A New Class of Corner Finder

Stephen Smith
Robotics Research Group, Department of Engineering Science,
University of Oxford, Oxford, England,
and
DRA (RARDE Chertsey),
Surrey, England

January 31, 1992

Abstract

An accurate, stable and very fast corner finder (for feature based vision) has been developed, based on a novel definition of corners, using no image derivatives. This note describes the algorithm and the results obtained from its use.

1 Introduction

A lot of vision research lies in the area of 'early vision'. This has as its goal the reduction of image data so that information becomes more manageable and more immediately useful. This is commonly achieved by reducing a greyscale image to a list of edges or a list of 'corners'. (See [1] for a discussion of the merits of such an approach.) The mathematical description of a greyscale edge has been fairly well defined, but there are many different mathematical descriptions of the 2–D image structure which defines a corner. This is not surprising as the term 'corner' is quite vague. Indeed, an accurate definition of a 'corner' cannot go much further than *a position in the 2–D array of brightness pixels which humans would associate with the word 'corner'*.

As a result of this, many different corner finding algorithms have appeared in vision literature. For a good review of most of the different methods of corner finding see [2]. Some of the more interesting work which has been done is covered in [3], [4], [5], [6], [7], [8] and [9]. Differing methods each have an idea of a definition of what a 'corner' is, and this idea is translated into a mathematical way of finding corners. Reference [2] in fact shows that most of the different methods end up with very similar mathematics. This note describes a completely new type of corner finder, in both its definition and the outworking of the mathematics. It is called the 'Smallest Univalue Segment Assimilating Nucleus' ('SUSAN') corner finder and is described below[1]. The idea behind it is relatively straightforward and is explained in section 2. In section 3 the algorithm that has been developed is described in detail. Section 4 gives various results of the SUSAN algorithm.

[1] Patents have been applied for (by DRA) completely covering the principles of the SUSAN corner and edge finders.

BMVC 1992 doi:10.5244/C.6.15

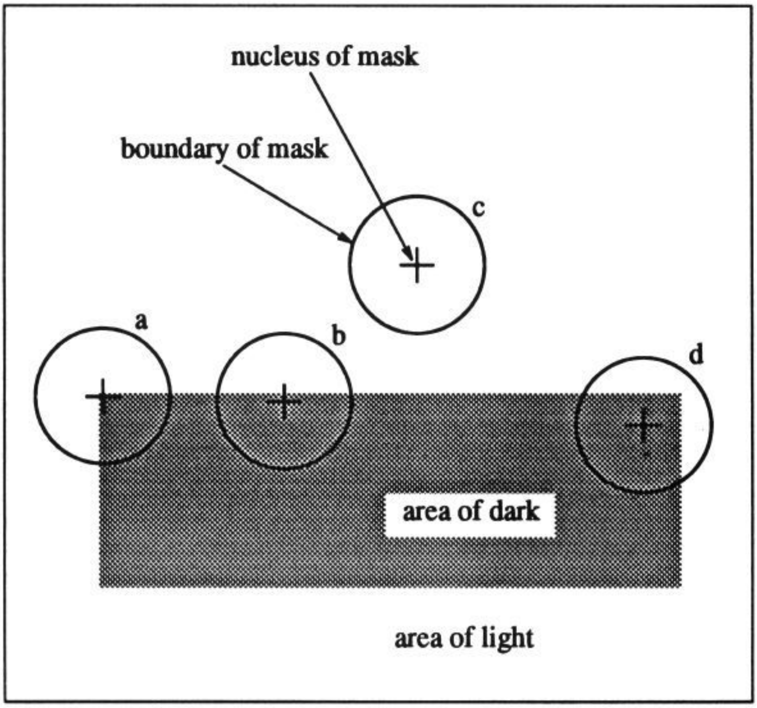

Figure 1: Four 'corner finding' masks at different places on a simple image.

2 The Principle Behind the SUSAN Algorithm

This section describes the SUSAN principle from an algorithmic point of view, as this is the simplest approach.

Each image pixel is used as the central point of a small mask (in this case circular) which contains the few pixels which are closest to the central one. This idea is shown for just four positions in an image in Figure 1. The mask is delimited by a circle and the central pixel (which shall be known as the 'nucleus') is marked by a cross.

