Real-Time Visual Recognition of Facial Gestures for Human-Computer Interaction

Alexander Zelinsky and Jochen Heinzmann
Department of Systems Engineering
Research School of Information Sciences and Engineering
Australian National University
Canberra, ACT 0200,
Australia
E-mail: alex@syseng.anu.edu.au

Abstract

People naturally express themselves through facial gestures and expressions. Our goal is to build a facial gesture human-computer interface for use in robot applications. We have implemented an interface that tracks a person's facial features in real time (30Hz). Our system does not require special illumination nor facial makeup. By using multiple Kalman filters we accurately predict and robustly track facial features. This is despite disturbances and rapid movements of the head (including both translational and rotational motion). Since we reliably track the face in real-time we are also able to recognise motion gestures of the face. Our system can recognise a large set of gestures (13) ranging from "yes", "no" and "may be" to detecting winks, blinks and sleeping.

1. Introduction

Gestures are an important form of communication between people. We regard expressions of the face as one of the most natural forms of human expression and communication. For people who are elderly, disabled or just inexperienced users of computer technology a gesture interface would open the door to many applications ranging from the control of machines to "helping hands". The crucial aspect of a gesture interface is not only real-time performance, but also the ablity to operate robustly in difficult real world environments.

Work on real-time face tracking has been recently reported [1-4]. Controlling a clone at 10Hz in virtual reality is described by Saulier *et al*. [1]. Specialized filters are used to extract the location of the facial features and the facial expression. This system requires cosmetic makeup to enhance the facial contrast. This is an undesirable restriction.

The detection of human body poses with a stereo camera system is described by Gavrila *et al.* [2]. Another face tracker reported by Jacquin *et al.* [3] works robustly without training for different people using only a general model.

To understand human gestures based on head movement a system must be capable of tracking facial features in real time. We consider real time to be NTSC video frame rate (30Hz). If facial tracking is done at lower rates then it is very difficult to understand gestures. We believe approaches [1-3] do not work sufficiently fast enough for motion based gesture recognition.

The research Media lab at MIT has developed a system for hand and facial gesture recognition with two cameras [4]. A static wide field camera is used to identify the area of interest and an active narrow field camera for obtaining high resolution pictures of the hand or the face. Our goal is to use a single passive camera in a natural scene to track a human face.

We use dedicated hardware which tracks features in real-time using template matching (See Section 2). Relying solely on such dedicated hardware it is not possible to reliably and robustly track a human face since under normal lighting conditions the shape and shading of facial features will change markedly when the head moves. This results in a failure by the vision hardware to correctly match the changing templates.

We solve this problem by using Kalman filters to fuse data from the tracking system with a geometrical model of the face. We have built a face tracker that operates under natural lighting without artifical artefacts. The system is robust and runs at video frame rate. (See Section 3).

Reliable and rapid tracking of the face gives rise to ability to recognise gestures of the head. We define a gesture to consist of a chain of atomic actions, where each atomic action represents a basic head motion e.g. upwards or to the right etc. The "yes" gesture is represented the atomic action chain of "move up", "stop", "move down", etc. Our system checks for all possible atomic actions in each video frame. An "observer" is instantiated for every atomic action that is detected. This approach allows us to detect several gestures in parallel. For example people can be winking or opening their mouths while they are nodding. If an observer reaches the end of a chain of atomic actions then a gesture is deemed to have been recognised. We use a probabilistic approach to decide if an atomic action has been triggered. This is necessary since it is rare for identical actions to be exactly the same e.g. nobody nods in the same way everytime. (See Section 4).

2. The Vision System

We use the MEP tracking vision system to implement our facial gesture interface. This vision system is manufactured by Fujitsu and is designed to track in real time multiple templates in the frames of a NTSC video stream. It consists of two VME-bus cards, a video module and a tracking module which can track up to 100 templates simulataneously at video frame rate (30Hz for NTSC). A MC68040 processor card running VxWorks executes the application program and controls the vision system.

