Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 1, 7–19
DOI: 10.2478/v10006-008-0001-3

amcs

A GENERALISED APPROACH TO THE USE OF SAMPLING FOR RAPID OBJECT LOCATION

E. R. DAVIES

Machine Vision Group, Department of Physics
Royal Holloway, University of London, Egham
Surrey, TW20 0EX, UK
e-mail: e.r.davies@rhul.ac.uk

This paper has developed a generalised sampling strategy for the rapid location of objects in digital images. In this strategy *a priori* information on the possible locations of objects is used to guide the sampling process, and earlier body-based and edge-based approaches emerge automatically on applying the right *a priori* probability maps. In addition, the limitations of the earlier regular sampling technique have been clarified and eased—with the result that sampling patterns are better matched to the positions of the image boundaries. These methods lead to improved speeds of operation both in the cases where all the objects in an image have to be located and also where the positions of individual objects have to be updated. Finally, the method is interesting in being intrinsically able to perform full binary search tree edge location without the need for explicit programming.

Keywords: Machine vision, automated inspection and assembly, rapid object location, sampling techniques.

1. Introduction

It is well over forty years since computers were first used to process images and to analyse the information in them. Subsequently, image analysis, computer vision and machine vision, to name a few closely linked areas, have developed into a wide discipline. In general, computer vision encompasses the underlying scientific concepts and existence theorems, while machine vision is more concerned with applying vision to real tasks such as automated inspection and assembly; in particular, it is very involved with practical issues such as reliability, robustness, accuracy, the speed of operation and system implementation (Davies, 2005). However, it would be a mistake to separate these subjects very far: they are actually inextricably linked and it is as well to remember that in an applied area what is achievable is always limited by what is scientifically possible.

In one paper it is impossible to cover all industrial applications and it will be useful to focus on automated inspection. In particular, all the practical issues mentioned above arise with some force in the food industry, though in the latter area, rapid processing and the cost of hardware implementation are of exceptional importance (Davies, 2000b; Davies, 2003). In fact, real-time processing has in the past necessitated the use of special electronic hardware to boost the speed of the host computer system so that incoming streams of quite large images can be managed. In earlier times, special electronic hardware was expensive, and this was problematic for the food industry because its cost often made sophisticated inspection systems unaffordable. In such cases, instead of being able to rely on the guaranteed quality resulting from the use of machines, factories had to continue to resort to whatever could be managed by human operators. This led to two problems: first, human operators easily become bored, tired and unreliable; second, they offer far from the desired 100% scrutiny of products. Thus the gradually reducing costs of fast electronic hardware were welcome. Nevertheless, it is still necessary to improve speeds of processing, so that: (a) more complex and sophisticated algorithms can be applied to industrial problems, and (b) more computation intensive tasks such as high volume cereal grain inspection can be carried out and 100% scrutiny achieved in practice.

This paper is particularly concerned with the speed of processing, and aims to see what can be done without the need for special hardware systems to boost processing speeds. Section 2 considers the image search aspect

of automated inspection. Section 3 goes on to analyse the types of algorithms that are commonly used for object location in inspection and other applications. Section 4 introduces the use of sampling techniques for drastically limiting the processing required for object location. Section 5 re-examines the sampling methodology and further develops and generalises it, while Section 6 provides more detailed theoretical analysis. Section 7 discusses 2D aspects of guided sampling and presents practical results obtained using it, while Section 8 presents general conclusions on the work.

2. Inspection process: Image search

As indicated in Section 1, the *raison d'être* of machine vision is to perform purposeful image functions: in different applications these might permit vehicles to be guided, blood cells to be located, fruit to be picked, or brake assemblies to be inspected. In this paper we concentrate on inspection, which has many aspects: checking on the sizes of components and the elimination of oversized or undersized items; the scrutiny of components to eliminate those with defects (e.g. bolts without threads); the location of contaminants such as glass shards or insects in raw foodstuffs; the detection of missing or misplaced parts in assemblies. In inspection, it is usual to initiate outright rejection of unacceptable products: while in some cases there is an opportunity for recycling, it is always a delicate decision whether it is financially worthwhile to do so. (However, latterly, both recycling in factories and recycling of rubbish—itself a matter of detailed scrutiny, though not of newly manufactured products—is becoming contentious and even a legal obligation.) Another function of inspection is that of providing information on the running of the plant, both to provide short-term feedback for control (as when the spread of cake mixture is measured and used to control its temperature) and to log measurable parameters such as the numbers of reject items.

While the aim of inspection is the scrutiny of products, it implies the need to locate all products in the input images. It turns out that locating products can be a quite computation-intensive task: in particular, it may be more computation-intensive than product scrutiny itself. This applies particularly when products appear sporadically in the input images and there is no short cut to searching everywhere for them; whereas once a product has been found, a region of interest can be set up around it and scrutiny only applied over this restricted area. Thus scrutiny may be quick and straightforward: e.g. a shape template may be applied or an intensity profile checked, and a simple accept/reject decision made. However, when surfaces have to be examined carefully for blemishes, or textures with their complex patterns have to be analysed statistically, scrutiny can be an exacting task and it may well be that it will be computationally demanding in its own right. Much will depend on the specific application. Overall, object location is liable to be the more computation-intensive task because of the need for unconstrained search over the whole image area, and is all the more tedious because objects will have random orientation as well as random position.

Finally, it should not be thought that the computation problem applies only to inspection: it is a very general factor, and applies in any application where many images have to be searched for events: one need only consider vehicle guidance and radar in air traffic control for other relevant examples. Hence, the purpose of this paper is to examine in some depth the methodology for rapid object location.

3. Object location

While the most obvious way of locating known types of objects in digital images is that of template matching, it usually involves excessive amounts of computation.[1] This is because objects appearing in 2D images have at least three degrees of freedom—in particular, two of position and one of orientation. However, instead of searching for objects using large whole-object templates, we can search for any small features they may have, and this will be much more efficient. For example, we may search for corners or edge points using sets of 3×3 masks and this will normally reduce the amount of computation by factors of the order of 10,000, since not only will the mask areas be reduced by about 1000 but also the number of masks needed to cope with varying orientation will be reduced by a factor of at least 10. However, this approach will often achieve far larger savings, because of any other degrees of freedom that may arise—especially those due to variations in size and shape. In fact, object variations of any sort play an important role in ruling out the use of whole-object templates. Nevertheless, any approach that does not use whole-object templates implies the need to infer rather than deduce the presence of objects (Davies, 2005): this necessitates some further computation and the development of appropriate methods for performing the process.

In fact, there are two classes of small features that are commonly used to initiate object location: one of these is point features such as corners and small holes, which are characterised by their (x, y) coordinates; the other is edge points, which are characterised by their (x, y) coordinates and by their orientation—though curiously edge points effectively involve only two parameters, because of the lack of constraint along the edge direction. Classically, the presence of objects may be inferred from edge features using methods such as the Hough transform, or from point

[1] For early discussions of template matching and how to reduce the computational load associated with it, see (Nagel and Rosenfeld, 1972; Rosenfeld and VanderBrug, 1977a; VanderBrug and Rosenfeld, 1977b).

features using graph matching techniques (Davies, 2005). Here we concentrate on the use of edge features for reasons of space, and because this fits in better with the sampling formalism to be developed.

