# Accurate Boundary Location from Motion

J.A. Marchant
Agricultural and Food Research Council,
Silsoe Research Institute,
Wrest Park, Silsoe, Beds. MK45 4HS.

## 1 Introduction

The ability to monitor a visual scene containing animals and to draw intelligent conclusions automatically would have a significant impact on agricultural practice. For example, if the gait of an animal could be objectively measured, early detection of lameness would be possible. If the motion of a sow and piglets could be analysed, a stockman could be alerted if the piglets were in danger of being crushed or were not feeding properly.

This work forms part of a programme to estimate the weight and hence growth rate of animals from images. In this case, accurate boundaries are required. Animals are often found in visual situations where the background is cluttered and cannot easily be controlled. Also their own surface is often marked either naturally or by contamination from their environment. Segmentation techniques based on thresholding are usually not successful but it may be possible to exploit the fact that animals move whereas the background is stationary.

## 2 Related work

Methods which seek to segment objects using their motion usually rely on an estimate of the object motion itself. Motion estimates for parts of the image can be gained by correlating between small windows in a pair from an image sequence [1]. Correlation can be done in the spatial domain where a window is fixed in position in one image and moved in the other until some measure of correspondence is maximised [2]. Alternatively, the process can be done in the frequency domain using the Fourier transform which is generally faster and suited to modern special purpose hardware. There is a basic problem in using any technique in which a finite sized window is used -estimates near the boundary of the moving object (the very places which require accurate location in this work) will be poor.

In principle this problem could be overcome by using differential techniques for motion measurement (e.g. [3]). However, these techniques are affected greatly by noise problems. Also it can be shown that no information can be obtained on the motion component parallel to an edge feature unless extra assumptions are made concerning the form of motion.

Murray et al. [4] exploit the fact that functions of the image intensity and its changes can be chosen which vary rapidly across object boundaries. However, the method still depends on using a finite sized operator to detect peaks in these functions. Rivero and Bouthemy [5] also use differential methods for motion estimation and collect together regions having similar motions. The size of these

BMVC 1992 doi:10.5244/C.6.10

regions are smaller near to object boundaries but still of a sufficient size to give a very "blocky" appearance to the edge.

In the work reported here a correlation technique is used to avoid noisy motion estimates. Poor estimates near to boundaries are avoided by using a motion model derived with a robust estimator. An accurate and reasonably complete boundary is then built up by integration of successive estimates over a motion sequence.

## 3 Outline of method

The method starts by finding the area of significant change in an image pair by differencing and thresholding. Such an image pair and the changed area is shown in Fig. 3.1. The changed area contains components from:

a) where background has been uncovered by the object,
b) where background has been covered up,
c) where object pixels have been replaced by other object pixels at a significantly different grey level,
d) noise giving rise to isolated pixels.

These changes occur over an area generally larger than the moving object and the changed area contains many missing pixels where a grey level change is within the threshold. Some of these missing pixels can be filled in by a number of dilation operations followed by an equal number of erosions.

As pointed out by Ostermann [6] the boundary of the moving object can be found by combining the changed area with a knowledge of the object movement as follows:

Calculate the motion vector for each point in the changed area (see below). Place the tail of the vector on each pixel in the changed area. If the head is within the changed area (i.e. the head is also on a changed pixel), the head point is on the moving object. Note that if the motion vector is other than zero, isolated noise points will be removed provided there is no second noise point at the head of the vector.

As the changed area is an incomplete representation of the areas where motion has occurred, this procedure gives an equally incomplete version of the moving object (Fig. 1).

In order to give a more complete rendition of the object the method is applied to a sequence of images. As the method proceeds, an "object mask" is maintained which is a binary image, grey level 255 signifies that particular pixel is on the moving object, level 0 signifies background. As each new image pair is analysed, the existing object mask is "warped" by moving each pixel by the calculated motion vector. Then new points are added to form the new object mask. Thus the object mask tracks the object through the image sequence and becomes more complete in the process. The underlying assumption is that object pixels which do not change significantly at any one iteration of the method (and thus do not form part of the changed area) will change significantly at some other stage in the sequence. To recover the moving object from any image in the sequence the object mask is simply combined with the image by a logical AND.