The tracking of objects is based on template (8x8 or 16x16 pixels) comparison in a specified search area. The video module digitises the video input stream and stores the digital images into dedicated video RAM. This RAM is also accessed by the tracking module. The tracking module compares the digitised frame with the tracking templates within the bounds of the search windows. This comparison is done by using a cross correlation which sums the absolute difference between corresponding pixels of the template and the frame. The result of this calculation is called the distortion and measures the similarity of the two comparison images. Low distortions indicate a good match while high distortions result when the two images are quite different.

To track a template of an object it is necessary to calculate the distortion not only at one point in the image but at a number of points within the search window. To track the movement of an object the tracking module finds the position in the image frame where the template matches with the lowest distortion. The motion is represented by a vector to the origin of the lowest distortion. By moving the search window along the axis of the motion vector objects can be easily tracked. The tracking module performs up to 256 cross correlations per template within a search window,.

The MEP tracking vision system works perfectly for objects that do not change their appearance, shade and are never occluded by other objects. In another project we have successfully used the MEP tracking system to successfullly implement a vision based mobile robot navigation system [5].

Problems arise when the vision system is used to track a face in a head and shoulders image of a person. Since the head occupies most of the image, one template of the entire face exceeds the maximum template size allowable in the vision system. Therefore, it is only possible to track individual features of the face such as the eyes or mouth. Through experimentation we have found that facial features with high contrast are good candidates as tracking templates. For example an eyebrow which appears to be a dark stripe on a light background (if the person has light skin) and the iris of the eye which appears as dark spot surrounded by the white of the eye are well suited for tracking. However, some facial features are not as easy to track. For example the corners of the mouth are difficult to track. This is because these features are made up primarily of plain skin (80%) and the correlation with a facial template of only plain skin is not significantly different i.e. yields a low distortion.

These problems are further complicated by the fact that well suited tracking features can change their appearance dramatically when a person moves their head. The shading of the features can change due to uneven illumination and the features appear to deform when the head is turned, moved up, down or tilted to the side. All these changes increase the distortion even if a template is matching precisely at the correct position. It also results in low distortions at the wrong coordinates which then cause the search window to be incorrectly moved away from the feature. This problem arises when a head is turned sufficently far enough for one half of the face with all its associated features to completely disappear. Once the tracking feature has left the search window the movement vectors calculated by the vision system are unpredictable. We have developed a method to allow a search window to correctly find its lost feature thus yielding a reliable face tracker.

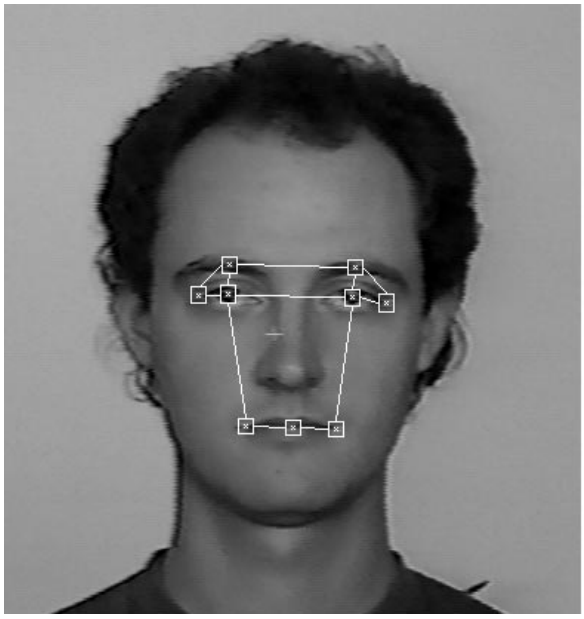

Figure 1: Facial Tracking Features

3. Tracking the Face

Our basic idea is that individual search windows help each other to track their features. From the known geometric relationship between the features in a face, a lost search window can be repositioned with help from features that are still tracking. We use a two dimensional model of the face in which features for tracking are joined to form a small network. The reference vectors connecting the features are derived from a single image automatically by the system or by a human operator. Figure 1 shows a face with boxes marking the nine (9) tracking features. We use the iris', the corners of the eyes, the eyebrows and the middle and corners of the mouth. The sizes of the boxes shown are the actual template sizes (16x16 pixels). The line connections shown in the figure indicate which

features assist the other features for readjusting the search windows. We also use several templates to track features that can change their appearance. For example the eyes can be open or closed. In such cases we use three (3) templates for the different states (opened, closed and half-open-closed) of the eyes simultaneously. This makes it possible to determine the state of the tracking features e.g. an eye is open or the mouth is closed.