To further back up the general approach, we mention the Sobel edge enhancement operator, which uses two 3×3 correlation masks:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(here unnormalised), which lead to estimates of the x and y components (g_x, g_y) of the local intensity gradient. From these components, we can deduce the intensity gradient magnitude g and orientation θ using the equations

$$g = (g_x^2 + g_y^2)^{1/2}, \tag{1}$$

$$\theta = \arctan\Big(\frac{g_y}{g_x}\Big). \tag{2}$$

It should be noted that the process of edge 'detection' involves making a decision about whether the magnitude component g is significant, e.g. greater than some set threshold (though this decision can be made more sophisticated by hysteresis thresholding, non-maximum suppression, or other means). Many alternative edge detectors exist, but there is little gain from exploring them further here.

3.1. Inference from edges. As remarked above, the Hough transform provides a standard means of inferring the presence of objects from edge location information (Davies, 2005). We illustrate the methodology by considering the case of circle detection. First, if an edge point is located at a position (x, y) and is on the boundary of a circle C of radius $\mathbf{R}$, then the centre of C must be located on a circle of radius $\mathbf{R}$ and centre (x, y). Similarly, if further edge points on C are found, the centre must lie on a set of such circles. Hence, by accumulating all such circles in a special image space called a 'parameter space', we can readily locate the most likely centre location for C—or for any other circles that might be present. The method is robust, because if any edge points are lacking through low edge contrast, noise, occlusion, or distortions, many 'votes' will have been accumulated and a lot of evidence built up supporting the hypothesis that a circle is present.

It will be clear that the method as outlined so far is wasteful in giving very many votes that will be far from circle centres, but this number can be vastly decreased by taking account of edge orientation information, and accumulating votes only along edge normal directions. The operation of this version of the Hough transform is indicated in Fig. 1. Note that now votes are mainly situated at peaks in the parameter space and that these are relatively easy to locate, both by virtue of being local maxima and by their absolute height.

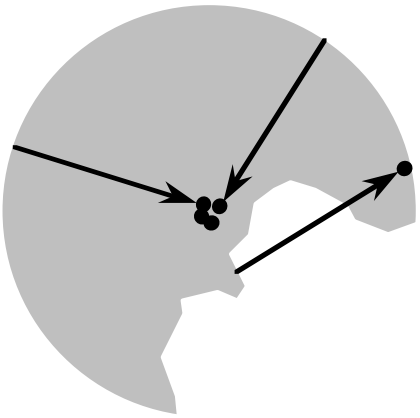


Fig. 1. Circle detection by the Hough transform: candidate circle centre locations are accumulated in the parameter space, as indicated by the arrows directed along the local edge normal directions. ©IET 2000

It is important to note that while peaks are most likely to represent the centres of circles, they could also represent chance placements of edges, and in this sense they amount to the hypotheses of the presence of objects, and the overall process of finding objects from the edge points is one of inference rather than absolute deduction. Nevertheless, in many applications (especially inspection applications) there may be a great deal of certainty that the specific objects are present, as in the case of biscuits on a biscuit line or round holes in an engine block.

To locate objects of other shapes in digital images, the Hough transform can be modified to detect ellipses, parabolas, straight lines, polygons and other analytically defined shapes. In addition, a powerful approach called the generalised Hough transform (GHT) may be used to detect objects of arbitrary shape, if suitable look-up tables are used (Ballard, 1981). We illustrate this method in Fig. 2 for the case where the objects have known or fixed orientation. Here, the local edge orientation ψ will indicate via a look-up table (the 'R-table') through what vector distance $\mathbf{R}(\psi)$ to move in order to get from the edge point to a reference point L within the object, thus enabling a vote to be cast in the parameter space. The computation is not great in such a case, but when object orientation ϕ is unknown, computation increases significantly, and increases even more if variable object size s or scale also has to be taken into account: in that case the R-table takes the form of a list of voting points with components (R, θ, ϕ, s) for each value of the indexing parameter ψ. For 3D work, especially when perspective projection has to be taken into account, the situation is even more complex and more specialised approaches tend to be adopted (Davies, 2005), but we do not consider this aspect here.

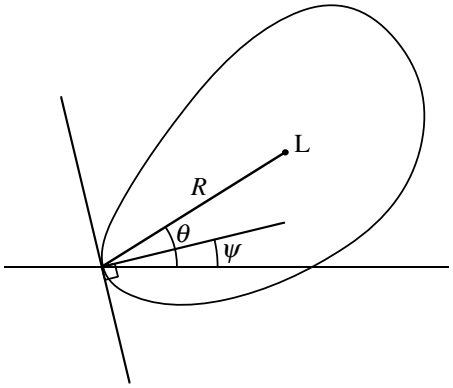


Fig. 2. Operation of the generalised Hough transform. To detect the shape, a vote is accumulated at the position of the reference point L that is estimated for each boundary point: L is assumed to lie at a vector location R relative to the boundary point, where $\mathbf{R} = (\mathbf{R}, \theta)$ is a function $R(\psi)$ of the local edge orientation ψ.

Most importantly, all the Hough transform-based techniques exhibit a high degree of robustness, and objects will have a high probability of being recognised and accurately located even if they are subject to low contrast, noise, serious damage, breakage, partial occlusion or contact with other objects—all of which confuse most boundary tracking algorithms, for example (Davies, 2005).

Although one of the aims of the Hough transform was that it would minimise computational load by virtue of using small templates in place of whole object templates, the GHT involves progressively higher computation as the number of free parameters increases. In fact, even the varying orientation version requires considerable processing. On the other hand, it is difficult to see how any other thoroughgoing approach could involve significantly less computation, as this is a direct result of the need for search in higher dimensional parameter spaces. Nevertheless, there are approaches that can reduce load, such as sequential analysis in carefully chosen subspaces. A simple case of this is searching for variable size circles by making multiple votes in a single 2D (x, y) parameter space, and following this up with a 1D search for appropriate radius values (Davies, 1988).

Another general approach to fast object location is that of multiresolution processing. Here an image is progressively smoothed to give a hierarchy of images, each image being typically twice as small in its linear dimensions as the preceding one. The areas of these smaller images will successively be $1/4, 1/16, 1/64, \ldots$ of the original area A, totalling $1/3$ of A, so analysing them instead of or as well as the original image need not be too burdensome. Then, by searching for larger objects in a smaller scale image, much processing can in principle be saved, even if further processing at the original resolution is required to bring accuracy back to ideal levels. Nevertheless, there is an increased possibility of error (either false positives or false negatives). In addition, the computation of the original reduced images does impose some additional load.

Finally, a projection method, which involves summing the image intensities along the x and y directions and forming two histograms, allows the possibility of 1D search operations to be carried out (Davies, 1987b). This can save immense amounts of processing, albeit at the expense of numerous false alarms if the image is at all complex. If nothing else, this does indicate that the minimum amount of processing for the whole image search can in principle be reduced to $O(M^2)$ for an $M \times M$ pixel image—essentially the processing needed to access each pixel twice.

Bearing these results in mind, there is some scope for searches that perform well, while accessing each pixel only a small number of times. In fact, we aim below to achieve similar results by sparsely sampling the image matrix, and performing significantly fewer than $M \times M$ pixel accesses.