## 4 Calculation of motion vectors

The early stages of motion measurement follow the work reported by Burt et al. [1]. Motion vectors are measured by correlation between small windows in the image pair. A fast Fourier transform is used to perform the correlation using a window size of eight pixels square. A coarse to fine procedure limits the motion at each level of the procedure to a few pixels. A pyramid of images is formed , each image being half the resolution of its parent. Some care must be taken when reducing resolution in order to avoid aliaising of frequencies which are present in the finer resolution image but above the Nyquist frequency at the lower resolution. The author convolved each image in the pyramid with a filter having three zeros at frequencies at and above the Nyquist frequency before sampling the filtered image (Appendix).

The sequence analysis depends on an accurate knowledge of the motion of each part of the object. In order to locate boundaries accurately (a major objective of this work) the information must be available at the boundaries of the object. However, these areas are also the points where correlations will be poor and motion estimates inaccurate. To avoid this problem, a motion model is used.

Following Burt et al. [1] the variation of motion across the object is explained by assuming the object to be a rigid body moving with six degrees of freedom in three dimensions. Thus a model for the coherent motion of the object can be obtained by fitting two functions, one each to the $x$ and $y$ components of the motion, of the form:

$$v_x = ax + by + c \quad \text{..... (1)}$$
$$v_y = dx + ey + f \quad \text{..... (2)}$$

Because of boundary effects, the raw motion data will contain a significant number of outliers. Three methods have been used to combat this problem.

1 After dilating and eroding the changed area (previous section) the area is further eroded to remove from the boundary a width equal to approximately half the window size used for correlation. This new area becomes the basis for raw motion estimates although the original area is retained to calculate the object mask.

2 The cross correlation for each motion estimate is normalised by dividing by

$$\left(\sum g^2 \sum f^2\right)^{0.5}$$

where g is the grey level in one window, f is the grey level in the other window at maximum correspondence and the sums are over the window areas [2]. This gives a value between 0.0 and 1.0. An average value is calculated over all the points in the changed area and only motions for those points above the average are passed on to the next stage.

3 A robust estimator is used to identify the parameters in Eqns. 1 and 2. In a normal least squares estimator it can be shown that each point is weighted according to its distance from the fitted plane. In the estimator used here [7],

a sinusoidal weighting function is used which peaks at a difference of 1 unit and returns to zero at 2 units. Thus points greater than 2 units from the fitted plane are ignored completely. The procedure results in a non-linear minimisation problem which has been solved here using the Simplex method [7]. The starting values for the estimated parameters are gained from a least squares fit.

## 5 Results

The method was used on two types of images. Firstly a random pattern of grey levels between 0 and 255 in a window 64 pixels by 48 which was moved against a second random pattern as background. A random number generator was used to produce displacements between ±9 pixels horizontally and ± 6 pixels vertically. For this test the window motion was confined to a translation in the image plane of a whole number of pixels in each direction.

Table 1 shows the number of edge pixels, object pixels, and background pixels found in two cases; firstly where the changed area was not modified by dilation and erosion and secondly where two stages of each were used. The images were numbered from 0 to 7 and sequential pairs were used for analysis.

**Table I. Performance of algorithm on a random pattern**

| | | dilation/erosion = 0 | | | dilation/erosion = 2 | | |
|---|---|---|---|---|---|---|---|
| image pair | displacement x,y | edge pixels | object pixels | back-ground pixels | edge pixels | object pixels | back-ground pixels |
| 0/1 | -7,-2 | 172 | 2375 | 0 | 220 | 3072 | 0 |
| 1/2 | 9,-2 | 202 | 2875 | 0 | 220 | 3072 | 0 |
| 2/3 | 5, 5 | 215 | 3021 | 0 | 220 | 3072 | 0 |
| 3/4 | 8, 0 | 219 | 3056 | 0 | 220 | 3072 | 0 |
| 4/5 | 4,-6 | 220 | 3070 | 0 | 220 | 3072 | 0 |
| 5/6 | 1,-6 | 220 | 3071 | 0 | 220 | 3072 | 0 |
| 6/7 | 7, 6 | 220 | 3072 | 0 | 220 | 3072 | 0 |
| Total no. of edge pixels = 220; total no. of object pixels = 3072 | | | | | | | |

It should be noted that the problem is made easier by the fact that the warping of the object mask is constrained to give an integer result. As the window was moved by integer amounts, the rounding process removes errors providing they are less than 0.5 pixels.