As discussed earlier if a person turns their head the distortions of all the templates increases greatly. In this situation some features may disappear and others may change their shade and appearance. It is difficult to determine whether search windows are tracking correctly or incorrectly. Lost search windows influence the tracking position of the other search windows. A situation can easily arise in which the system will loose the entire face. Simple thresholding of the distortion is insufficient to distinguish the lost windows from the tracking ones. An approach that can cope with noisy data is needed. Kalman filters were used solve this problem.

The Kalman filter is a recursive linear estimator which merges the measurement of sensors observing the environment with a prediction that is derived from a system model [6-7]. The Kalman filter is used in many applications such as navigation of planes, missiles and mobile robots where uncertain measurements from sensors that observe landmarks are used to localise a vehicle. By merging sensor information, the Kalman filter guarantees an optimal estimate of the sensor data in terms of a minimum mean-square error if an appropriate system model is used. All sensor data has covariances associated with it which indicate the reliablity of the data. The output of the filter also has a covariance, so the control system does not only obtain an estimate, but it also knows the reliability of the estimate.

The use of Kalman filtering to assist in tracking for has been previously reported McLauchlan et.al. [8]. We apply also separate Kalman filters to each tracking feature. However, in our approach also features assist each other in the tracking of features. Hager et.al. [9] proposed a similar idea of using geometric constraints and feature states for robust tracking. The Hager et.al. technique uses a binary switching between winning features and loosing features to decide who gives direction for other features to track. Our approach uses all the features for tracking and is weighted to the features that are tracking best.

We use the motion vectors calculated by the vision system is used as the external sensor measurements and the covariances are calculated from the distortion of the template correlations. We use the relative positions of the other search windows and the geometric model of the face to derive the prediction of the internal system model for input into the Kalman filter as the internal sensors.

A facial motion vector is calculated by averaging all nine (9) feature motion vectors. This calculation is weighted by the variances of the tracking features. This ensures that the facial motion vector is biased towards the features that are tracking well. The facial motion vector is used in the calculation to predict the position of the tracking features.

The predicted position of a feature is determined from the position of other features in the previous video frame together with the 2D face model and the facial motion vector. The calculation is weighted by the variance from the previous video frame. Figure 1 shows the features that use each other as references are connected by a white line. For example the left eye corner only uses the left eye and the left eyebrow to predict its position.

Using Kalman filtering yields a system which copes with head rotations of about 30 degrees during facing tracking. Further robustness was added to the face tracking by implementing dynamic search regions which look for a feature inside a specifc area of the image. The size of the search region is dependent on the variance of the features (determined from the Kalman filter). We also extended our 2D model of the face to allow for tilting. These extra techniques allow the head to be rotated up to 60 degrees, tilted acutely from side to side, and enables quick recovery even when all the tracking features have been lost.

Figure 2 shows four (4) images taken from a tracking sequence. The predicted estimates of the tracking features are marked with small white crosses.