4. Sparse sampling of the image space

4.1. Basic technique. In an initial approach to sampling, the author had the problem of finding the centres of circular objects such as coins and biscuits significantly more rapidly than for a conventional Hough transform, while retaining as far as possible the robustness of that approach (Davies, 1987a). The best solution appeared to be to scan along a limited number of horizontal lines in the image, recording and averaging the x-coordinates of mid-points of chords of any objects, and repeating the process in the vertical direction to complete the process of centre location (Fig. 3). The method was successful and led to speedup factors as high as 25 in practical situations.

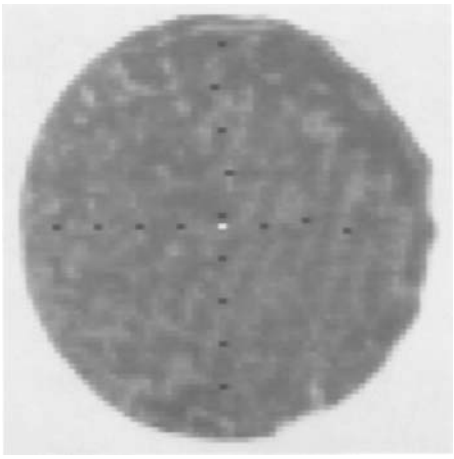


Fig. 3. Result of line-scanning for a circular object. Here the mid-points of chords are determined in the x and y directions after sampling every eighth row and column in the image.

In later work, which involved the inspection of huge numbers of wheat grains (Davies, 1998), extreme robustness was not necessary, and it seemed worth finding how much faster the scanning concept could be taken. It was envisaged that significant improvement might be achieved by taking a number of individual sampling points in the image rather than by scanning along whole lines: thus a 'body-based' approach rather than an 'edge-based' approach was adopted.

Suppose that we are looking for an object such as that shown in Fig. 4(a), whose shape is defined relative to a reference point R as the set of pixels $A = \{\mathbf{r}_i : \ i = 1 \text{ to } n\}$, n being the number of pixels within the object. If the position of R is $\mathbf{x}_\mathrm{R}$, pixel i will appear at $\mathbf{x}_i = \mathbf{x}_\mathrm{R} + \mathbf{r}_i$. This means that when a sampling point $\mathbf{x}_\mathrm{s}$ gives a positive indication of an object, the location of its reference point R will be $\mathbf{x}_\mathrm{R} = \mathbf{x}_\mathrm{s} - \mathbf{r}_i$. Thus the reference point of the object is known to lie at one of the set of points $U_\mathrm{R} = \cup_i(\mathbf{x}_\mathrm{s} - \mathbf{r}_i)$, so the knowledge of its location is naturally incomplete. Indeed, the map of possible reference point locations has the same shape as the original object, but rotated through $180°$—because of the minus sign in front of $\mathbf{r}_i$. Furthermore, the fact that reference point positions are only determined within n pixels means that many sampling points will be needed, the minimum number required to cover the whole image clearly being N/n, if there are N pixels in the image. This means that the optimum speedup factor is $N/(N/n) = n$, as the number of pixels visited in the image is N/n rather than N (Davies, 1997).

Unfortunately, it is not possible to find a set of sampling point locations such that the 'tiling' produced by the resulting maps of possible reference point positions covers the whole image without overlap. Thus there will normally be some overlap (and thus loss of efficiency in locating objects) or some gaps (and thus loss of effectiveness in locating objects). Clearly, the set of tiling squares shown in Fig. 4(b) will only be fully effective if square objects are to be located.

However, a more serious problem arises because objects may appear in any orientation. This prevents an ideal tiling from being found. It appears that the best that can be achieved is to search the image for a maximal *rotationally invariant* subset of the shape, which must be a circle, as indicated in Fig. 5(a). Furthermore, as no perfect tiling for circles exists, the tiling that must be chosen is either a set of hexagons or, more practically, a set of squares. This means that the speedup factor for object location will be significantly less than n, though it will still be substantial.

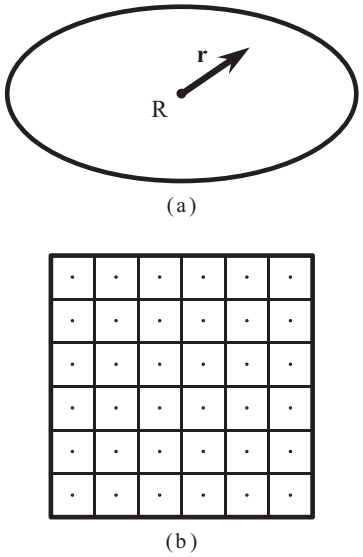


Fig. 4. Object shape and method of sampling: (a) object shape, showing reference point R and vector $\mathbf{r}$ pointing to a general location $x_R + \mathbf{r}$, (b) image and sampling points, with associated tiling squares. ©EURASIP 1998

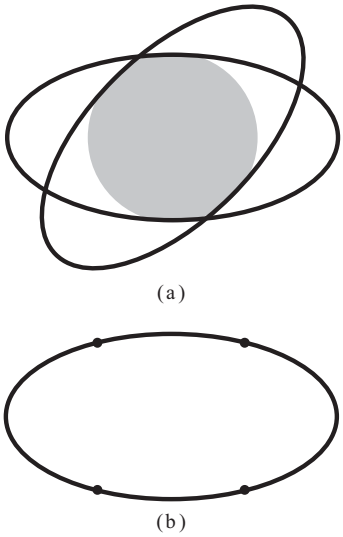


Fig. 5. Geometry for the location of ellipses by sampling: (a) ellipse in two orientations and a maximal rotationally invariant subset (shaded), (b) horizontal ellipse and geometry showing size relative to the largest permitted spacing of sampling points. ©EURASIP 1998

4.2. Application to grain inspection. When applying this technique to the location of wheat grains, it was noted that these grains are quite well approximated by ellipses in which the ratio of semi-major (a) to semi-minor (b) axes is almost exactly 2 (the shape deviation is normally less than 20%). According to the above theory, this means that the (non-ideal) $b \times b$ square tiles have to fit inside the circular maximal rotationally invariant subset (MRIS) of the ellipse, so that $\sqrt{2}L = 2b$, i.e. $L = \sqrt{2}b$.

To understand the efficiency of the process, it is necessary to determine how many sample points could give

positive indications for any one object. Now the maximum distance between one sampling point and another on an ellipse is $2a$, and for the given eccentricity this is equal to $4b$ which in turn is equal to $2\sqrt{2}L$. Thus an ellipse of this eccentricity could overlap three sample points along the x-axis direction if it were aligned along this direction; alternatively, it could overlap just two sample points along the $45°$ direction if it were aligned along this direction, though in that case it could also overlap just one laterally placed sample point. In an intermediate direction (e.g. at an angle arctan 0.5 to the image x-axis), the ellipse could overlap four points. Similarly, it is easy to see that the minimum number of positive sample points per ellipse is 2. The possible arrangements of positive sample points are presented in Fig. 6(a).

