Cooperating motion processes

P H S Torr, T Wong, D W Murray and A Zisserman

Robotics Research Group
Department of Engineering Science
Oxford University Parks Road, Oxford, OX1 3PJ, UK

Abstract

This paper describes the use of a low level, computationally inexpensive motion detector to initiate a higher level motion tracker based on an elliptical active contour or snake. The contour tracker is in turn used to direct a camera mounted on a robot arm to track head shaped objects.

1 Introduction

An imperative for any autonomous agent which relies on visual perception is the ability to interpret time-varying imagery.

A cursory glance at the literature would suggest that the issues of primary concern are the computation of an explicit representation of visual motion from the imagery, and its subsequent analysis in terms of structure of the environment and the motion of the camera relative to that environment. A somewhat more careful study, however, would show that there is a range of uses for which motion interpretation may be put at an earlier level than computation of egomotion and structure from motion. Amongst the more important of these are (i) *alarms* — the detection and flagging of things of interest or danger in the image; (ii) *segmentation* — the dividing up of the scene into separate cohesive areas; and (iii) *tracking* — to retain things of interest on the sensor by nulling their motion (this has the additional advantage that background objects are effectively removed by motion blur).

The predominance of the approach of recovery of visual motion followed by structure from motion or egomotion computation has had three unfortunate consequences. First, the three tasks mentioned above are rather under-explored. Secondly, the tasks have often been explored in a narrow way, confined within some existing framework of visual motion and structure from motion algorithms. Thirdly, and in contrast, other often more direct and promising methods have been experimented with in isolation from existing visual motion and structure from motion algorithms.

We suggest that to deal with the rich variety of world tasks that demand motion understanding — involving interaction both with other autonomous agents and with the environment — requires a similarly rich variety of "motion sensors". Such sensors should range from the crude, fast and robust, to the more refined and stately, and should utilize more than the single conventional representation of image motion. The cruder processes can run quasi autonomously, but may act as bootstraps for the more sophisticated ones and, as knowledge sources, should report not only their motion information but also its reliability.

BMVC 1991 doi:10.5244/C.5.19

The proposed use of several parallel motion detectors, or cooperating motion processes, differs from the most successful current motion systems for navigation [5, 11] which, being designed around a single activity, use a single method of recovery of and interpretation of image motion.

In this paper we give a first demonstration of this theme. There are three main stages to consider. The first is event detection. Initially the camera is stationary. A frame differencing method detect "events" which occur in scene, and if judged of sufficient interest the tracker is initialized by placing an ellipse around the detected region of interest. The second is the ellipse tracker itself. The ellipse acts as a blob tracker applied to the image. It tracks points of high spatial derivative around its perimeter. The five parameters of the ellipse are updated using a Kalman filter. Finally, the camera is moved to keep the centre of the ellipse fixed at the centre of the image. These are discussed in more detail now.

2 Event detection

The event detector we use here uses straightforward grey-level subtraction. Ullman [10] noted that although direct use of grey-level operations is inadequate to compute long-range motion in both human and computer vision systems, intensity-based processes were adequate for an early warning system, detecting changes and directing attention. They might also be useful in detecting discontinuous boundaries where velocity in the visual field change abruptly.

For a static camera, image subtraction between successive images of the same scene will act as a high pass filter, provided that the temporal difference is not too great (the sampling theorem suggests the necessary image rate). To reduce the effects of noise it is best to smooth the images by convolving with a Gaussian both spatially and temporally. A key assumption however is that areas of high temporal difference correspond to areas of motion interest. There are of course several problems with this assumption, problems which are well known from more sophisticated gradient-based motion analysis. These are that an object of similar intensity to its background will be almost invisible, sudden changes in lighting, specularities and shadows will produce high frequency change, and the method will run into problem if the ego motion is not zero.

Despite these difficulties Nagel, Jain and coworkers [7] devised several practical methods for grey-level change detection. However by [6] the method had become one of such statistical sophistication that it ran very slowly indeed. The processes gave exceptionally reliable change detection from just two successive frames, but was no longer faithful to its original goal of rapid event detection.

The approach of cooperating processes is rather different. Rather than "improve" the event detection process until it becomes cumbersome, we require it to be fast, but allow it to make mistakes. It is the role of the more sophisticated concurrent process to decide whether the events are worth pursuing.

