Tracking a driver's hands using computer vision

G. McAllister, S.J. McKenna and I.W. Ricketts
University of Dundee
Dundee, DD1 4HN, Scotland
email: gmcallis@computing.dundee.ac.uk

Abstract

This paper describes a computer vision system for tracking a driver's hands. The applications of such a system are discussed with particular attention given to the control of a novel vehicle systems interface. This interface is controlled by pointing at the required function on a display mounted behind the top of the steering wheel. The vision system must detect and track the driver's hands to allow efficient recovery of a search space in which to look for a pointing finger. Cameras mounted above the driver provide images of the steering wheel area. The scene is segmented using adaptive background and foreground models. A distance transform is applied to the resulting contours. A geometric model is fitted to the resulting distance map by maximising an objective function over the model parameters in a local search space defined by the output of a Kalman filter. The appearance-based model fitting process can resolve ambiguous situations such as the hands touching or crossing, and parting again. The system is demonstrated running at 10Hz on a standard PC.

1 Introduction

The use of computer vision to monitor a driver's hands opens up new opportunities for developing innovative or augmenting existing in-vehicle applications. Knowledge of current and recent hand movements coupled with a model of the driver's normal driving patterns may be useful, for example, in helping to predict manoeuvres. It may also be possible to develop a training system where the driver is encouraged to keep their hands at certain points on the wheel such as the 'ten-to-two' position. A variant of such a system could potentially be used to aid the detection of driver fatigue by monitoring hand movement and position on the steering wheel. The application which has motivated this research has been the use of computer vision to control advanced driver-vehicle interfaces.

The ACTIVE project is concerned with increasing safety by improving the driver's interface with the vehicle's non-safety critical controls, including functional groups such as radio and ventilation. The controls are replaced with a display which sits behind the steering wheel. The driver will interact with this interface using pointing gestures. Drivers do not need to remove their hands from the steering wheel nor need they shift their gaze to the same extent as with a traditional dashboard interface. The underlying ideas and safety issues as well as a prototype of the system are described in [2].

The ACTIVE vision system incorporates two cameras mounted in the roof of the vehicle roughly above the driver's shoulders which are directed towards the steering wheel area. A typical view from the right camera is shown in the image sequence given in Figure 1. Ultimately, the vision system's task will be to find a pointing finger in this area and resolve its target on the interface. In order to maintain a real-time response, the search space for the finger must be efficiently focused on the small area where the finger is expected to appear. Focusing attention in this way means that the hands must be detected and tracked so an estimate can be made in each frame regarding the position and size of the search space. The developed model is intended to be simple, intuitive and computationally inexpensive whilst remaining accurate and robust to possibly ambiguous situations such as touching and crossing hands.

Deploying a vision system in a visual environment as dynamic as a car interior poses many challenges. Local lighting changes in the scene, caused for example by shadows, are potentially confusing. The variability between drivers means that the modelling of such visual cues as hand colour, shape and size must be flexible and make as few prior assumptions about the driver as possible. The remainder of this paper is organised as follows. Section 2 gives a brief account of

0-7803-6583-6/00/$10.00 © 2000 IEEE

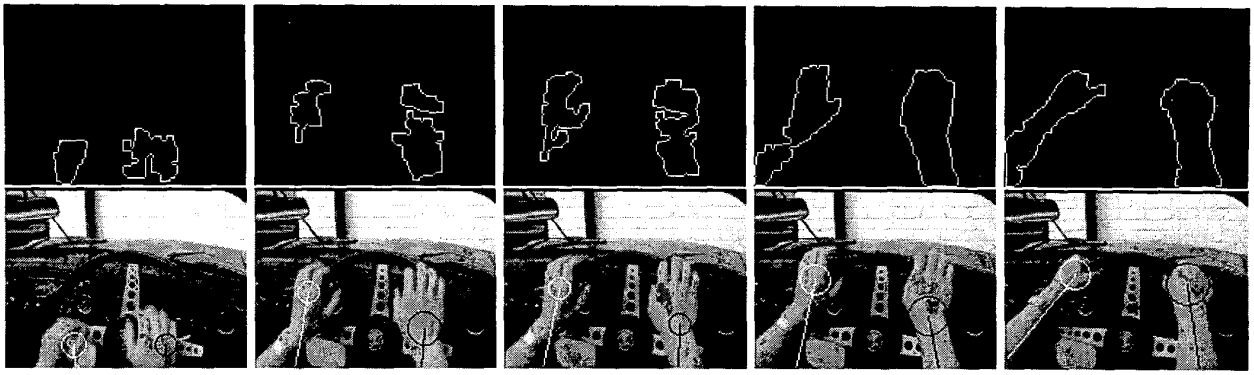

Figure 1: Hands entering the scene

some related work. Section 3 describes the adaptive scene segmentation technique. The methods used for tracking and the problems involved are discussed in Section 4. Section 5 describes the performance of the system in real-life scenarios and some future directions are discussed in Section 6.

