Multi-Modal Tracking of Faces for Video Communications

James L. Crowley and Francois Berard
GRAVIR - IMAG, I.N.P. Grenoble
46 Ave Félix Viallet
38031 Grenoble, France

Abstract

This paper describes a system which uses multiple visual processes to detect and track faces for video compression and transmission. The system is based on an architecture in which a supervisor selects and activates visual processes in cyclic manner. Control of visual processes is made possible by a confidence factor which accompanies each observation. Fusion of results into a unified estimation for tracking is made possible by estimating a covariance matrix with each observation.

Visual processes for face tracking are described using blink detection, normalised color histogram matching, and cross correlation (SSD and NCC). Ensembles of visual processes are organised into processing states so as to provide robust tracking. Transition between states is determined by events detected by processes. The result of face detection is fed into recursive estimator (Kalman filter). The output from the estimator drives a PD controller for a pan/tilt/zoom camera. The resulting system provides robust and precise tracking which operates continuously at approximately 20 images per second on a 150 megahertz computer work-station.

1. Introduction

The images transmitted for video-communications are highly repetitive. In such images, a human face (and torso) undergoes a limited set of deformations, as the subject speaks and gestures with his body movements. These deformations can be captured in an orthogonal "basis space" of images. Such a space permits each individual image to be coded and transmitted as a relatively small vector of coefficients. As few as 15 such coefficients (coded as 60 bytes) can be sufficient for quite realistic reconstruction of a talking face, provided that the face is registered and normalised in position and size.

Locating and normalising a face is a processes of tracking. A variety of methods for detecting and registering the position and scale of a face can be demonstrated in laboratory environments. However, each of these methods can fail in naturally occuring circumstances. A reliable tracking system for registration can be obtained by integrating and coordinating several complementary tracking processes. Integration and coordination are performed using a synchronous architecture in which a supervisor activates and controls visual processes in cyclic manner.

This system demonstrates that robust operation can be achieved by coordinating multiple visual processes. Control of individual processes is made possible by the inclusion of a confidence factor accompanying each observation. Fusion of the results is made possible by the determination of an error estimate (a covariance matrix) for each observation. Composing a system from a redundant ensemble of processes permits the overall system to automatically adapt to a variety of operational circumstances.

The following section reviews the visual process architecture used for the system and describes techniques for estimating a confidence factor and an error bounds for visual processes. A tracking process based on a zero-th order recursive estimator is described. Visual processes are described for the detection and tracking of faces using blink detection, color, and correlation, as well as processes for estimation and camera control. Visual processes are grouped into states with state transitions triggered by events. An example of an execution trace is provided with the system as configured at the time of writing.

2. A Synchronous Ensemble of Visual Processes

The face tracking system described in this paper is based on an architecture in which a supervisor activates and coordinates a number of reactive visual processes. We call such an architecture a synchronous ensemble of reactive visual processes (SERVP). This architecture has been developed in the context of robotics [3] and surveillance tracking [4].

2.1 The SERVP Architecture

The SERVP architecture is designed for controlling soft real-time processes embedded within single processor. Within the SERVP architecture, processes are executed in a synchronous manner with an explicit limit placed on the computing time which each process may use in each cycle. When executed in a standard Unix environment, such a system provides only soft real-time response. Hard real time response can be obtained when such an architecture is used with a real time kernel. In either case, the supervisor must manage the time used by each phase so as to assure a fixed cycle time.

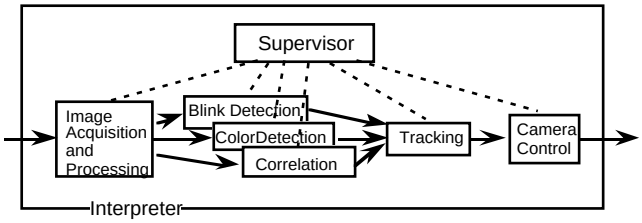

Figure 1. A Supervisory controller selects and controls the sequencing of perceptual processes. Multiple processes can be active at the same time.

The SERVP model is illustrated in figure 1. The core of the supervisor is an interpreter in which procedures for image acquisition, visual processes, and device controllers have been linked. The supervisor acts as a scheduler which drives the system as a sequence of phases. The system supervisor is expressed as a set of objects which represent the current state of visual processes, and a set of rules which react to events and commands. Versions of the supervisor have been built in both CLIPS (C Language Integrated Production System) and TCL/TK. The experiments described below are based on a TCL/TK implementation.