In fact, the MRIS rule is over-rigorous. What is actually required is that the sampling tile must be of such a size that all possible orientations of the shape are allowed for. In the present example the limiting case that must be allowed for occurs when the ellipse is orientated parallel to the x-axis, and it must be arranged that it can just pass through four sampling points at the corners of a square, so that on any infinitesimal displacement at least one sampling point is contained within it. For this to be possible it can be shown that $L = (4/\sqrt{5})b$, as depicted in Fig. 5(b). This leads to the possible arrangements of positive sampling points shown in Fig. 6(b)—representing a significant saving in computation.

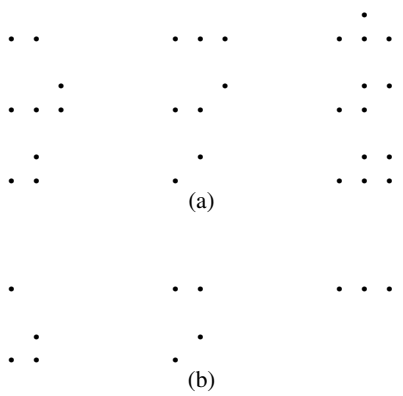


Fig. 6. Possible arrangements of positive sampling points for the ellipse, (a) with $L = \sqrt{2}b$, and (b) with $b = (4\sqrt{5})b$. ©EURASIP 1998

Object location normally takes considerable computation because it involves an unconstrained search over the whole image space, and in addition there is normally (as in the ellipse location task) the problem that the orientation is unknown. This contrasts with the other crucial aspect of inspection, that of object scrutiny and measurement, in that relatively few pixels have to be examined in detail, requiring relatively little computation. Clearly, the sampling approach outlined above largely eliminates the search aspect of object location, since it quickly eliminates any large tracts of blank background. Nevertheless, there is still the problem of refining the object location phase. One way of approaching this problem is to expand the positive samples into fuller regions of interest and then to perform a restricted search over these regions. For this purpose we could use the same search tools (e.g. Hough transforms) that we might use over the whole image if sampling were not being performed. However, the preliminary sampling technique is so fast that this approach would not take full advantage of its speed. Instead, in the wheat grain inspection problem the following *triple bisection* algorithm (Davies, 1998) was used.

Draw horizontal (or vertical) chords through adjacent vertically (or horizontally) separated pairs of positive samples, bisect them, join and extend the bisector lines, and finally find the mid-points of these bisectors (Fig. 7). (In cases where there is a single positive sampling point, another positive sampling point has to be found, say $L/2$ away from the first.) The triple bisection algorithm has the additional advantage of not requiring estimates of tangent directions to be made at the ends of chords (as for some Hough transform implementations (Davies, 2005)), which can prove inaccurate when objects are somewhat fuzzy, as in many grain images. The result of applying this technique to an image containing mostly well-separated grains is shown in Fig. 8: this illustrates that the whole procedure for locating grains by modelling them as ellipses and searching for them by sampling and chord bisection approaches is a viable one. In addition, the procedure is very fast, as the number of pixels that are visited is a small proportion of the total number in each image.

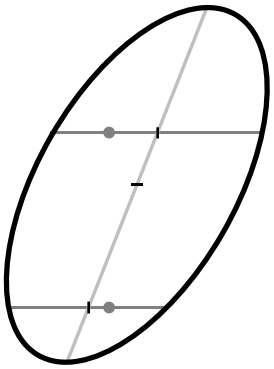


Fig. 7. Illustration of the triple bisection algorithm. The round spots are the sampling points, and the short bars are the midpoints of the three chords, the short horizontal bar being at the centre of the ellipse. ©EURASIP 1998

To see why the triple bisection algorithm presented above is valid, note that because of symmetry it is correct for a circle: this means that it also applies for ellipses, because the circle case can be projected into any elliptic case using orthographic projection—which has the property of

preserving parallelism and midpoints. (For a rigorous algebraic proof, see (Davies, 1999).)

Clearly, the aims of the above work are to reduce, or to see how to reduce, computational loads down to the minimum possible, and above all to identify what the latter really is. Naturally, sparse sampling reduces robustness because redundancy is reduced to a minimum. However, as scanning can be restored to achieve any level of robustness that is required, this should not be regarded as a failure of the method: there will be applications where it can be useful and effective, and equally there will be those where it should not be applied. Failure modes were considered in (Davies, 2001), just as failure modes in the original edge-based sampling method were fully investigated in (Davies, 1987a).

5. Further development of the methodology

Having seen how the basic sampling approach works both in theory and in practice, we are now in a good position to develop it further. First, we note an important aspect of the procedures followed in the earlier work—that the image was always scanned fully at a uniform spatial rate. While the early work tested hexagonal scanning and tiling patterns, rectangular or square scanning patterns emerged as the optimum if no objects were to be missed. Naturally, on an inspection line, 100% untiring inspection is required, thereby dictating that there should be no chance of missing a defective product. However, other situations can arise in industrial applications. For example, brake assemblies or other complex manufactured parts might normally be placed near the centre of the image, and then need to be located exactly. Similarly, if a number of objects are being tracked, their positions might be fairly predictable and again it will be necessary to pinpoint their positions. And when guiding a robot vehicle, the position of the road may need to be confirmed from time to time to be sure that the vehicle is not drifting sideways: this also applies for a fruit-picking vehicle, which will have to be kept on track between rows of trees or bushes. Similar situations apply for many control applications, both in industry and agriculture and transport.

Clearly, when inspection *per se* is replaced by guidance, surveillance and a host of other more complex, less repetitive tasks, the nature of the vision task can change quite significantly. Overall, this means that some information will often be available about *a priori* object placement, and this will need to be taken into account by a visual object location system. This means that we need to generalise the sampling methodology appropriately. Specifically, we need to allow the object probability distribution to vary over the image, rather than simply taking it to be constant, as assumed earlier.

To proceed, we take the *a priori* probability of a sampling point actually hitting an object as $P(x, y)$ for a 2D image. Thus we can cover practical situations—such as objects being more likely to be nearer the centre of the viewing area. We can also cover cases where a previous image has been interpreted and objects within the scene may have moved a small distance by the time the current image has been obtained. In that case, $P(x, y)$ can reasonably be modelled as the convolution of the solution space for the earlier image (viz. a binary probability image where 1s correspond to object blobs and 0s correspond to background) with a Gaussian distribution representing the likely amount of object migration over the intervening time interval.

Next, it is important to note that if $P(x, y)$ is anywhere equal to unity, the outcome is certain, and there is no value in sampling at that position, because nothing is learnt by doing so. Similarly, if $P(x, y)$ is close to unity, very little will be learnt. Contrariwise, there will be little point in searching where $P(x, y)$ is close to zero, as the greatest likelihood is that nothing will be found. Overall, we can see that the greatest gain in certainty (amount that can be learned) is obtained by looking where the following function is a maximum:

$$F = P(1 - P). \tag{3}$$

The justification is that the amount learned will be proportional to $1 - P$, while the probability of learning it will be proportional to P, so F represents the best estimate of the amount that can be learned. In fact, this argument forgets the amount learned (P) when a sample finds nothing, this being learnt with the probability $1-P$. Thus we could correct (3) by doubling the result. However, as we are seeking the maxima of F, this does not substantively change the situation. Similarly, if entropy were used instead of the form of F given above, the overall effect would again be insubstantive.