2 Related Work

There has been relatively little work published in which computer vision systems have been used to look inside a car. Several groups have used near-infrared (NIR) illumination with a monochrome camera to help control the lighting in the scene: the driver recognition system of Vetter *et al.* [12] attempted to match the current occupant's face to a database of those permitted to drive the car with the face illuminated by an IR spotlight; Nakano *et al.* [6] described a method of illuminating the driver using a pulsed NIR source synchronised with the opening of a shutter on a CCD camera in their blink measurement system to monitor driver drowsiness; Gautama *et al.* [3] described and evaluated algorithms to detect passenger seat occupants and alter airbag firing times accordingly. Park *et al.* described an intruder detection system using a monochrome CMOS camera [7]. Sequences showing an intruder's arm reaching into the car highlighted the problems shadows cause when segmenting an image. They also described some basic tracking with an NIR light source but did not report transferring this into a vehicle. Tock and Craw tackled the problem of driver drowsiness using a colour camera [11]. Their system searched images for chromatic components within a pre-defined range which they had identified as being skin-coloured. No work on tracking drivers' hands in a vehicle has been published to the authors' knowledge.

There has been much work published on the problem of modelling hands for gesture recognition, an overview of which is given by Pavlovic *et al.* [8]. These models range from complex 3D models including skeletal and volumetric models to simple 2D representations such as image moments. Focus of attention in our application does not require the recovery of explicit 3D pose information so there is no need to use computationally expensive 3D modelling techniques. Schemes using rigid templates would tend to be computationally expensive due to the wide range of hand poses the system may encounter (although see Gavrila *et al.* [4]). Deformable template approaches are overly complex for the task of focusing attention in this case. Perhaps the closest hand model to that presented here is described by Sato *et al.* [9]. They used morphology to find the palm region and fit a model in far infrared images for a smart desk application.

3 Scene Segmentation

An intial segmentation of the scene is performed at each frame using background and foreground appearance models that are learned and adapted on-line. This process is given a more thorough treatment in [5] but for completeness a brief description is given here.

The aim of this procedure is to find the regions of interest in the scene, i.e. regions which may be hands or arms. First we define the foreground as the hands and forearms and the background as everything else in the scene, including the steering wheel and dashboard.

A pixelwise background model is initialised with pixel statistics from each image in a sequence of an empty

scene. New images are compared against this model and pixels that differ significantly from the corresponding model pixel have their colour information added to a foreground histogram-based model and are flagged as being possible foreground areas (and hence regions of interest). Initially only the background model is used to search for regions of interest but when the foreground model has been sufficiently populated it is also used. This on-line population of the foreground model is a more robust solution than pre-computed skin colour models since the model adapts to the current situation and makes fewer assumptions about clothing (e.g. the driver can be wearing gloves). When both models are in use, each outputs a likelihood map giving the likelihood that each pixel belongs to foreground or background. These maps are combined using Bayes' rule to give a posterior probability map of foreground regions from which connected components are extracted. A size filter is run on the connected components and the remaining components have their contours traced. The subsequent contour image (see top row of Figure 1) is used as the input to a distance transform algorithm.

4 Hand Detection and Tracking

The detection and tracking of the hand is only one stage in the process of focusing attention to find a pointing finger so it is important that the process can be carried out quickly. However, it must also be robust or there is little to be gained from the effort. To achieve the combination of speed and effectiveness, an appearance-based approach is used consisting of simple geometric shapes and a fitting process which uses the distance transform responses obtained from the segmentation process outlined in the previous section.

4.1 Distance Transform

The distance transform of an image records at each point the distance to the nearest feature point, in this case the nearest contour point. This representation is used for fitting the hand model. The transform has high responses down the centre of the forearm and a spread of high responses in the centre of the hand (depending on its shape). The benefits of this when fitting the model will be explained when the model is described in Section 4.