In our work, simple frame rate image differencing has been used to detect events. Images are captured using a Datacube Digimax and passed through a VFIR-II where they are convolved with a Gaussian. The current convolved image is subtracted from the convolved frame stored from the previous frame time to provide a difference signal, and is also passed to a framestore for use in the next frame time. The

subtracted signal is then subsampled to give a 32×32 image which is transferred to a Sun4 workstation. A grey-level difference threshold is applied to the difference image, and if this is exceeded an event is marked in the event map. In our indoor experiments, a difference threshold of 20 grey-levels has been found suitable.

The elliptical snake expects to track quite large objects, and so a snake is only initialized in response to several nearby events being triggered in the event map. A simple graph colouring algorithm links active events that have active neighbours (these need not be nearest neighbours). If the number of events in a clique exceeds an activity threshold then the tracker is initialized. At present, the smallest rectangle containing all the active grid points is determined and the ellipse is initialized so that it just fits within it.

3 Tracking with an elliptical snake

The use of an ellipse as a blob tracker has a number of advantages over more traditional snakes [8]. It shares with spline based snakes [4, 3] the advantage of a small number of parameters — five — the centre (x, y), half length of major and minor axes (a, b) and the orientation θ. Moreover, because of its structure it does not suffer from the two common failings of snake tracking, viz. (i) part of the snake gets left behind and so the curve straddles front and back of the blob and (ii) the snake crosses itself, or folds and partially collapses onto itself. It is difficult to recover from these situations, and tracking ability is obviously impaired. The principal disadvantage of using a snake structure is, of course, that it is model-based, introducing strong expectations about the scene. If the image area on which the ellipse is initialized is not elliptical, only part of the snake will be comfortably attached to the area's boundary.

Deformable templates based on ellipses were first used in ref [9]. Their ellipse tracker was computationally costly because iterative update of ellipse parameters involved forming a 2D attractor field and integrating the derivative of this around the ellipse perimeter. The update algorithm used here improves on this by using a small number (10–20) of evenly spaced points selected around the ellipse perimeter. From each of these we search up to some fixed distance along the local normal to the ellipse for the nearest image edge. Given the new set of edge points, the new ellipse is fitted using Bookstein's algorithm [2]. This avoids the computation of a 2D attractor field (only a 1D search is needed) and provides a one shot, rather than an iterative, update procedure.

Reducing search with a Kalman Filter

To reduce search further when fitting the elliptical snake we have exploited prediction from a Kalman Filter applied to the five parameters describing the ellipse. For each parameter we run a polynomial filter which assumes constant acceleration of the parameter [1]. The model assumes no correlation between the different parameters, and so, for example, the update condition for x is

$$\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{(k+1)} = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{(k)} + \begin{bmatrix} 0 \\ 1/2 \\ 1 \end{bmatrix} v_{acc} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_{pos}$$

where ΔT is the timestep between updates, v_{acc} is a zero-mean white noise sequence modeling the effects of non-zero rate of change of acceleration ($\frac{d}{dt}\ddot{x} = v_{acc}$) and v_{pos} is a zero-mean white noise sequence which accounts for model errors, particularly motion in jerks. The motion of the target is observed by a sensor which only measures position:

$$z_{(k+1)} = x_{(k+1)} + w$$

where z_x is observation of x, and w is a zero-mean white noise sequence.

4 Experiments

Figures 1–3 show the output from a typical trial run of the entire system.

Figure 1(a) shows the initially stationary head and (b) the result of moving it. The small boxes are the active outputs in the 32×32 event map found as a result of differencing, and the larger box is a bounding box for this group. It is into this box that the snake is initialized.

Figure 2(a) shows the situation some frames later. The snake is well attached to the head outline. The box draw on this image is the search region for the ellipse updating. Figure 2(b) shows the snake remaining attached as the head moves against a *static* background (note that the no-smoking sign remains stationary in the image).

Figures 3(a) and (b) show the snake being used to drive the robot arm holding the camera. The Adept SCARA arm was programmed to only use 2 degrees of freedom: translation along and rotation about the image y axis. The camera moves to maintain the ellipse at the centre of the image. Notice now that as the head is moved it remains central on the image, but the static background moves on the image.

5 Conclusions

In this paper we have argued that motion detection and motion understanding require a rich variety of mechanisms and representations which should behave as knowledge sources. There is a need and — as we demonstrate — a use for very crude robust motion processing which can be used to initiate more sophisticated processes. Although the crude processes will make mistakes, the burden of assessment should be on the more sophisticated processes.