At this point we have essentially unified the body-based and edge-based approaches to object location. Specifically, if P is everywhere much less than 0.5, F will have the same form as P, and its maximum will be at nearly the same position as for P (Fig. 9). i.e. for small P:

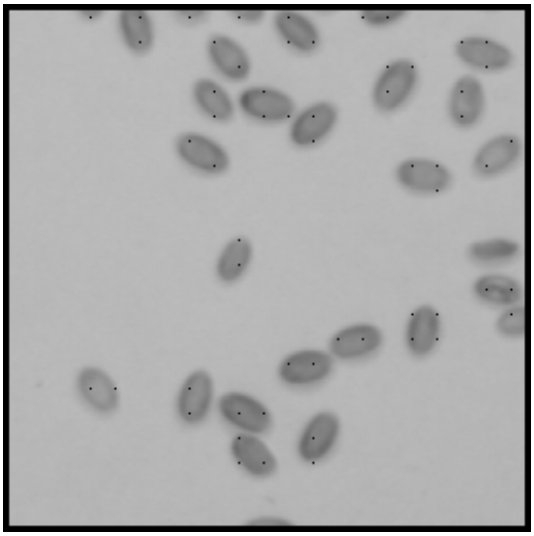
$$F \approx P. \tag{4}$$

This leads to the body-based solution being optimal. However, when there are objects whose positions are fairly well known, so that $P > 0.5$ in their locality, we have the situation shown in Fig. 10, which has peaks around the object edge locations. In this case the edge-based solution is optimal.

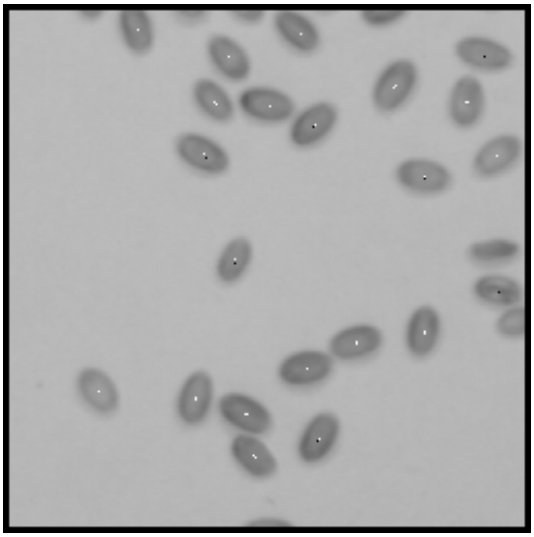
The sampling strategy is now to sample at a succession of locations giving the highest priority to those with the highest values of F. However, we need to take account of the fact that as each sample is taken, the probability P changes and thus F changes: this is what we study next.

For each sample, if no object point is found, we can set P to zero. If an object point is found, this means that

(reverting again to the 1D case) an object of width W is known to be within a distance $\pm W$ of the particular sampling point, and we can set P to a linearly reducing value $1 - |\delta x|/W$, where δx is the distance from the sampling point (Fig. 9). (This is reasonable as P must be zero at a distance $\pm W$: see Section 6 for a detailed proof.) Outside the range $-W$ to $+W$, the sampling point will give no information on how to update P, so we leave it unchanged at the previous *a priori* value. Note that, to eliminate the discontinuities (see the lowest trace in Fig. 9) resulting from this way of updating P, the probabilistic analysis must be changed to take account of interactions between objects: this point is discussed in greater detail in Section 6.



(a)



(b)

Fig. 8. Image showing grain location using the sampling approach: (a) sampling points, (b) final centre locations. ©IET 1999

Subsequent samples may relate either to new objects or to those that have already been located. Here we start by considering the latter case. First, note that the linearly reducing probability value $1 - |\delta x|/W$ will lead to maximum values of F where $P = 0.5$, i.e. where $|\delta x| = W/2$. Testing at each of these points will result in a new zero of P (in the case of no object) or a new position where $P = 1$ (new point on the object). In the former case, the linearly reducing value of P becomes steeper as it has to go from 0 to 1 in a shorter distance (Fig. 11). Also, the flat tops in the third and fifth traces in Fig. 11 arise because if a single object of width W is known to pass through two sampling points a distance $D < W$ apart, the probability of hitting the object must be unity everywhere between the two sampling points. Likewise, if no object is detected at two sampling points a distance $D < W$ apart, there is no chance of the object lying on or between the two sampling points and the probability of hitting it must be zero at all intervening points. The overall result is that after a series of iterations, the samples get closer and closer to the true edge positions of the object, so, in effect, a binary search tree has been implemented using the guided sampling technique. Notice, however, that this capability did not have to be explicitly programmed into the technique: rather, it arose spontaneously as a special case of the general principle of selecting sampling points with maximum F.

Ultimately, the technique finds all $2n$ edge locations for n objects, and iterations need to be continued until the accuracy of location is sufficient or the limits of image resolution are reached.

Next, we consider what happens when subsequent samples refer to new objects rather than those that have already been located. In particular, we consider the combining rules. If two sampling points are a distance $D < W$ apart, and both register a 'hit', there will now be reduced the probability of hitting an object point between these locations because of the reduced probability P that either

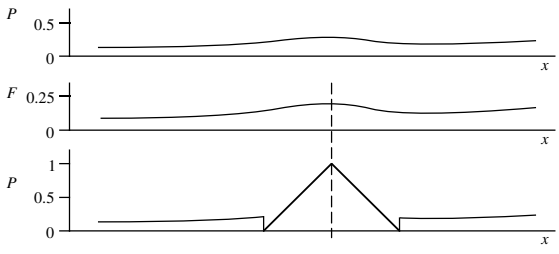


Fig. 9. Case where object locations are not well known. In this 1D model, the top trace represents the *a priori* probability of hitting an object pixel, and F is the likely amount learnt by sampling. The lowest trace shows the result of sampling at the position of maximum F. Note that $F \approx P$ for the top trace, as per Eqn. (4). ©IET 2007

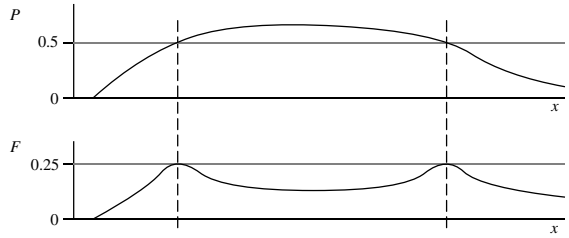


Fig. 10. Case of an object whose location is quite well known. Notice the two positions near the edges of the object where F has the maximum value of 0.25. ©IET 2007