Figure 2: Tracking the Face

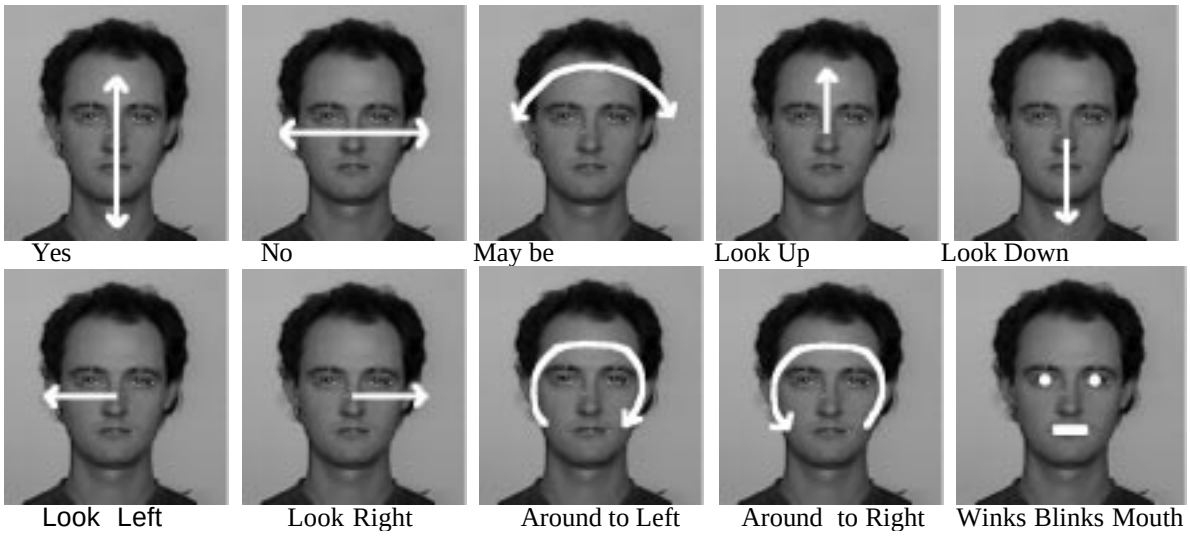

Figure 3: Recognisable Gestures

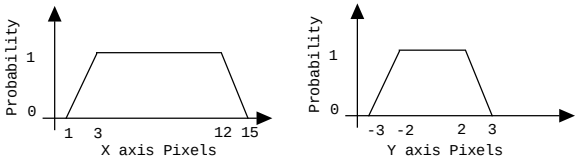

Figure 4: Movement Activation in the X and Y axes

Determining if an atomic action has been triggered is not easy since gestures are rarely identical e.g. people do not nod in the same way every time. To solve this problem we video taped and analysed people performing various gestures. Initially, we planned to use a Gaussian distribution to describe an atomic action. We found trapezoidal functions to be most effective. Figure 4 shows the movement activation functions of the atomic action MoveRight.

The activation level A_j of an atomic action AA_j is defined in a feed forward recursive manner:

$$A_i = \frac{(A_{i-1}+1)}{Frames} P(x)P(y)$$

where $P(x)$ and $P(y)$ represent the movement activation levels of AA_i in the X and Y axes and $Frames$ represents the expected number of frames to complete an atomic action. In the case of the MoveRight we expect the action to complete within 4 frames and that the head will generally move between 3 and 12 pixels in the X axis and minimally in the Y axis in a single video frame. The output activation level OA_i is tuned to gradual ly increase when a action begins using:

$$OA_i = A_i^2$$

which results in the OA_i having the graph profile shown in Figure 5.

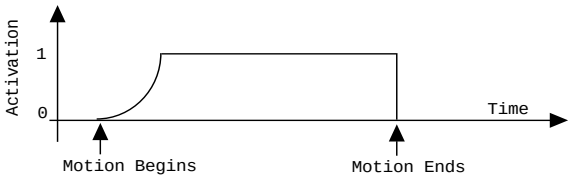

Figure 5: Atomic Action Activation

An observer is instantiated for every atomic action AA_i that is the first action of a gesture G_j. O_k is instantiated only after OA_i becomes zero. The observer then checks if the next detected atomic action AA_{i+1} matches the atomic action expected by the observer O_k. If a match occurs then the observer advances to the next action in the sequence. If the observer reaches the end of an atomic action sequence then gesture G_j is deemed to have been recognised. This approach allows us to detect several gestures in parallel. For example people can be winking or opening their mouths while they are nodding.