In scheduling and executing visual processes, the supervisor must manage the time spent in each of the processes. The supervisor receives messages from the visual processes concerning commands and visual events. In reaction to these messages, visual processes are activated or dis-activated. At the beginning of each cycle, the supervisor assigns priorities to individual processes and then translates these priorities to allocations of time slices. The supervisor then executes the image acquisition procedure, as well as initial image processing, such as resolution reduction. The supervisor then activates individual visual process, managing the time budget as the processes are executed. Processes generate symbolic messages to the supervisor, which can change the state of subsequent processes. The visual events used in this system are based on the confidence factor which accompanies the results of each process.

In our example, visual processes pass information to a tracking process which maintains an estimate of the center point and size of the face. This tracking process is a form of recursive estimator (Kalman filter), commonly used for sensor fusion [2]. This recursive estimator provides a reference signal to a PD controller for a RS232 controlled pan/tilt/zoom camera. Multiple copies of a visual processes may be simultaneously active. In such a case, each process possess a separate data component which contains its parameters and state. When several copies of the same process are active, the supervisor must assure that they are not performing the same task on the same data.

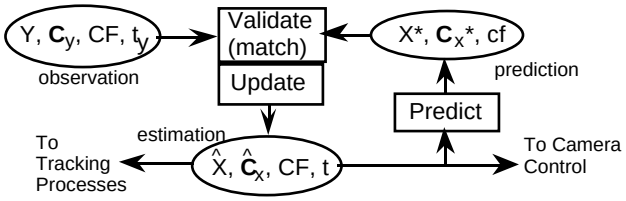

Figure 2. The Tracking Process is a zero-th order recursive estimator for position and size.

2.2 Fusion and Integration in a Recursive Estimator

The use of estimation theory for tracking and for fusion of information in computer vision and robotics is well established [1], [2]. For our face tracker we use a zero-th order recursive estimator to maintain estimates of independent state vectors for the center position of the face X_p = (i, j) and the vertical and horizontal size of the face, X_s = (h, v). We chose to estimated the horizontal and vertical size of the face as two parameters because the aspect ratio of the face can change with rotations. Thus our state vector has four parameter, (i, j, h, v), measured in pixels.

In our application, size and position are independent. Thus we can replace the inversion of a 4 x 4 matrix with two inversions of a 2 x 2 matrix in the fusion stage by separating the state vector into a position component, X_p, and a size estimate, X_s. Both vectors are measured in pixels. Each vector is accompanied by a covariance

$$\hat{X} \equiv \begin{bmatrix} \hat{X}_p \\ \hline \hat{X}_s \end{bmatrix} \equiv \begin{bmatrix} \hat{i} \\ \hat{j} \\ \hat{h} \\ \hat{v} \end{bmatrix}$$

Fusion of perceptual information is made possible by an explicit estimate of the precision and confidence of each observation [1]. The covariance matrix, $\mathbf{C}_x$, is an estimation of the error of the estimated state vector.

$$\hat{\mathbf{C}}_X \equiv \left[\begin{array}{c|c} \mathbf{C}_{ij} & 0 \\ \hline 0 & \mathbf{C}_{hv} \end{array}\right] \equiv \begin{bmatrix} \sigma_i^2 & \sigma_{ij} & 0 & 0 \\ \sigma_{ji} & \sigma_j^2 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & \sigma_{hv} \\ 0 & 0 & \sigma_{vh} & \sigma_v^2 \end{bmatrix}$$

Every visual process in our system provides an observation of the state vector (or a subset) accompanied by a time stamp, t, a covariance matrix, $\mathbf{C}_y$, and a confidence factor, CF. The update of the estimation by an observation uses the covariances to determine relative weights for the observation and the prediction.

Our demonstration system is built under UNIX and is thus soft real-time; we can not guarantee the time step. Thus each observation is accompanied by a time stamp, t. The step, Δt, is the different between the time the estimation was last updated, and the time the observation was made.

$$\Delta t = t_{obs} - t_{est}$$

Movements of the subject between observations are unpredictable, and can be in any direction. Thus we make no attempt to estimate derivatives. As a result, the prediction of the state vector, X^* at time t_{obs}, is simply the last updated estimate at time t_{est}.

$$X^* := \hat{X}$$

The covariance, on the other hand, does depend on the time step. The uncertainty in position of the subject is a quadratic estimate which grows as the square of the time step. This growth is captured in a 4x4 matrix **W**, whose terms give the loss in precision of each component as a function of seconds-squared. Thus the uncertainty is updated as:

$$\mathbf{C}^*_x := \hat{\mathbf{C}}_X + \Delta t^2 \mathbf{W}$$

The loss in precision with time is calibrated by observing a sequence of position and size estimates of a normal user, at a regular (unit) sampling interval, Δt_{min}. The coefficients of **W** are given the products of the expected values of the change in parameters in adjacent frames.