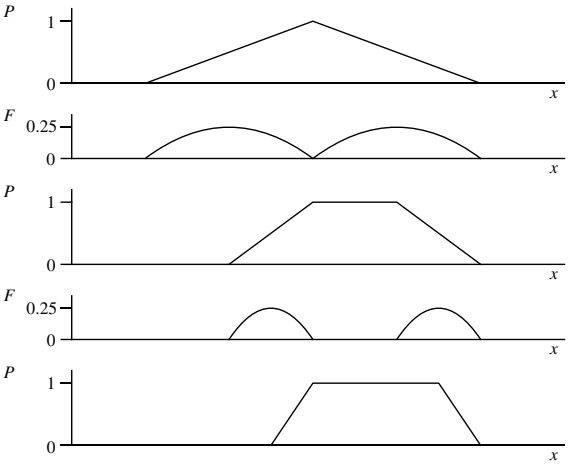


Fig. 11. Case where an object has been found and its exact position needs to be determined. The top trace shows the initial uncertainty in position; this is refined twice by the third and fifth traces; between each pair of traces, the maxima of F are used to determine the optimum sampling positions. In this example, at each test, object location is negative on the left and positive on the right. ©IET 2007

object will pass through any intervening point—as indicated by the linearly reducing probability function. (A rigorous proof requires first calculating the probability $1-P$ of not hitting either object at the point in question, so we have $P = 1 - (1 - P_1)(1 - P_2)$, where P_1 and P_2 are the individual linearly reducing probabilities.) However, the probability of hitting an object between two sampling points a distance $D < W$ apart at each of which no hits are recorded remains at zero, as no object of width W can lie completely between these locations. The combining rule for two sampling points a distance $D < W$ apart when one records a hit and the other does not must be a probability function that changes linearly from 1 to 0 between the two respective locations. This is because the number of possible object positions that give rise to the probability function steadily increases from zero at the $P = 0$ position. (For a more rigorous procedure for performing this type of calculation, see Section 6.)

6. Detailed analysis

At this point we return to some of the ideas outlined above with a view to placing them on a more secure theoretical footing.

6.1. Proof of the formula for the linearly reducing probability. First, we consider the linearly reducing probability $1-|\delta x|/W$ quoted in Section 5. Here we need to examine the possible 'microstates' of the system—specifically, all possible positions of the object of width W pixels. By placing the profiles in all possible positions and averaging at each pixel, we can assess the probability of occupation of all relevant pixels. The result is the convolution of the object profile with the identical shape representing the possible positions of its centre. Thus we convolve two identical rectangular profiles of length W and obtain a triangular profile of length $2W$. This immediately proves that the probability function has the stated form. Note that this is not a probability distribution in the normally accepted sense, as it does not integrate to unity: it gives the 'spot' probability that any pixel is within the boundary of an object.

6.2. Variation of *F* for the linearly reducing probability. Next, we examine the shape of the F profile resulting from the triangular P profile. Substituting for P in (3) gives

$$F = \frac{|\delta x|}{W}\Big(1 - \frac{|\delta x|}{W}\Big) = \frac{|\delta x|}{W} - \frac{|\delta x|^2}{W^2}. \qquad (5)$$

This gives an inverted parabolic shape whose value is zero when $|\delta x| = W$ and when $|\delta x| = 0$, corresponding to P being 0 and 1 respectively—as indicated in various levels of Fig. 11. The parabolic shape has a maximum when $|\delta x| = W/2$, at which point its value is 0.25.

6.3. Eliminating discontinuities in probability estimation. Case when occlusion is possible. We now return to the discontinuity in the bottom trace of Fig. 9. First, suppose that objects in the field of view can occlude each other. In this case, each microstate in our calculation consists of the steady background probability, interrupted by a region of (near) certainty. Averaging all the microstates gives exactly the background level B plus the convolution of two regions of width W and respective heights $1 - B$ and 1; hence we get a triangle of height $1 - B$ on top of the background level B. The result is that the overall probability ranges from 1 down to B, with no discontinuity between the two regions, as indicated in the middle trace of Fig. 12—and unlike the situation shown in the bottom trace of Fig. 9. (A formal proof of this result is obtained by taking the probability $(1 - B)(1 - P)$ that no object is present and deducing the probability that at least

one object is present.) Finally, note that there will still be a discontinuity of gradient between the two regions.

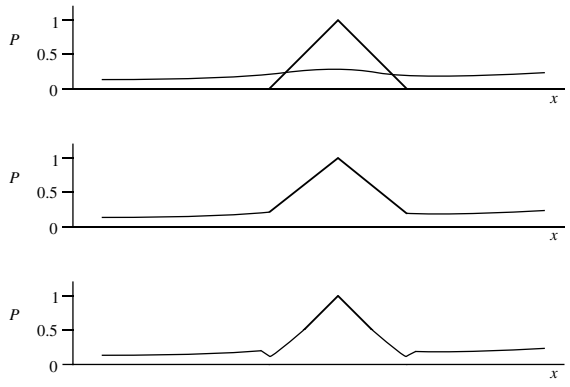


Fig. 12. Formation of the combined probability function. (top) Two functions to be combined. (middle) Combined function for the case of occlusion. Notice the reduced gradients of the central two lines. (bottom) Combined function for the case when occlusion cannot occur. Here the central two lines have almost the same gradients as for the original triangle function.

Case when occlusion cannot occur. Another scenario is when objects are never allowed to overlap or occlude—as for a monolayer of pennies on a counting board—though they may touch. This is a more complicated problem to handle. To solve it we take one central object microstate and average the microstates for all other objects. Then we average all the resulting microstates to give the overall probability function. The result is quite similar to the probability function for the occluded case, but the probabilities are reduced near the central object, because of the restriction that no objects can partially occlude or be occluded by it: hence there are slight dips in the function, as sketched in the bottom trace of Fig. 12. The full theory for this case will be presented in a future publication.

6.4. Case when object width differs from expected width. Finally, we consider the case where objects of width W are being sought, but the actual width of an observed object is W'. First suppose that W' is greater than W. Then a new version of Fig. 11 will be spelled out with final total width equal to at most $2W$: hence if $W' > 2W$, the whole of the object will not be detected, and the length will be mis-measured. However, the very fact that the final length is apparently equal to $2W$ will signal not only that something has gone wrong but also that the length is greater than supposed, so a further search for the ends of the object can be instigated. Indeed, it could reasonably be assumed that two objects are touching, so the true distance from end to end of the combined object is $2W$, and this hypothesis could be checked. In this case the execution time for convergence will be more than double that for Fig. 11.

Next suppose that W' is less than W. In this case there will be no problem in finding the exact positions of the ends, and the method illustrated in Fig. 11 will give the required result. In this case the speed of convergence will be exactly the same as for Fig. 11.

To summarise, the situation is that the measured width W'' is equal to the actual width W', unless $W' > 2W$, in which case $W'' = 2W$. (For proofs of these results, note that the maximum edge distance relative to the positive sampling point is $\pm D$, where $D = (1 + 1/2 + 1/4 + \ldots)W/2$, which is equal to W, so that the maximum object width that can be estimated without error is $2W$.)

Overall, if W' is likely to be greater than W, this factor could be incorporated into the methodology by making W equal the maximum likely length. In particular, if two objects could be touching, but three touching objects will be a rare occurrence, then it may be best to make $W = 2W'$. Once the basic situation has been clarified, and the various likelihoods evaluated, it should be straightforward to determine how the system can be optimised using the methods described above.

7. Assessment of probability in 2D

The assessments of probability for optimum sample placement that have been presented in the previous two sections all represent the situation in the 1D case. It will next be necessary to extend the ideas to 2D ready for practical application to real images. Unfortunately, the situation is not so tidy in the 2D case. In particular, the combining rules for pairs of nearby samples are more complex (though they necessarily devolve into 1D cases along lines joining pairs of samples).