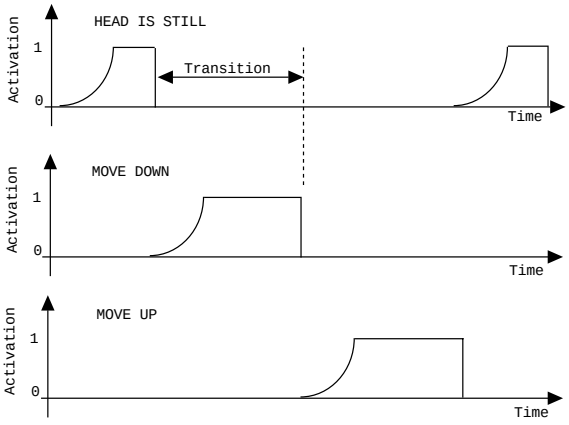

Figure 6: Gesture Transition States

Figure 6 shows the output activations for a nodding gesture. Initially the head is still, then it is moved down, then up and finally it comes to rest again. A key point to recognising such gestures correctly involves detecting the transition between atomic actions in the sequence. The transitions between states can sometimes be noisy. They are handled in the following manner. If OA_i for AA_i becomes zero a look up is done to find the next atomic action AA_{i+1} in the gesture, the look up also returns a transition timing function (measured in frames) that is used to judge the validity of the transition. Figure 7 shows the timing function for the MoveUp atomic action.

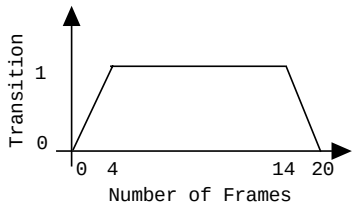

Figure 7: Transition Timing Function

The transition function TAA_m for atomic action AA_i is defined as:

$$TAA_i = T(AA_{i+1})OA_{i+i}$$

where $T(AA_{i+1})$ is the transition frame timing of next atomic action AA_{i+1} in the gesture and OA_{i+1} is the ouput activation level of AA_{i+1}. In our system if $TAA_{i+1} > 0.3$ then the transition to the next action state occurs otherwise the observer is deleted.

We also keep a running sum score for each active observer O_k. `Score` indicates what proportion of a gesture has been seen.

$$Score = \sum TAA_i$$

We also sum the scores for all the observers that are monitoring the same gesture G_i. $Score_g$ is necessary when detecting oscillating head gestures because the first atomic action of a gesture can be seen several times. For example the trigger to create an observer for the nodding gesture of yes can be detected several times.

$$Score_g = \sum Score$$

A `TotalScore` is also kept which measures how many gestures are currently being processed.

$$TotalScore = \sum Score_g$$

The `TotalScore` is used as a parameter to the sigmoid function `Unknown` shown in Figure 8. The purpose of this function is to give a heuristic measure of how much uncertainty there is in the current gestures being processed. The function returns higher values as the gesture being processed gets closer to the endof a sequence. We have high confidence about the recognition of a gesture if we have detected the 6th atomic action in a 7 action sequence. This confidence is higher than the detection of the second atomic action in another gesture.

$$Unknown(x) = \frac{e^{3-x}}{1+e^{3-x}}$$

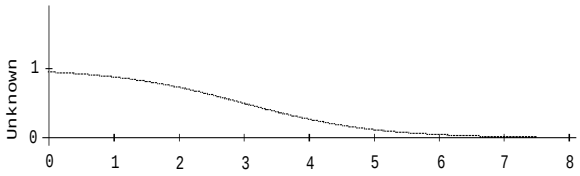

Figure 8: Unknown Gestures

We then compute the probabilities for all the gestures currently in the system.

$$Known(TotalScore) = 1 - Unknown(TotalScore)$$

$$P(G) = \frac{Score_g}{TotalScore} Known(TotalScore)$$

where P(G) is the confidence that the gesture G has been recognised. We output this confidence when the observer reaches the last atomic action in the sequence. A decision can be made to accept or reject the recognition of the gesture.