Statistical estimates combine as the momentum of masses. Or more precisely, mass is a statistical estimate. In any case, each visual process in our system produces an estimate of the uncertainty of each observation represented by a covariance matrix, $\mathbf{C}_y$. In most of our visual processes the covariance matrix can be directly estimated from the results of processing, as will be shown below. The correction of the estimation by an observation uses the covariance to give a relative weight to the observation and the prediction. The new estimated covariance is given by:

$$\hat{\mathbf{C}}_X := (\mathbf{C}^*_x{}^{-1} + \mathbf{C}_y{}^{-1})^{-1}$$

The new estimate can then be computed as a weighted average:

$$\hat{X} := \hat{\mathbf{C}}_X (\mathbf{C}^*_x{}^{-1} X^* + \mathbf{C}_y{}^{-1} Y)$$

2.3 Estimating the Confidence of Visual Processes

The primary visual event used in the current demonstration system is the confidence factor, CF, which each process attributes to its result. Confidence is represented by a numerical value between 0 (no confidence) and 1 (certainty). The CF factor estimates the likelihood that a successful detection was achieved. Confidence is generally computed as a probability using a pre-trained sample set of correct detections. During system set up, a large number of correct detections are hand selected and catalogued. The mean, μ_S, and covariance, $\mathbf{C}_S$, for these sample detections are computed.

During ordinary operation, the probability of a correct observation, given the observed vector, Y, is computed using this pre-calibrated mean and covariance are parameters for a un-normalized Gaussian density function. This probability defines the confidence factor used in controlling and coordinating processes.

$$CF_y = e^{-\frac{1}{2}(Y-\mu_S)^T \mathbf{C}_S^{-1} (Y-\mu_S)}$$

3. Visual Processes for Detection and Tracking of Faces

Robust continuously operating tracking can be obtained by driving the tracking process with several complementary detection processes. The tracking process then provides a reference with which individual processes can be re-initialised when their result becomes unreliable. Such synergistic integration greatly improves both the reliability and the precision of the tracking process. This section describes processes for detecting faces using blinking, normalised color and cross-correlation.

3.1 Detecting Faces from Blinking

A human must periodically blink to keep his eyes moist. Blinking is involuntary and fast. Most people do not notice when they blink. The fact that both eyes blink together provides a redundancy which permits blinking to be discriminated from other motions in the scene. We have found that detecting the motion pattern of blink is an easy and reliable means to detect the presence of a face. The fact that the eyes are symmetrically positioned with a fixed separation provides a means to normalize the size and orientation of the head from the detection.

Blink detection is based on the difference of successive images. The difference image generally contains a small boundary region around the outside of the head. If the eyes happened to be closed in one of the two images, there are also two small roundish regions over the eyes where the difference is significant, as shown in figure 3.

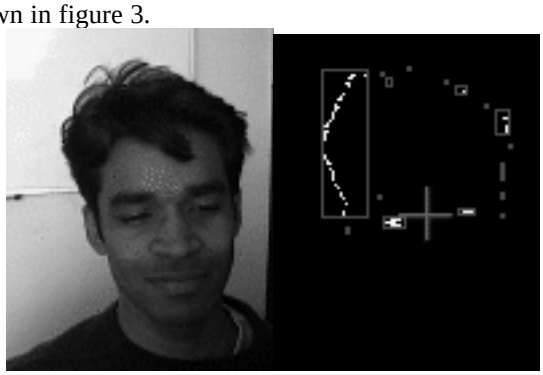

Figure 3. A face image and the thresholded difference with bounding boxes and face position.

The difference image is thresholded, and a connected components algorithm is run on the thresholded image. A bounding box is computed for each connected component. Candidate regions for an eye are selected based on horizontal and vertical size of the bounding box. Candidate regions are then paired and tested for a small vertical displacement and an appropriate horizontal separation. When this configuration of two small bounding boxes is detected, a pair of blinking eyes is hypothesized. The position in the image is determined from the center of the line between the bounding boxes. The distance to the face is measured from the separation. This provides the size of a window which is used to extract the face from the image. This simple technique has proven quite reliable for determining the position and size of faces [8].