The distance transform is potentially the slowest single operation in the process so it is important that the calculation can be performed as quickly as possible. The Euclidean distance transform is too expensive to calculate in real-time so an approximation based on the *3–4 Chamfer* distance [1] is used instead. This approximation is sufficiently accurate for the needs of the model fitting algorithm whilst remaining fast enough to satisfy the real-time constraint.

4.2 Geometric Hand Model

The model consists of a circle to describe the palm area and a line segment with an endpoint at the centre of the circle to describe the orientation of the forearm. The model, $\mathbf{h}$, is defined as

$$\mathbf{h} = [x_c, y_c, r, \theta]$$

where (x_c, y_c) and r are the centre and radius of the circle and θ is the orientation of the forearm line. This model was chosen as a simple but reasonable approximation to the hand and forearm shape. Incorporating both the hand and forearm into one model and fitting them simultaneously is very important for dealing with ambiguous circumstances, such as touching and crossing arms. The forearm provides a constraint on the position of the hand belonging to that arm. This constraint is absorbed into the model fitting function as described in subsection 4.5.

4.3 Model Initialisation

When a new region of interest is detected, a new model is instantiated and fitted to it. The new model is initialised by finding the centre of the palm and the orientation of the forearm. The arm is assumed to be moving into the scene hand first so the centre of the circle can be initialised as the point inside the region of interest where a circle with maximum possible radius can be fitted. This point lies where the distance to the nearest contour point is at a maximum. This is also the maximum of the distance transform inside the region of interest. Once the maximum of the distance transform is located, the radius is recovered immediately as the distance transform value at that location. The forearm line segment is fitted by first picking any point on the contour of the region of interest that is also on the side of the image. Starting with this as an endpoint, simulated annealing is used to find the value of θ that maximises the sum of the distance transform values at pixels on the line segment. This function basically determines which line best 'fits' the distance transform response along the forearm. The distance transform is masked so that pixels outside the regions of interest are assigned distance values of zero. At

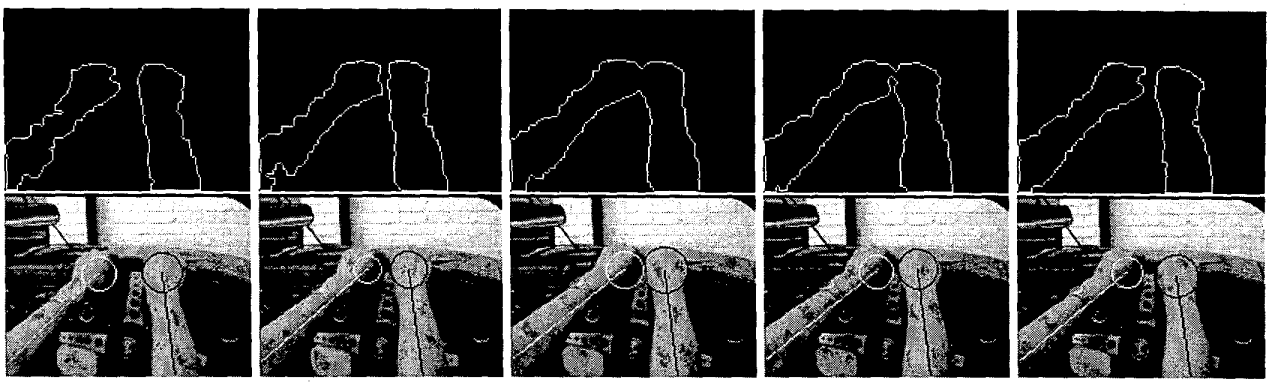

Figure 2: Hands touching

this early initialisation stage there may not be enough information to get an accurate value of the orientation of the forearm. It is highly probable that only a small part of the hand is showing. This does not matter since the line will begin to move towards the true orientation once more information becomes available.

4.4 Tracking

The model instances must be carefully managed so that they are added and deleted at the correct times and the models are associated consistently with the same hand and forearm over time. Whenever a new component appears, a new model is initialised using the procedure described in the previous subsection. In the subsequent frames, if the model centre is contained within a region of interest then it is associated with that region, otherwise it is associated with the closest region. The addition and deletion of components is not straightforward, with one case in particular proving problematic. This particular scenario is where there are two components at time t and one component at time $t+1$. There are two possible causes for this: either one hand/arm has left the image or the hands have moved together producing just one connected component. One solution is to use a heuristic which states that if, at time t, the arm was at the edge of the image and moving out of the scene then if there is only one component at time $t+1$ then it is because there is only one hand now in the scene. Separate Kalman filters are used to predict the displacement of the model parameters x, y, and θ in each frame so it is reasonable to claim that a hand may be moving out of the scene if the displacement predicted by the Kalman filters would take it out of the scene. The filters are used for two purposes: to smooth the movement of the model and to define a search space for the model fitting process. The filters use a constant velocity dynamic model.