We have implemented a gesture recognition module which runs in parallel with the face tracking module at video frame rate (30Hz). The approach we have adopted produces reliable results and is robust to noise. The system accurately discriminates between 13 different gestures. Even though some gestures are quite similar to each other.

5 Conclusions and Further Work

The real-time facial gesture recognition system we have developed consists of two modules running in parallel; a Face Tracker and a Gesture Recogniser. The face tracking module fuses information from the vision system with information derived from a two-dimensional model of the face using multiple Kalman filters.

Our system is able to track the features without special illumination or makeup. It can track features change shade, deform or even disappear. We have had experimental success in all situations where a person turns their head 60 degrees (it is physically difficult to turn further!). Further rotation increases the risk of loosing all features since the initial templates are all taken from images with the person looking straight into the camera. Our system does recover from such situations by using dynamic search. However, the recovery can take several seconds. In future work we plan to introduce a 3D-model of the face which will allow us to more precisely predict the position of the face.

Another improvement we are considering is to grab templates of the features dynamically while the system is tracking the face. This would not only improve the tracking., but the system would also cope with much greater ranges of changing illumination.We are planning

to create a dynamic face model that adapts to the gathered data. Such a dynamic system would learn how to track the face of a unknown person. The system would be initially provided with several generic faces including startup templates and face geometries. It selects the most similar model for the unknown person and then learns the exact templates and geometry.

Our Gesture Recogniser module which runs in parallel with the face tracking module is capable of recognising a wide variety of gestures based on head movements. Gesture recognition is robust due to the statistical approach we have adopted. If future we plan to record and analyse the head gestures of a large sample of people. The statistical parameters of head motion will be incorporated into our program. We also plan to explore the prospect of allowing the machines to learn gestures based on observation.

Our ultimate aim is to use our facial gesture recognition system in a robotic system for the disabled. Our interface will allow disabled persons to feed themselves by using facial gestures to communicate with the helping robot.

Acknowledgements

This work has been funded by the Australian Research Council and by the generous support of Wind River Systems, supplier of VxWorks.

References

[1] A. Saulier, M.-L. Viaud and D. Geldreich, "Real-Time Facial Analysis and Synthesis Chain", Proceedings of the International Workshop on Automatic Face and Gesture Recognition pp. 86-91, 1995.

[2] D.M. Gavrila and L.S. Davis,"Towards 3-D model-based tracking and recognition of human movement: a multi-view approach", Proceedings of the International Workshop on Automatic Face and Gesture Recognition, pp. 272-277, 1995.

[3] A. Jacquin and A. Eleftheriadis, "Automatic location tracking of faces and facial features in video sequences", Proceedings of the International Workshop on Automatic Face and Gesture Recognition, pp. 142-147, 1995.

[4] T. Darrel and A.P. Pentland, "Attention-driven Expression and Gesture Analysis in an Interactive Environment", Proceedings of the International Workshop on Automatic Face and Gesture Recognition, pp. 135-140, 1995.

[5] G. Cheng and A. Zelinsky, "Real-Time Visual Behaviors for Navigating a Mobile Robot", International Conference on Intelligent Robots and Systems (IROS), 1996.

[6] H. Durrant-Whyte,"Autonomous GuidedVehicle Technology", Proceedings of the Joint Australian-Korean Workshop on Manufacturing Technology,Australian Academy of Science, ISBN 1-875618-20-1,pp. 51-60, 1995.

[7] S.M. Bozic, *Digital and Kalman Filtering*, Edward Arnold Publishers, 1986.

[8] P.F. McLauchlan, I.D. Reid, D.W. Murray, Recursive Affine Structure and Motion from Image Sequences, Lecture Notes in Computer Science Vol.800, Proceedings of ECCV'94, pp 217-224, ISBN 3-540-57956-7, Springer Verlag Berlin Heidelberg 1994.

[9] G.D. Hager, K. Toyama, X Vision: A Portable Substrate for Real-Time Vision Applications, Description of the X Vision class library and applications, Technical Report, Yale University, 1995.