The first step is to look at the greyscale (brightness) of the nucleus and compare this with the greyscales of the other pixels within the circular mask. The pixels with similar brightness to the nucleus are then assumed to be part of the same surface in the image (for example the light or dark surface in Figure 1). Figure 2 shows the masks with pixels of similar brightness to the nucleus coloured white, whilst pixels with different brightness are coloured black. The white portion of the mask is now used to determine whether the nucleus is positioned over a 'corner' or not.

The white portion of the mask shall be known as the 'USAN' or 'univalue segment assimilating nucleus'. (Here 'univalue segment' means a segment of the small mask which locally has constant or nearly constant brightness. It is of interest when it contains the nucleus.) Although a detailed study of the

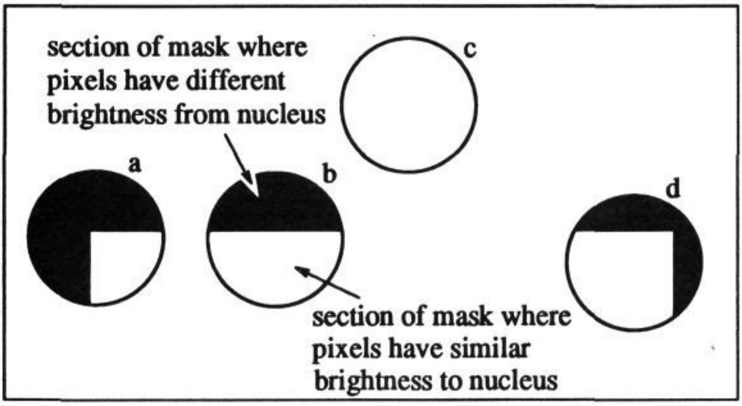

Figure 2: Four 'corner finding' masks with similarity colouring; USANs are coloured white.

shape of the USAN would enable corners to be detected, this is not necessary. Instead a much simpler rule suffices. It can be observed from Figure 2 that the USAN corresponding to the corner – case (a) – has relatively small area. In fact, if the USAN has an area of less than half the mask area, then the nucleus is placed on or near a convex edge of a univalue surface. Cases (b) and (c) clearly fail this condition. If an upper limit for the USAN's area is set at some fraction less than 1/2 (for example 2/5) of the mask area, then the curvature of the convex edge must be above some minimum level.

It is also clear that a local minimum in USAN area will find the exact point of the corner. For example, the USAN in case (d) of figure 2 will not have as small an area as it will when the nucleus is placed exactly on the corner – compare with case (a). Hence the term 'smallest USAN' gives rise to the algorithm's name.

One of the novel aspects of this corner finder is that it does not use any image brightness spatial derivatives in the algorithm. It has been common until now to use first and second derivatives of the brightness, amplifying the problem of noise in the image. Most former methods have attempted to reduce the effects of noise by smoothing the image or the derivatives, but this inevitably reduces the quality of the localisation (accuracy in 2–D location) of the corners. The SUSAN algorithm is the first corner finder to use no spatial derivatives; it therefore does not need any smoothing process, and so there is no degradation in localisation.

3 The SUSAN Algorithm in Detail

In this section the SUSAN algorithm is described in detail.

The current implementation of SUSAN uses for its mask a 5 pixel by 5 pixel square with 3 pixels added on to the centre of each edge. This shape forms a fairly good digital approximation to a circle. The size was carefully chosen. A smaller mask gave results that were not very stable, that is, a corner was not reliably detected at the same place in several images taken in succession (even if the camera was static). A larger mask (based on a 7 by 7 square) gave stable results but was too large for the smallest structures (which were identifiable in the digitised image) to be detected. Also the larger mask was twice as slow as the one finally used.

The next step in the SUSAN algorithm is to compare the brightness of each pixel in the defined mask with that of the nucleus. A simple function determines this comparison;

$$c(\vec{r},\vec{r_0}) = 100e^{-\left(\frac{I(\vec{r})-I(\vec{r_0})}{t}\right)^6}, \tag{1}$$

where $\vec{r_0}$ is the position of the nucleus in the 2–D image, $\vec{r}$ is the position of any other point within the mask, $I(\vec{r})$ is the brightness of any pixel, t is the brightness difference threshold and c is the output of the comparison.