Table I shows that the algorithm yields the moving window exactly, correctly finding all object pixels (including those on the edge) and no background pixels. With no dilation/erosion of the changed region seven iterations of the algorithm are required. With two stages of dilation and erosion the algorithm finds the window on the first iteration.

In the second set of tests the trunk and head of a person was used as the target object. Two situations were chosen - where the person was wearing patterned clothing against a cluttered background, and where relatively plain clothing was worn against a plain background.

Eight images were used in each sequence numbered zero to seven. Figs. 2 and 3 show image number 7 along with results from image pairs 0/1, 3/4, and 6/7 from each of the two situations. In each case, the movements were a combination of rotations and translations in the dimensions. Some care was taken to ensure that the head/body combination moved as a rigid body i.e. articulation at the neck was kept to a minimum. This restriction was imposed, for this stage of the work, to avoid contravening of the basic assumptions of the method. Note that a second assumption, that the body is planar, was regularly violated. For dilations of the changed area were used followed by four erosions when finding object pixels from motion vectors.

Fig. 2 shows the gradual improvement of the object segmentation throughout a sequence. With the exception of a few isolated background areas, the only significant addition to the object is a small area to the left hand side if the neck. A small part of the boundary on the left shoulder has become slightly ragged. Some areas of the body, notably the forehead and below the neck have been missed. This is due to insufficient texture in these areas giving regions with no significant change in grey level with movement. These areas could be filled in by noting the fact that there are no holes in the real object. As the cause of the problem is lack of texture, a reasonable estimate of the grey level of the holes could be made by averaging the grey levels over the corresponding areas in the whole sequence. However, the performance on the task in hand, finding an accurate boundary, is good. Note that the cracks in the objects are caused by using integer arithmetic in the warping process to produce the object mask.

Where there is less texture in the image (Fig. 3) the performance is worse, as expected. However, with the exception of the area to the right of the neck, the segmentation of the head is good. The ragged boundary to the left and right of the body could possibly be improved by smoothing the boundary direction. As with Fig. 2 the holes (this time much larger) on the body could possibly be filled in with an average of grey levels over the sequence.

## 6 Conclusions

A method has been proposed which can derive an accurate boundary of an object from an image sequence. The method depends on a number of assumptions, in particular that the motion is due to a planar object which has sufficient texture on its surface.

Tests on random dot patterns which translate an integral number of pixels give perfect results when there is sufficient grey level texture but, as expected, poorer results on more uniformly shaded objects.

In order to use the method on more general animal images, a technique will need to be developed to handle objects which cannot be represented as rigid bodies. For instance, those that deform or articulate. Future work will address this problem.

## References

[1] P.J. Burt, J.R. Bergen, R. Hingorani, R. Kolczynski, W.A. Lee, A. Leung, J. Lubin and H. Shvaytser. Object tracking with a moving camera. Proc. IEEE Workshop on Visual Motion, Irvine CA, 1989, pp 2-12.

[2] A. Rosenfeld and A. Kak. Digital picture processing (2nd ed.). Academic Press, Orlando, 1982.

[3] A. Verri, F. Girosi and V. Torre. Differential techniques for optical flow. J. Optical Society of America 1990; 7:912-922.

[4] D.W. Murray and N.S. Williams. Detecting the boundaries between optical flow fields from several moving planar facets. Pattern Recognition Letters 1986; 4:87-92.

[5] J.S. Rivero and P. Bouthemy. A hierarchical likelihood framework for motion based segmentation from image sequences. Proc 5th Scandinavian Conf. on Image Analysis. Stockholm, 1987, pp 623-631.

[6] J. Osterman. Modelling of 3D moving objects for an analysis - synthesis coder. Proc. SPIE Conf. Sensing and reconstruction of 3D objects and scenes, Santa Clara CA, 1990, pp 240-249.

[7] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling. Numerical recipes in C. Cambridge University Press, Cambridge, 1988.