There are three cases of note: (1) the case of two negative samples, (2) the case of two positive samples, and (3) the case of a positive and a negative sample. In all of these, it is easiest to consider first where the object centre could be after obtaining each sample; then, one can convolve the position possibilities with the object shape relative to the centre, in order to determine the probability of hitting a point on an object.

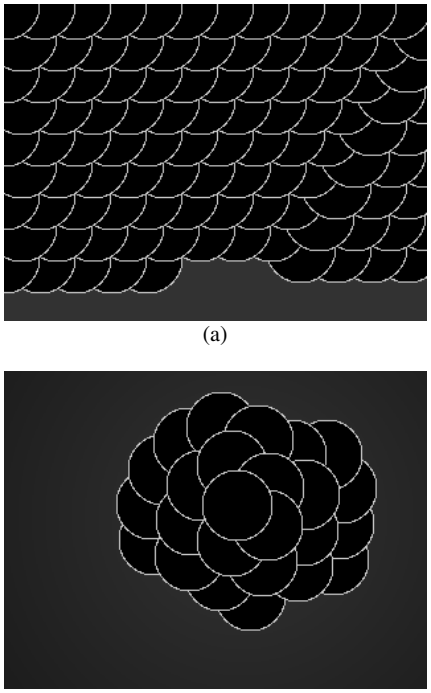
For the case (1) with a circular object, we get two circular regions which overlap, and in the overlap region there is zero probability of finding an object point. Here we have developed an alternate approach in which the rule for combining two negative samples ignores the fact that in the overlap region the probability should be zero: instead we compute the probability using the linearly reducing type of formula (see Section 6.1). In fact, this will not matter in one full pass over the whole image, because the probability only guides the sampling and is programmed to utilise the highest available value of F: this means that

it will not matter whether P has been reduced from B to zero or merely to some other value less than B. The problem only emerges later, when no locations remain with the default probability B, and only then will the exact values assigned to them dictate how ideal the guiding actually is. The result is that later in the sequence *reasonable* sampling points will be provided rather than absolutely optimal sampling points. This will never prevent an object from being found, but in some cases may delay it. However, the average time to find an object will be little impaired. Here a lot depends on the aims of the search: in some cases (as in inspection), it will be necessary to guarantee finding all objects; in other cases (such as assembly), it may only be necessary to locate a handful of objects such as washers; and in yet other cases, it will be possible to stop once one object of a certain type is found (even in inspection this could occur, as when a single insect might render an entire batch of grain unacceptable). Certainly, in the last instance, what might happen late in a sampling sequence would matter very little. However, in the first instance, where every object has to be located, there seems to be little to be gained from using the probability formalism: it is bound to be better to use a regular scan that is guaranteed to locate each object in a single pass over the whole image.

With this background, we have developed the less intensive approach of (a) using the simplified combining rule for the case of two negative samples, and (b) locating an object completely using a 1D binary search tree approach as soon as a positive sample is obtained. (Recall that the binary search tree was shown to be equivalent to the probabilistic approach in the 1D case.)

7.1. Results. First we consider how background sampling progresses in the absence of any objects. This provides an effective comparison of guided sampling with the regular sampling used in the previous work (Davies, 2001). In the earlier work, samples were taken in a regular square array, but the increase in efficiency for a hexagonal array was also investigated. Clearly, packing should be significantly better for a hexagonal array, so fewer samples would be needed to cover an image, with the result that greater execution speeds should be attained. In fact, when searching for moderately large objects, it was found that the gain from using hexagonal arrays was largely lost because of bad fit around the boundary of the image, and so extensive work using hexagonal arrays was curtailed (it appeared that considerable effort would be required to achieve worthwhile gains when using them in practical situations).

What is interesting here is that the guided sampling procedure resulting from Equation (3) led automatically not only to an intrinsic hexagonal array (top image in Fig. 13), but also to reasonably optimal allowance for image boundary effects, making this a potential practical alternative to regular sampling. In fact, the guided sampling locations can be computed offline and used to replace the regular sampling patterns characteristic of the earlier work—necessitating no additional run-time computation.



(a)

(b)

Fig. 13. Sampling patterns obtained during null searches: (a) case of the uniform *a priori* probability: first 105 sampling points, (b) case of the slowly varying *a priori* probability: first 25 sampling points. In both cases the black regions indicate where *a posteriori* probabilities have been calculated. The white borders are purely graphics demarcating the probability regions; they also indicate which scanning points were performed after which others (paradoxically, the earlier ones appear to lie on top of the later ones). Part of the remaining original background level, including in (b) its variation, is also visible.

The lower image in Fig. 13 illustrates what happens when a very slight (20%) variation in *a priori* probability is instituted in order to guide the algorithm towards one portion of the image. For example, it could be imagined that a person is heard talking and his/her voice appears to come from a direction slightly to the right: then attention can be drawn to this approximate region of the image so that his//her face can be focussed upon and watched carefully. Notice that the scanning pattern has now taken place in a nearly spiral scan, and that the resulting scan pattern is approximately hexagonal, though this time with a fair degree of randomness imposed upon it. This has happened completely autonomously, and in an optimal manner in

the sense that an optimal criterion has been used to select each scan point in the sequence.

Interestingly, the slight apparent randomness of the scanning behaviour observed here is reminiscent of the saccadic scanning patterns of the human eye when focussing on details in a visual scene (Palmer, 1999).

Next we consider an object location task, using: (a) a plain (no preference) background probability function, aiming to locate all the objects in the image, and (b) a localised scan where approximate location information is provided by an *a priori* probability map. The results are shown in Fig. 14. In Fig. 14(a), notice how the presence of the object results in slight disruption of the subsequent sampling pattern. In Fig. 14(b), notice that the object is detected using just six samples, and the main sequence then stops, though exact location using the binary search tree technique (in this case employing two linear 1D scans) takes another 12 samples.

8. Conclusion

This paper has investigated how a generalised sampling strategy can be developed for rapid location of objects in digital images. It has found how *a priori* information on the possible locations of objects can be brought into play in a probabilistic formulation which determines how to guide the sampling process. As a result it is shown how body-based and edge-based approaches emerge automatically on applying the right *a priori* probability maps, while the limitations of the regular sampling technique used in the previous work have been clarified. Indeed, in that case the probabilistic formalism has been found to lead to improved sampling patterns that take better account of the positions of the image boundaries. This means that improved speeds of operation can be achieved both in the cases where the whole image has to be scanned in order to locate all the objects, and also in other cases such as where the position of a single object has to be updated. It is also interesting that the new technique is able to carry out full binary search tree edge location without explicit programming.

While in 1D the sampling procedures can be characterised relatively easily, in 2D they become more complex: to contend with this, it has been necessary to develop a less intensive approach, (a) using a simplified combining rule for the case of two negative samples, and (b) locating an object completely using a 1D binary search tree technique as soon as a positive sample is obtained. Overall, this approach maintains the spirit of the probability concept, remembering that its function is to guide object location in such a way as to minimise computational load.

Here, we have not followed the attention-based approach of Itti *et al.* (1998) (amongst others) because, in such approaches, the image is first scanned to locate salient features, which itself consumes significant computation. In contrast, the aim of the technique developed here is to visit relatively few pixels in any image, thus cutting computational load by a large factor. Thus our method is a different type of technique that should find use in the cases where alternate methods are less suitable.