The form of equation 1 was chosen to give a similar response to a box function but with slightly rounded corners. This allows a pixel's brightness to vary slightly without having a large effect on c, even if it is near the threshold position. The exact form for equation 1 was carefully chosen (empirically) to give a balance between good stability about the threshold and the function originally required (namely to count pixels that have similar brightness to the nucleus as "in" the univalue surface and to count pixels with dissimilar brightness as "out" of the surface). The equation is implemented as a look up table for speed.

This comparison is done for each pixel within the mask, and a running total, n, of the outputs (c) is made;

$$n = \sum_{\vec{r}} c(\vec{r},\vec{r_0}). \tag{2}$$

This total n is just 100 times the number of pixels in the USAN, i.e. it gives the USAN's area. As described earlier, this total is eventually minimised.

Next, n is compared with a fixed threshold g (the 'geometric threshold') which should be set to just under half of the maximum possible value for n. This prevents straight boundaries from giving false positives.[2]

The two thresholds so far introduced are good examples of two different types of threshold. The geometric threshold clearly affects the 'quality' of the output. Although it affects the number of corners found, much more importantly, it affects the shape of the corners detected.[3] For example, if it were reduced, the allowed corners would be sharper. Thus this threshold can be fixed (to the value previously explained) and will need no further tuning. Therefore no weakness is introduced into the algorithm by the use of the geometric threshold. The brightness difference threshold is very different. It does not affect the quality of the output as such, but does affect the number of corners reported. Because it determines the allowed variation in brightness within the USAN, a reduction in this threshold picks up more subtle variations in the image and gives a correspondingly greater number of reported corners. This threshold

[2] In practice, the threshold is set to exactly half of the maximum of 3700 (this number comes from the 37 pixels in the mask; the nucleus automatically contributes a count of 100). This threshold was chosen because a straight edge including the nucleus will always be greater than half of 3700 (1850) by at least 50, due to quantisation of the pixels, and the threshold should be set as large as possible to allow the maximum variety of corners.

[3] This assumes that the USAN is a contiguous region. The refinements described in [10] are designed to enforce this contiguity.

can therefore be used to control the 'quantity' of the output without affecting the 'quality'. This can be seen as a negative or a positive point. On the one hand, it means that this threshold must be set according to the contrast etc. in the image, and on the other, this threshold can be easily tuned to give the required density of corners. In practice, there is no problem. A fixed value of 25 is suitable for almost all real images, and if low contrast images need to be catered for, the threshold can be set automatically, simply by varying it to give the required number of reported corners. When SUSAN was tested on an extremely low contrast image this threshold was reduced to 7. This gave a 'normal' number of corners. Even at this low value, the distribution was still good (i.e. not over-clustered) and the corners found were still quite stable.

Next an intermediate image is created from the value of n found at each image position. If $n(x,y)$ is less than the geometric threshold then $(g - n(x,y))$ is put into the new image at (x,y). If it does not pass this test, then zero is put into the image at this place. This stage is done so that local minima in the values of n which pass the geometric test may be found, to locate corners exactly. In practice therefore, the intermediate image is searched over a square 5 by 5 region for local maxima (above zero). Finally, these local maxima are reported as corners. The entire process, run on a Sun Sparc 2 processor, using a 256 by 256 pixel image takes about 1/3 of a second for an average scene.

This simple algorithm is the basis for a successful corner finder. It is a completely new way of approaching the problem.

A rigorous mathematical analysis of the SUSAN algorithm would be very complicated as its validity, as explained here, though fairly obvious, is more 'intuitively picturesque' than analytic. However, a mathematical analysis has been performed, and can be found in [10]. This analysis gives as the final mathematical interpretation of the SUSAN principle,

$$\frac{dI}{d\vec{r_0}}(\vec{r_0}) =< \frac{dI}{d\vec{r_0}}(\vec{r}) >_{\text{over edges}} . \tag{3}$$

This equation expresses the fact that the intensity differential at the nucleus must be the same as the intensity differential averaged over all of the edges lying within the mask. This must be the case for both the x and y differentials. For this to hold, the nucleus must be placed on a line of reflective symmetry of the boundary pattern within the mask area. This is so that the contributions to the average differential on either side of the nucleus may balance out. This is equivalent to saying that the nucleus must lie on a local maximum in the edge curvature; consider a Taylor's expansion of the edge curvature. At the smallest local level, the nucleus will only lie on a centre of symmetry of the curvature if it lies at a maximum (or minimum) of curvature.

