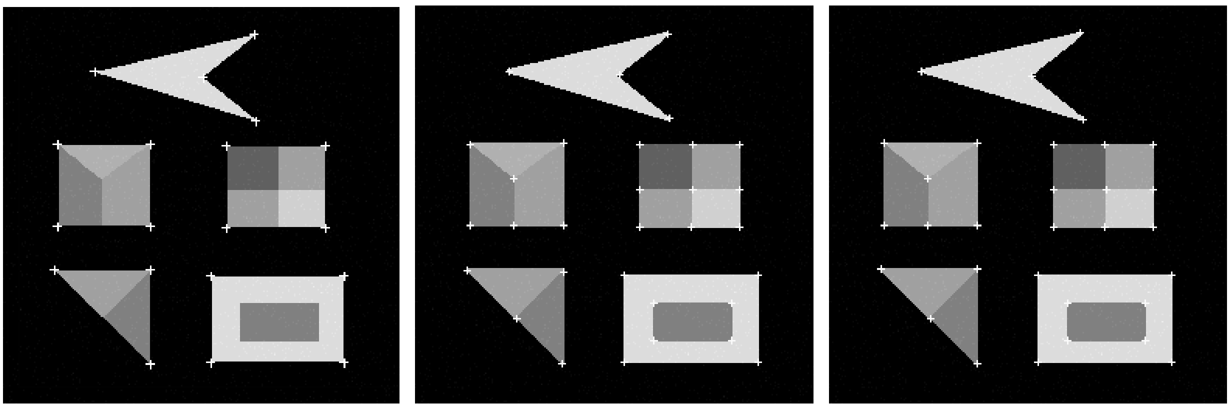
To test the assumption that using the ratio of the arc will give a similar result as using the ratio of the area, both *SM* and *CH* algorithms are tested on two real images. The first image is a form of geometrical shape (fig. 3). The second image is a natural shape (fig. 4). The results show that *CH* algorithm is consistently 2.7 times faster than *SM*. The matched ratio is considered as $(m - n_{SM}) / n_{SM}$ (e.g. use *SM* as reference). Both tests have good matches (95% and 100%). Miss matches due to noise in *CH* is acceptable (n_{CH} / m = 3.9% and 17%).

Conclusion

This work presents an algorithm to find corner points which is not based on spatial derivative of brightness. It has been tested on real images and is as accurate as its parent's algorithm (Smith) but it is faster. The future work will be to put an effort on improving the robustness of this algorithm and to test it on the real robotic applications.

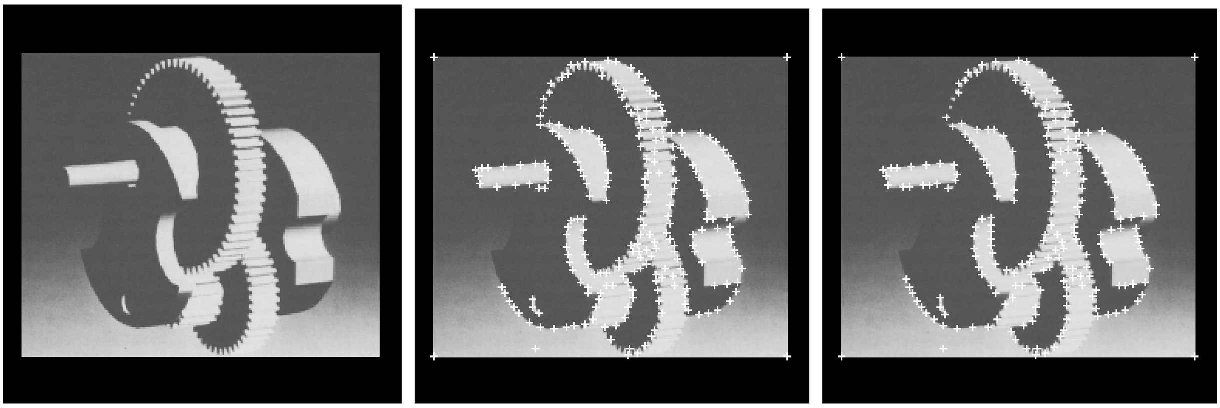
	m	n	n_{CH}	s.d.	matched	noise
synthetic	31	0	0	0.67	100 %	0 %
gear	227	11	9	3.89	95 %	3.9 %
face	79	0	14	3.10	100 %	17 %

Table 2 : The accuracy test results



a) b) c)

Figure 2 : A synthetic image a) process by *MA* b) process by *SM* c) process by *CH*



a) b) c)

Figure 3 : A geometric shape image a) original b) process by *SM* c) process by *CH*

a) b) c)

Figure 4 : A natural shape image a) original b) process by *SM* c) process by *CH*

References

Chongstitvatana, P., and A. Conkie. (1992). An uncalibrated stereo visual servo system. *Proc. of 23rd Int. Sym. on Industrial Robots*, Barcelona, Spain, pp. 393-397. Also Department of Artificial Intelligence, University of Edinburgh, research report (DAI RP) 590.

Chongstitvatana, P. (1992). The design and implementation of vision-based behavioural modules for a robotic assembly system. PhD. thesis, Department of Artificial Intelligence, University of Edinburgh.

Harris, C.G. and M, Stephens. (1988). A combined corner and edge detector. *4th Alvey Vision Conference,* pp. 147-151.

Kitchen, L. and A. Rosenfeld. (1982). Grey-level corner detection*. Pattern recognition letters*, 1:95-102.

Malcolm, C.A. (1983). The outline corner filter. *Proc. of the 3rd Int. Conf. on Robot Vision and Sensory Controls*, pp. 61-68. Also DAI RP 212.

Medioni, G. and Y. Yasumoto. (1986). Corner detection and curve representation using curve b-splines. *Proc. CVPR*, pp. 764-769.

Singh A. and M. Shneier. (1990). Grey level corner detection: A generalisation and a robust real time implementation. *Computer Vision, Graphics and Image Processing*, 51:54-69.

Smith, S. (1992). A new class of corner finder. *Proc. of British Machine Vision Conf.*, Hoggs, D. and R. Boyle eds., Springer Verlag, pp.139-148.