## APPENDIX

### Anti-alias filter for image sampling.

If $g(m,n)$ is the grey level of an image at point $m,n$ then the image can be represented in the frequency domain as $G(u,v)$ where $G$ is the two dimensional discrete Fourier transform:

$$G(u,v) = \sum_{m,n} g(m,n) \exp(-2\pi i(um+vn)/N)$$

$N$ is the image size and $u$ and $v$ complex frequency components.

Consider an image at level $\ell$ in the pyramid of decreasing resolution. This is derived by sampling an image at level $\ell-1$ at every other pixel. The Nyquist frequency at level $\ell-1$ is $\pi$ radians/pixel and so that at level $\ell$ is $\pi/2$ radians pixel.

Note that the pixel dimension for both levels in this explanation is that for level $\ell-1$. Hence, in order to avoid aliasing, all frequencies above $u = v = \pi/2$ should be removed in the image at level $\ell-1$ before it is sampled. This cannot be achieved exactly but an approximation can be made by convolving three masks

$$\begin{matrix} 1\ 1 \end{matrix} * \begin{matrix} 1\ 1\ 1 \\ 1\ 1\ 1 \end{matrix} * \begin{matrix} 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \end{matrix}$$

to give a filter mask

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 6 | 5 | 3 | 1 |
| 3 | 9 | 15 | 18 | 15 | 9 | 3 |
| 5 | 15 | 25 | 30 | 25 | 15 | 5 |
| 6 | 18 | 30 | 36 | 30 | 18 | 6 |
| 5 | 15 | 25 | 30 | 25 | 15 | 5 |
| 3 | 9 | 15 | 18 | 15 | 9 | 3 |
| 1 | 3 | 5 | 6 | 5 | 3 | 1 |

The 4 x 4 filter has a zero at $\pi/2$, the 3 x 3 at $2\pi/3$ and the 2 x 2 at $\pi$, and so the composite filter has zeros at these three frequencies.

Note that the 7 x 7 filter can be implemented by convolving the image firstly with the 1 x 7 filter formed by the first row, then with the 7x1 filter formed by the first column. This procedure speeds up the implementation significantly.

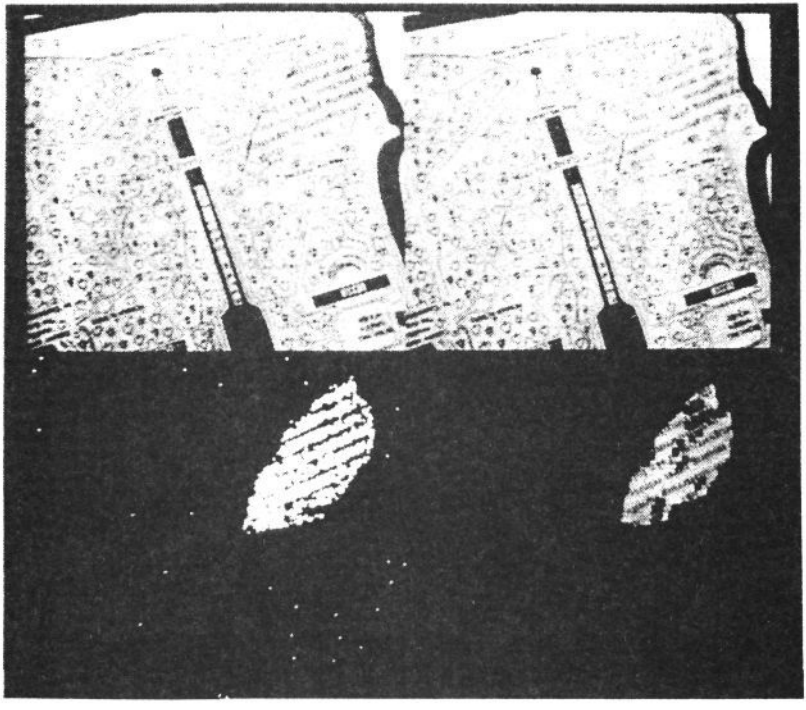

**Fig. 1**

Top, image pair consisting of an area torn from a page of text moving on a similarly textured background. Bottom left, changed region. Bottom right, incomplete rendition of object.