The conditions derived ensure that the nucleus lies not only on an edge, but also that it is placed on a sharp point on the edge. It will as it stands give a false positive result on a straight edge; this is countered by not just finding local minima in n, but by forcing n to be below the geometric threshold g.

4 Results

The results of testing this corner finder are shown and discussed in this section. Firstly, the output of SUSAN given a test image[4] is shown in figure 3 (top). Compare this with figure 3 (bottom), the output from the Plessey corner finder, which looks for large and similar principal curvatures in the local autocorrelation function. The accurate localisation and reliability of the SUSAN algorithm is apparent; it does all it is expected to do. Algorithms based on image derivatives often have problems at some of these junctions, for example where more than two regions touch at a point. There is no problem for SUSAN. This is expected, as the corner finder will look at the point of each region individually; the presence of more than two regions near the nucleus will not cause any confusion. The local non-maximum suppression will simply choose the pixel at the point of the region having the sharpest corner for the exact location of the final marker. The inaccuracies of the Plessey corner finder, even at simple two region corners, are visible. This is discussed and explained in [11]. With respect to speed, SUSAN took 0.3 seconds to process this picture on a single RISC processor; the Plessey corner finder took 3.5 seconds.

The SUSAN algorithm has also been tested with respect to its sensitivity to noise. The results are very good; the quality of its output (both the reliability and localisation) degrades far less quickly than other algorithms tested as noise in the image is increased. In the following example, the original test image had a considerable amount ($\sigma = 3$) of gaussian noise added. The outputs of SUSAN and the Plessey corner finder are shown in figure 4 (top and bottom respectively).

SUSAN has also been tested with very many individual real images; unfortunately they must be seen in [10] due to space constraints.

The temporal stability of SUSAN has also been analysed. The output from several consecutive frames from a moving camera was used as the first stage of the DROID 3–D vision system developed by Plessey (see [1], [12] and [13]). This program tracks corners through time in order to reconstruct a 3–D description of the world. The results obtained when the Plessey corner finder was used were compared with those obtained when SUSAN was used. Several different sequences were tested. Some sequences gave slightly better 3–D output data when using the SUSAN algorithm, and the rest gave similar results with both algorithms. The results were compared by observing the quality of a least squares plane fit through the tracked 3–D points which were on a plane in the world, and also by using this to detect small obstacles in the vehicle's path. It was found that the size of the smallest objects which could be detected by DROID was less when the SUSAN corner detector was used than when the Plessey corner finder was used. This suggests that in this example the quality of the data was slightly better when using the SUSAN corner finder.

In these tests, SUSAN ran on average 10 times faster than the Plessey algorithm.

[4] This test image has been developed by the author to include two dimensional structures of many different types, and ones which existing corner finding algorithms often cannot correctly interpret. Note simple 90° corners, corners with angles close to 0° and 180°, various junctions with more than 2 regions meeting at a point, two corners close together, and corners created by taking a single brightness ramp and raising a rectangle in its centre by a uniform brightness.