We wish to pursue these ideas to include further processes based on conventional 2D motion representations and to explore how these can interact to drive camera motions.

For the particular pair of processes reported here, there are several further investigations that should be made. Of a routine nature, first, the snake initialization could be improved by using the convex hull of the connected active grid points as initial ellipse fitting data, and second the effect of the Kalman filter covariance matrix on the tracker is as yet poorly charted. More interestingly, a versatile tracker would be created if closed snakes could split and merge. This would allow the effects of occlusion to be taken into account e.g. a snake could spilt in two if two objects in the same vicinity travel in different directions.

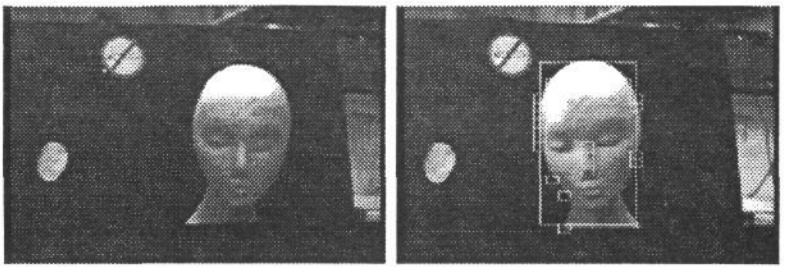

Figure1: *This figure shows the initialization of a window of attention about the head, the small boxes indicate areas of temporal intensity difference, and the bounding box of these indicates the window of attention.*

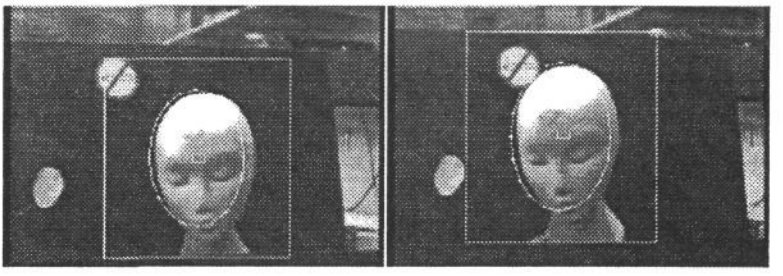

Figure2: *The ellipse is defined by the window of attention. The larger box around the ellipse shows the image search area, the small box marks the ellipse centre.*

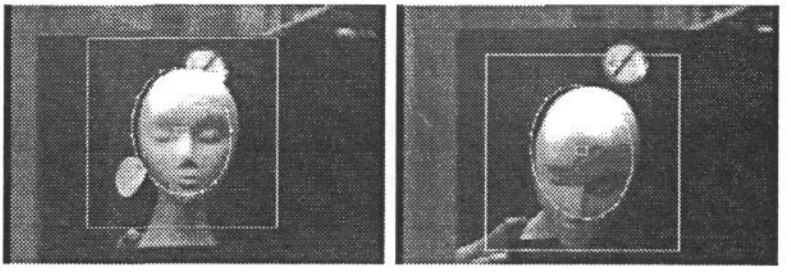

Figure3: *The head begins to move past the no smoking sign and is tracked by the camera.*

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[2] F. Bookstein. Fitting conic sections to scattered data. *CVGIP*, 9:58–91, 1979.

[3] R. Cipolla. *Active Visual Inference of Surface Shape.* PhD thesis, Oxford University, 1991.

[4] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 616–623, 1990.

[5] C.G. Harris and J.M. Pike. 3d positional integration from image sequences. *Image and Vision Computing*, 6:87–90, 1988.

[6] Y.Z. Hsu, H.H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, 1984.

[7] R. Jain and H.H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. Pattern Analysis and Machine Intell.*, 1(2):206–214, 1979.

[8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. Proceedings of the First International Conference on Computer Vision, London, IEEE Computer Society Press, Washington DC pages 259–268, 1987.

[9] Yuille A.L. et al Lipson P. Deformable templates for feature extraction from medical images. In *Proc. 1st European Conf. on Computer Vision*, pages 413–417, 1990.

[10] S. Ullman. *The interpretation of visual motion.* MIT Press, Cambridge,USA, 1979.

[11] Z. Zhuang and O.D. Faugeras. Calibration of a mobile robot with application to visual navigation. Proceedings, Workshop on Visual Motion, (Irvine, CA, March 20–22) pages 306–313, 1989.