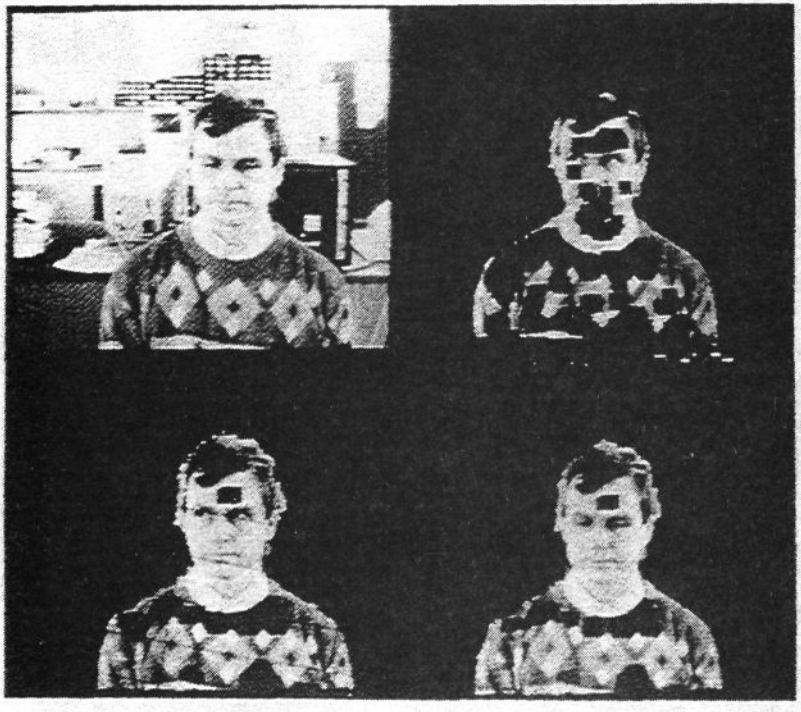

**Fig. 2**

Results for relatively patterned object on a cluttered background. Top left, image sequence No.7. Top right, bottom left, bottom right, improvements of object rendition.

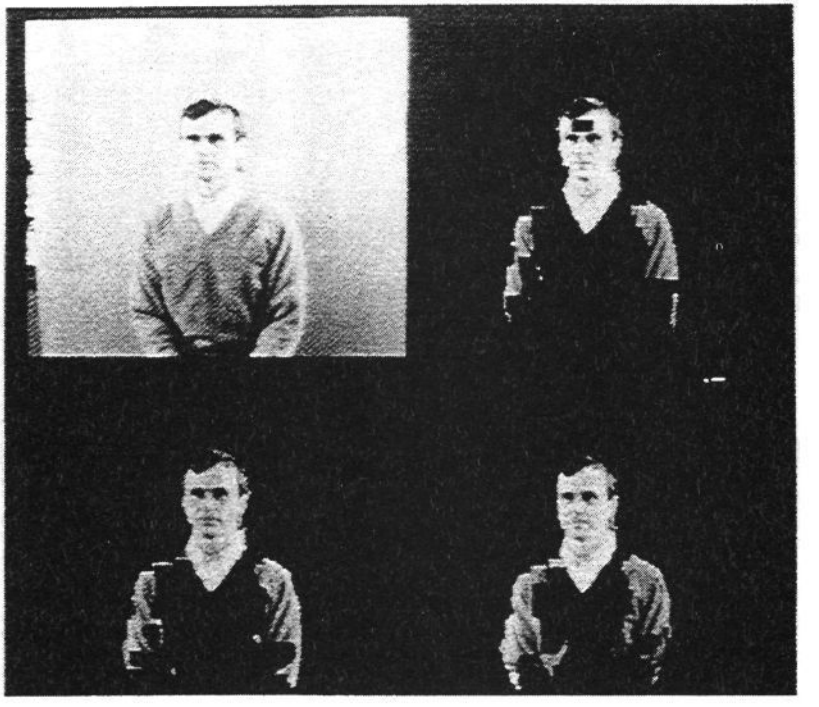

**Fig. 3**

Results for relatively plain object on plain background. Top left, image sequence No.7. Top right, bottom left, bottom right, improvement of object rendition.