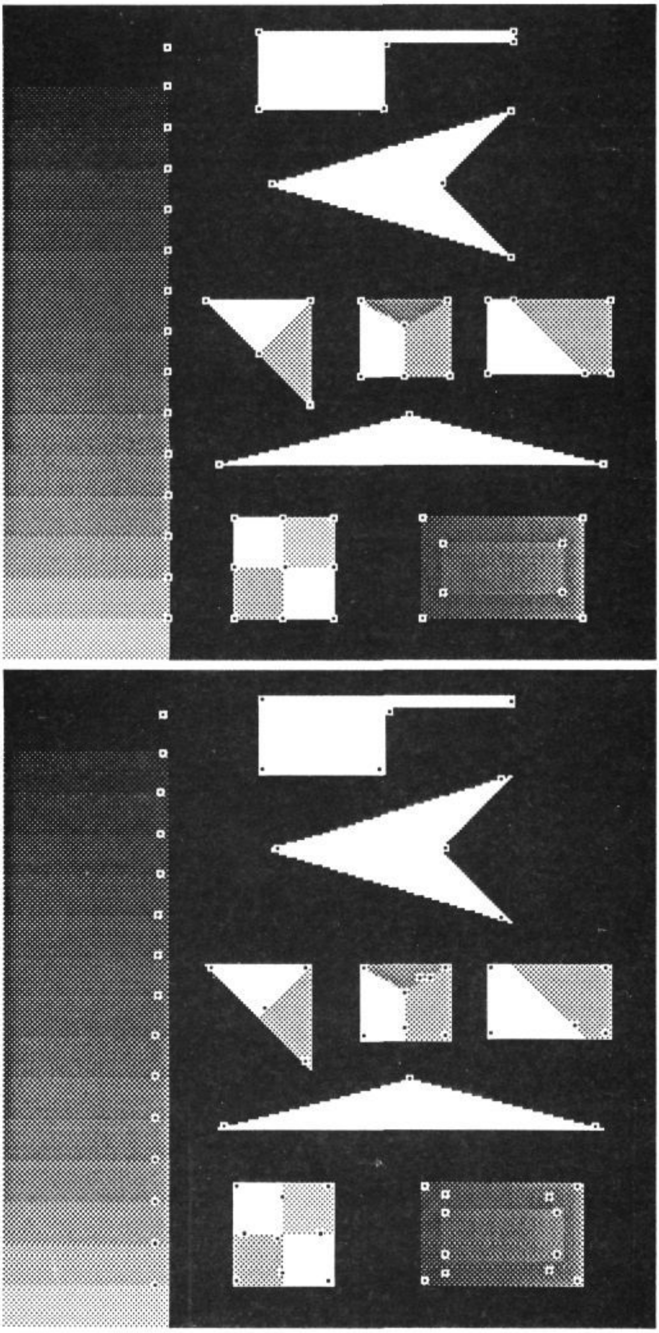

Figure 3: Result of SUSAN corner finder on test image (top), result of Plessey corner finder on test image (bottom).