Acknowledgement

The author is grateful to Research Councils UK for the grant GR/R87642/02, awarded under the Basic Technology scheme. Figure 3 is reproduced from (Davies, 1987a) with permission from Elsevier. Text and Figs. 4–7 are reproduced from (Davies, 1998) with permission from EURASIP. Text and Figs. 1 and 8–11 are reproduced from (Davies, 1999; Davies, 2000a; Davies, 2007) with permission from the IET.

References

Ballard D.H. (1981). Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* **13**(2): 111—122.

Davies E.R. (1987a). A high speed algorithm for circular object location, *Pattern Recognition Letters* **6**(5): 323–333.

Davies E.R. (1987b). Lateral histograms for efficient object location: Speed versus ambiguity, *Pattern Recognition Letters* **6**(3): 189–198.

Davies E.R. (1988). A modified Hough scheme for general circle location, *Pattern Recognition Letters* **7**(1): 37–43.

Davies E.R. (1997). Lower bound on the processing required to locate objects in digital images, *Electronics Letters* **33**(21): 1773–1774.

Davies E.R. (1998). Rapid location of convex objects in digital images, *Proceedings of the European Signal and Image Processing Conference (EUSIPCO'98)*, Rhodes, Greece, pp. 589–592.

Davies E.R. (1999). Algorithms for ultra-fast location of ellipses in digital images, *Proceedings of 7th IEE Int. Conference on Image Processing and Its Applications, Manchester, UK*, pp. 542–546.

Davies E.R. (2000a). Low-level vision requirements, *Electronics and Communication Engineering Journal* **12**(5): 197–210.

Davies E.R. (2000b). *Image Processing for the Food Industry*, World Scientific, Singapore.

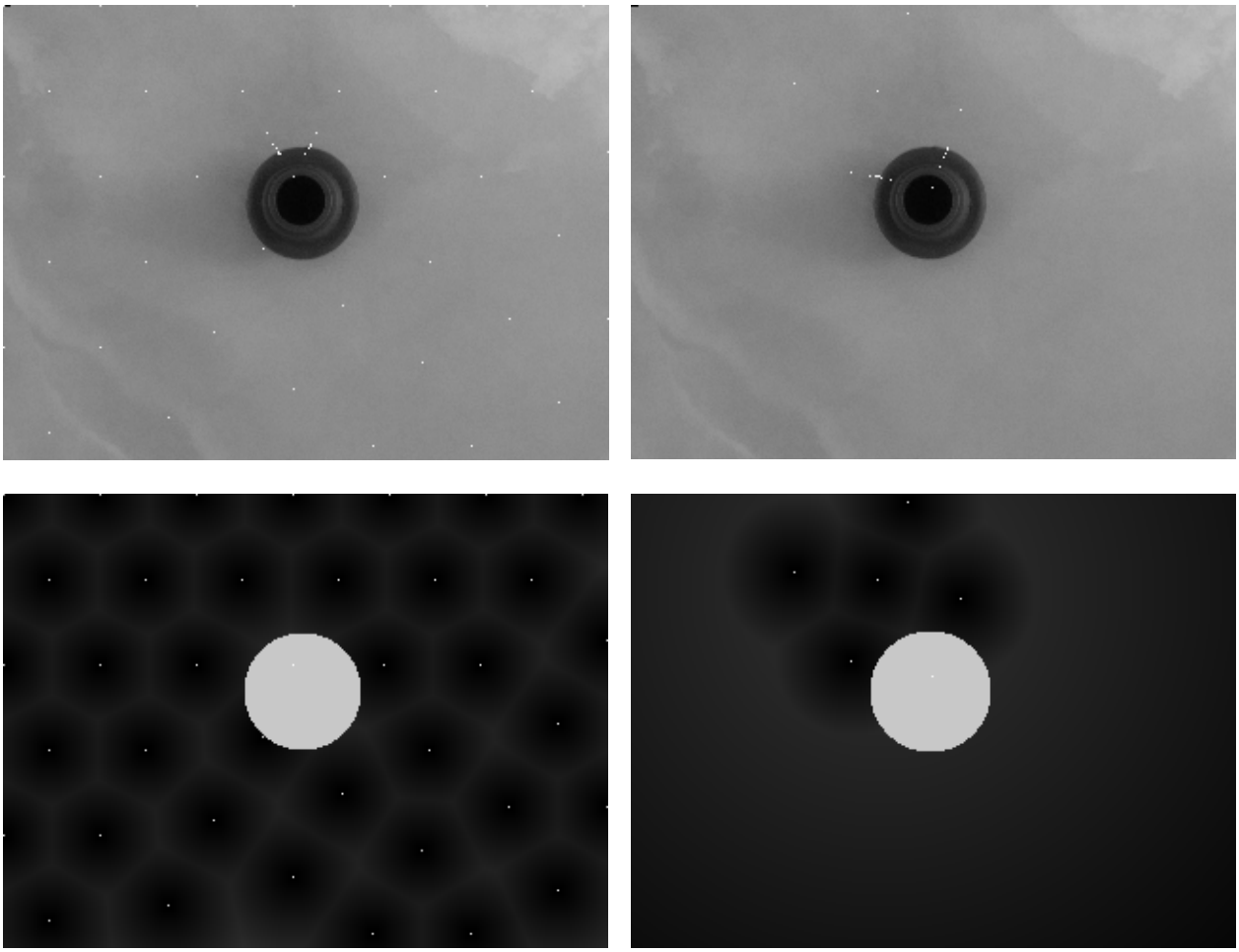
Davies E.R. (2001). A sampling approach to ultra-fast object location, *Real-Time Imaging* **7**(4): 339–355.

Davies E.R. (2003). Design of real-time algorithms for food and cereals inspection, *Imaging Science* **51**(2): 63–78.

Davies E.R. (2005). *Machine Vision: Theory, Algorithms, Practicalities, 3rd*, Ed. Morgan Kaufmann, San Francisco.

Davies E.R. (2007). Guided sampling for rapid object location using biologically motivated model, *Electronics Letters* **43**(9): 508–510.

Itti L., Koch C. and Niebur E. (1998). A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11): 1254–1259.



(a) (b)

Fig. 14. Examples of object location. Here an image containing a round object is analysed (a) assuming a plain background probability function, and (b) assuming a preference for the top left of the image. The upper images show the object and all the sampling points used to locate it. The lower images show the final probability maps and the main sampling point positions. In (b) the final probability map shows the varying background level over most of the image.

Nagel R.N. and Rosenfeld A. (1972). Ordered search techniques in template matching, *Proceedings IEEE* **60**(2), 242–244.

Palmer S.E. (1999). Visual selection: Eye movements and attention, In S.E. Palmer, (Vision Science: Photons to Phenomenology), Bradford Books/MIT Press, Cambridge, MA, pp. 519–571.

Rosenfeld A. and VanderBrug G.J. (1977a). Coarse-fine template matching, *IEEE Transactions on Systems, Man and Cybernetics* **7**(4): 104–107.

VanderBrug G.J. and Rosenfeld A. (1977b). Two-stage template matching, *IEEE Transactions on Computers* **26**(2): 384–393.