Blink detection detection initially produces a vector of 8 parameters:

- v_l Vertical size of left rectangle.
- h_l horizontal size of left rectangle.
- v_r Vertical size of right rectangle.
- h_r horizontal size of right rectangle.
- v_s vertical separation of the rectangles
- h_s horizontal separation of the rectangles
- i horizontal part of mid-point between rectangles.
- j vertical part of mid-point between rectangles.

The midpoint between the rectangles is used as the observation X_b = [i, j]. A 2 x 2 covariance matrix for position $\mathbf{C}_b$ is given as a constant which was calibrated during system set-up. No estimate is produced for the horizontal and vertical extent of the face. The confidence of a blink detection, CF_b, is the resemblance of the eight parameters to an ideal prototype, P_{blink}, and its covariance $\mathbf{C}_b$. This prototype was computed by recording a large number of blink detection and removing any false detections by hand.

3.2. Detecting the Colour of Skin

Color histograms have been used in image processing for decades, particularly for segmenting multi-spectral satellite images, and medical images. In the early 1990's Swain and Ballard [9]showed that the intersection of color histograms was a reliable means of recognizing colored objects. Unfortunately, their technique is sensitive to the color and intensity of the ambient light source. Schiele and Waibul [8] have demonstrated that skin could be reliably detected by normalising the color vector by dividing out the luminance component.

A 2-D joint histogram of the luminance normalised color components (r, g) can be computed from a patch of an image known to be a sample of skin. For color components (R, G, B):

$$r = \frac{R}{R+G+B} \quad g = \frac{G}{R+G+B}$$

The histogram of normalised color gives the number of occurrences for each normalised color pair (r, g). This histogram must be periodically re-initialised to compensate for changes in ambient light, or differences in skin color of different users. In our early experiments with this technique, a cooperative user presented his face or hand to the camera to initialise the histogram in less than a second. In our latest system, the color sample is captured automatically whenever eye blink has been detected with a sufficient confidence.

A normalised color histogram h(r, g) based on a sample of N pixels, gives the conditional probability of observing a color vector $\vec{C}$ = (r, g), given that the pixel is an image of skin. $p(\vec{C} \mid \text{skin})$. Using Bayes rule, we convert this to the conditional probability of skin given the color vector, $p(\text{skin} \mid \vec{C})$. This allows us to construct a probability image in which each pixel is replaced by the probability that it is the projection of skin. An example is shown in figure 4. The center of gravity from the probability of skin gives the estimate of the position of the face. The bounding rectangles gives an estimate of size. A confidence factor is the computed by comparing the detected bounding box to an ideal width and height, using a Normal probability law. The average width and height and the covariance matrix are obtained from a number test observations selected by hand.

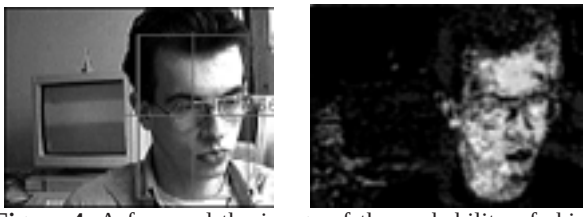

Figure 4. A face and the image of the probability of skin.

3.2 Tracking with Cross-Correlation

Detection of a face by color is fast and reliable, but not always precise. Detection by blinking is precise, but requires capturing an image pair during a blink. Correlation can be used to complete these two techniques and to hold the face centerred in the image as the head moves. Energy normalised cross-correlation tracking can be shown to be optimum in the presence of additive Gaussian noise [5]. The dominant noise in the case of face detection is neither Gaussian nor additive. However, when assisted by other detection processes, correlation tracking provides a technique which is inexpensive, relatively reliable, and formally analysable.

Correlation tracking processes are initiated by blink detection. The template for correlation is taken from the estimated position of the eyes. The search region for each tracker is estimated from the expected speed of the users movements measured in pixels per frame. This

value can be kept quite small if the frame rate is kept high [5]. Each reference template is a small neighborhood, W(m, n), of size Δx, Δy, of the image P(i, j) obtained during initialisation just after blink detection. In subsequent images, the reference template is compared to an image neighborhood (i, j), by computing the N by N template to the neighborhood of the image whose upper left corner is at (i, j). The system contains correlation processes using sum of squared difference (SSD) and energy normalised cross correlation (NCC). We have found the SSD generally gives superior results.