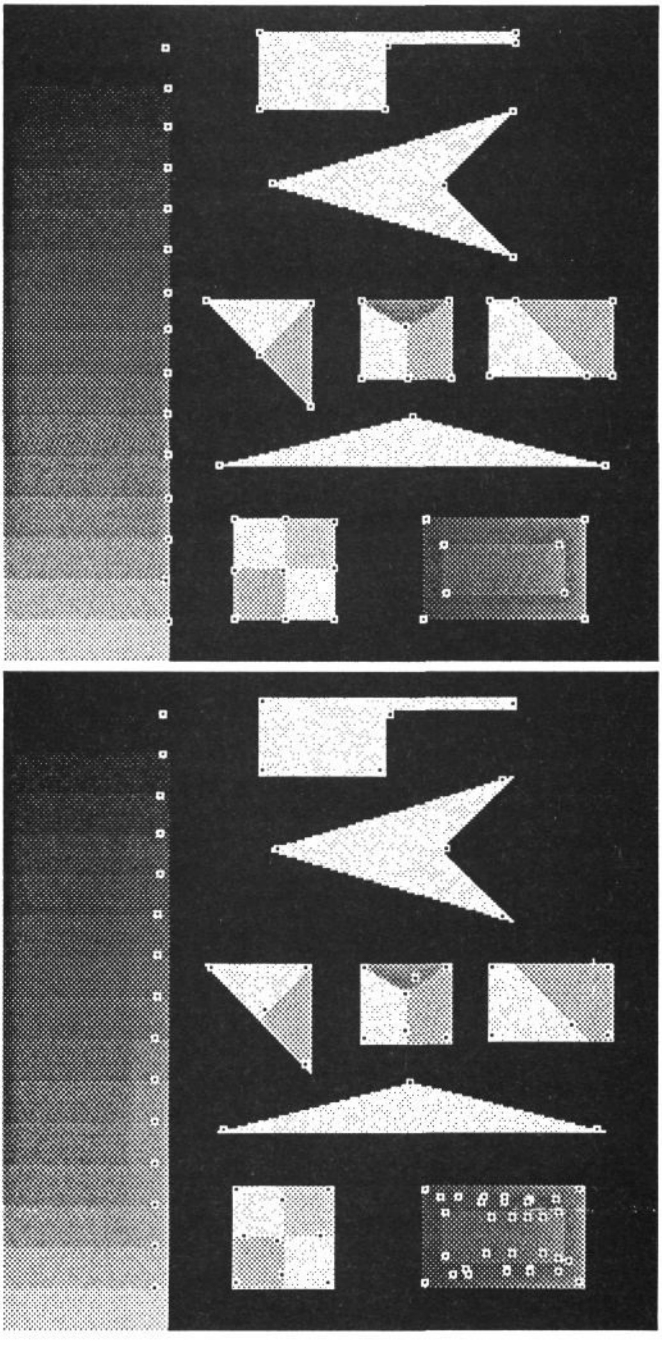

Figure 4: Result of SUSAN corner finder on test image with gaussian noise ($\sigma = 3$) added (top), result of Plessey corner finder on same test image (bottom).

As another quantitative test of its stability and general suitability for real applications, SUSAN has been used to provide corners for a program (at present being developed by the author) which segments a stream of images into independently moving objects. This program uses corner motion to segment the images into parts which have different motion from each other. The results are extremely good, with successful segmentation of two vehicles travelling in front of a third carrying a video camera.

5 Conclusions and Future Work

A completely new approach to corner finding has been developed and tested. It is accurate, stable and very fast. It has been tested for accuracy on both test and real images, and additionally for speed and stability on image sequences. It has given very good results in all the tests and has been used with complete success as a front end for both a 3–D structure-from-motion program and a motion segmentation program.

The SUSAN algorithm has been shown to be related to finding local maxima in edge curvature. However, because the algorithm uses regions to find corners and not first or second image derivatives, it is very good at ignoring noise in the image, and also very good at producing well localised corners. This is the first approach which does not use any image derivatives.

The author has recently extended the principle of minimising the number of similar brightness neighbours to give a combined corner and edge detector. Using the same initial response map as described above, the geometric threshold can be eliminated so that edge enhancement is achieved. Edge direction can be found from USAN centre of gravity and direction of symmetry, and the edges can be thinned and localised to sub-pixel accuracy.

The resulting algorithm has the advantage over most current edge finders that there is continuity at junctions, as the response rises rather than falls at junctions of more than two edges. It is also very fast. As with the SUSAN corner finder, the setting of the remaining threshold is very simple and not sensitive to finding an exact "right" value. The results are extremely good, and the author is preparing a paper giving more details of the algorithm and its results.

6 Acknowledgements

The author thanks his wife, J. M. Brady, J. Savage and R. Taylor for their input and encouragement.

References

[1] D. Charnley, C. Harris, M. Pike, E. Sparks, and M. Stephens. The DROID 3D vision system – algorithms for geometric integration. Technical Report 72/88/N488U, Plessey Research Roke Manor, December 1988.

[2] J.A. Noble. *Descriptions of Image Surfaces.* D.Phil. thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1989.

[3] L. Dreschler and H.H. Nagel. Volumetric model and 3D trajectory of a moving car derived from monocular TV-frame sequence of a street scene. In *IJCAI*, pages 692–697, 1981.

[4] H.-H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21(1):85–117, 1983.

[5] C.G. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[6] L. Kitchen and A. Rosenfeld. Grey-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[7] G. Medioni and Y. Yasumoto. Corner detection and curve representation using curve b-splines. In *Proc. CVPR*, pages 764–769, 1986.

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

[9] H. Wang. Corner detection for 3D vision using array processors. In *Proc. BARNAIMAGE 91*, Barcelona, 1991.

[10] S.M. Smith. *Feature Based Image Sequence Understanding.* D.Phil. thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1992.

[11] S.M. Smith. Extracting information from images. First year D.Phil. report, Robotics Research Group, Department of Engineering Science, Oxford University, June 1990.

[12] D. Charnley and R.J. Blissett. Surface reconstruction from outdoor image sequences. *Image and Vision Computing*, 7(1):10–16, 1989.

[13] C.G. Harris and J.M. Pike. 3D positional integration from image sequences. *Image and Vision Computing*, 6(2):87–90, 1988.