4.5 Model Fitting

The fitting process for a model begins in the frame subsequent to the model's initialisation and aims to find the model parameters which maximise both the radius of the circle and the fit of the line to the orientation of the forearm. Recall that the distance transform is masked so that pixels outside the regions of interest are assigned distance values of zero. Let $DT(x,y)$ denote the masked distance transform value at pixel (x,y). The model is fitted to the distance transform response by maximising the objective function:

$$f(x_c, y_c, \theta) = \alpha DT(x_c, y_c) + \sum_{(x,y)\in L} DT(x,y) \quad (1)$$

where L is the set of pixels on the line segment joining (x_c, y_c) to the side of the image at orientation θ. The second term rewards a long line segment that lies along the middle of the forearm. The first term rewards a circle with a large radius. The weight α is set based on the maximum value of the distance transform in the current region of interest. This has the effect of penalising moving the endpoint (x_c, y_c) away from the palm centre up into the finger region. Masking the distance transform ensures that pixels likely to correspond to scene background do not contribute to the objective function. Fitting is performed within a 3D search space, the extent of which is calculated using the Kalman filter predictions. The use of such a search space assumes that there will not be sudden, extreme movement between frames.

5 Results and Discussion

The performance of the model is demonstrated using sequences recorded inside a vehicle parked outdoors. Images from the first sequence are given in Figure 1. Starting from an empty scene, the driver brings his hands into the scene and grasps the steering wheel. In the first few frames the segmentation is quite noisy since the foreground colour model is sparsely populated. Once more information has been added to the colour model a cleaner segmentation is achieved. The first frame shows the first attempts to fit the hand model to the regions of interest. These appear wrong when superimposed over the original image because of the errors in the initial segmentation. In the fourth pair of images the two hand models are beginning to move towards their correct positions. At this frame, the left model's line orientation describes a line segment which passes over the background. Equation (1) rewards the line for moving into the position where the line segment lies down the middle of the arm, although the use of the Kalman filter means that θ takes a couple of frames to adjust to the optimal value from the intial estimate. The last image is captured five frames later and shows the models settled in positions which resonably approximate the position, scale and orientation of the hand.

Figure 2 shows the scenario where two hands touch on the steering wheel. This sequence illustrates the problem described in Section 4 regarding the association of models to regions of interest. In the second image there are two models and two regions but in the subsequent frame (next image) there is only one region. The heuristic whereby the position and previous displacement of the models are examined is used to decide if a model has truly disappeared. In this case it is highly improbable that a hand has moved out of the scene so two models are fitted to the one region. The models stay in the correct hands and arms because they would penalised if they attempted to move into the wrong hand. This would incur a drop in the value of the objective function since the arm line of each model would be moving away from the 'true' orientation of the arm. When the hands move apart again, there are once more two separate regions with one model belonging to each.

Figure 3 is a sequence of the driver crossing his hands on the steering wheel. The figure shows the system successfully tracking the hands through the action, i.e. in each frame it places each model over the hand it was associated with before the two hands moved together. The top-right image illustrates a temporary uncertainty in the location of the hands as the right hand begins to occlude the left. The line segment constraint prevents the model from mistakenly 'locking on' to the right hand and following it because the model is penalised if the line segment moves away from the orientation where it lies down the arm. The situation is resolved when the left hand begins to show from beneath the right arm. Since the model has not mistakenly followed the right hand and is still aligned down the left arm it is able to reacquire the left hand when it reappears.

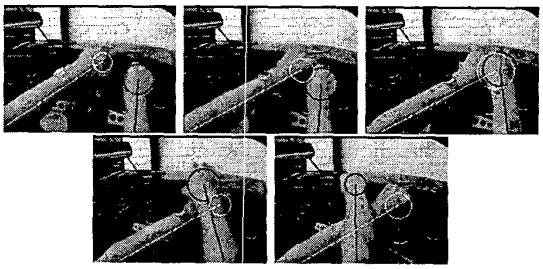

Figure 3: Hands crossing

Figure 4 is a sequence of the hands uncrossing after a a crossing motion as illustrated in Figure 3. As the hands pull away, the left hand model is temporarily erroneously positioned with the circle being placed in the right hand (bottom-left image). This situation is again resolved by the line segment constraint. As the hands pull farther apart, the hand model is encouraged to return to the left hand because remaining in the right hand results in part of the line segment lying across the background causing a drop in the value of the objective function.

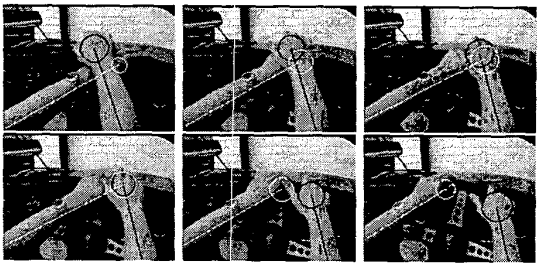

Figure 4: Hands uncrossing

If a hand is connected to an arm then the line segment connecting the hand through the forearm cannot cross the background except in the extreme and unlikely case where the wrist joint is pitched at plus or minus 90 degrees. From these sample sequences it

is apparent that this line segment constraint plays an important role in resolving the ambiguities that arise when hands touch and cross. The image sequences were captured using a Matrox Meteor-II framegrabber and processed using a Pentium-III 500MHz PC with 256MB memory.

6 Future Work

The distance transform is the slowest operation in the process and it may become necessary to speed it up even further. To achieve this we are investigating the possibility of creating a version which takes advantage of the multiple instruction capability of the Pentium-III processor such as the SIMD extensions. Some work has been published on SIMD distance transforms [10] but not for the platform we are using, although the algorithms used would be similar. The next stage of this research is to identify a pointing finger and its target direction to allow control of the proposed vehicle interface.

Acknowledgements

This research is funded by EPSRC grant number GR/M26633 as part of the ACTIVE Foresight Vehicle Link project in collaboration with Daewoo Motor Company, Vision Dynamics Ltd. and the Institute of Behavioural Sciences at the University of Derby.

References

[1] G. Borgefors. Distance transformations in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 321–345, 1984.

[2] N. Cairnie, S.J. McKenna, I.W. Ricketts, and G. McAllister. A prototype adaptive finger-pointing interface for operating secondary controls in motor vehicles. In *IEEE Conference on Systems, Man and Cybernetics*, Nashville, Tennessee, October 2000.

[3] S. Gautama, S. Lacroix, and M. Devy. Evaluation of stereo matching algorithms for occupant detection. In *IEEE International Conference on Computer Vision*, pages 439–448, Corfu, Greece, September 1999.

[4] D.M. Gavrila and V. Philomin. Real-time object detection using distance transforms. In *IEEE International Conference on Intelligent Vehicles*, Stuttgart, Germany, 1998.

[5] G. McAllister, S.J. McKenna, and I.W. Ricketts. Towards a non-contact driver-vehicle interface. In *IEEE Conference on Intelligent Transportation Systems*, Dearborn, Michigan, October 2000.

[6] T. Nakano, K. Sugiyama, M. Mizuno, and S. Yamamoto. Blink measurement by image processing and application to warning of driver's drowsiness in automobiles. In *IEEE International Conference on Intelligent Vehicles*, pages 285–290, 1998.

[7] S.-B. Park, A. Teuner, B.J. Hosticka, and G. Triftshauser. An interior compartment protection system based on motion detection using CMOS imagers. In *IEEE International Conference on Intelligent Vehicles*, pages 297–301, Stuttgart, Germany, 1998.

[8] V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 677–695, July 1997.

[9] Y. Sato, Y. Kobayashi, and H. Koike. Fast tracking of hands and fingertips in infrared images for augmented desk interface. In *IEEE International Conference on Face & Gesture Recognition*, pages 462–467, Grenoble, France, March 2000.

[10] J.H. Takala and J.O. Viitanen. Distance transform algorithm for bit-serial simd architectures. *Computer Vision and Image Understanding*, pages 150–161, May 1999.

[11] D. Tock and I. Craw. Tracking and measuring driver's eyes. *Image and Vision Computing*, 14:541–548, 1996.

[12] V. Vetter, T. Zielke, and W. von Seelen. Integrating face recognition into security systems. In *Audio- and Video-based Biometric Person Authentication*, pages 439–448, Crans-Montana, Switzerland, March 1997.