$$SSD(i, j) = \sum_{m=0}^{N} \sum_{n=0}^{N} (P_k(i+m,j+n) - W(m, n))^2$$

The estimated position of the target is determined by finding the position (i, j) at which SSD measure is a closest to zero. The actual center position can be determined by adding the half size of the mask to the corner position (i, j). By keeping the search region small, we obtain a processing rate of 25 hz. Figure 5 shows a typical map of the SSD values obtained when a template for the eye is convolved with a face. The local image of SSD values is inverted to provide the CF and Covariance. The covariance of the detection is estimated from the second moment of the inverted SSD values. A sharp correlation peaks give a small covariance, while a larger correlation gives a larger spread in covariance. The confidence is estimated from the peak value of the inverted SSD. When this confidence measure drops below a threshold the tracking processes is halted or re-initialised.

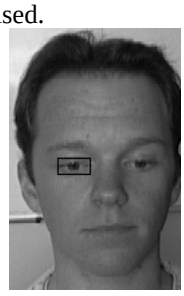

Figure 5a. Correlation template is taken from eye (detected from blink detection)

Figure 5b. Map of values from Sum of Squared difference with later image in sequence.

4 Coordination of Multiple Perceptual Processes.

The perceptual processes of eye blink detection, color histogram matching, correlation tracking, and sound localisation are complementary. Each process fails under different circumstances, and produces a different precision for a different computational cost. For example, eye blink is relatively inexpensive in cost and gives a precise localisation when it works, which is approximately once every 40 seconds. Thus eye blink is ideal for initialising, and re-initialising, the other tracking processes. Correlation tracking of the eyes is extremely fast when limited to a small search region and produces a precise result. However, experience shows that correlation will sometimes loose its track when the user turns his head more than about 15 degrees or makes a movement which is too sudden. In some cases, correlation can be recovered by enlarging the search region, but if this fails, another tracking mode is required.

Color histogram matching almost always produces a result, but tends to have an uncertainty of a few pixels. In particular, color histogram matching produces a reliable bounding box which can be used to limit other processes, including background suppression for incremental eigen-space compression. The probability of skin is computed by table lookup, but the connectivity analysis is relatively expensive. If computing cost were not a constraint, all processes would be run at each cycle. The fact is that computing cost is an important constraint. The fact that all three processes produce a confidence factor makes it possible for the supervisor to coordinate the different processes in order to maximize confidence and precision while minimizing computing cost.

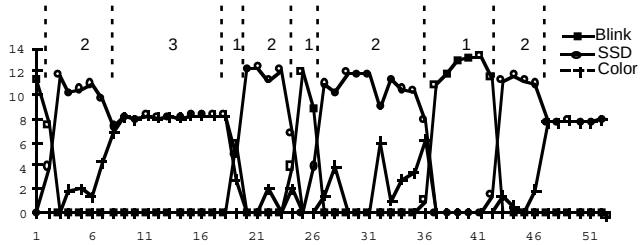

Figure 6. Cycles per second, for each of three processes, as the supervisor steps through the 3 states. Square is Blink Detection, o is correlation, + is histogram.

The control logic for the supervisor can be defined by a finite state machine. A current doctoral thesis in our group is investigating techniques to automatically generate such control graphs. In the mean time, we design control graphs by hand. At the time of writing of this paper, we obtain quite reliable tracking with the following states:

State 1) Initialisation: When tracking confidence is low, the supervisor runs blink detection to look for a face. When blink is detected, a color histogram is initialised, and a correlation mask is stored for each eye. The supervisor then shifts to state 2.

State 2) As long as the tracking CF remains high,

correlation is used to track the eyes. When the correlation CF drops below .05, control switches to state 3.

State 3) Re-initialisatin: Color histogram detection tracks the face while blink detection runs to try to re-acquire a correlation mask. If blink detects a face, the color histogram and correlation masks are re-initialised and the supervisor reverts to state 2. If blink fails to detect a face, then color histogram detection is run.

Figure 6 shows an example of the cycle time as the process moves through the three states. Throughout the entire process, the tracking precision, as measured by the covariance remained under 2 pixels.

5. Conclusions

The integration of complementary visual processes can produce a reliable and robust system, provided that all processes produce a confidence factor and an error estimate. Integration and coordination requires an architecture. Such an architecture is provided by the Synchronous Ensemble of Reactive Visual Processes model developed in the VAP project [3]. Coordination of visual processes requires signalling visual events to the supervisory controller. An important class of visual events are the confidence which each process can gives for its results. Fusion of results is made possible by estimating error bounds for each process in the form of a covariance matrix. Individual processes can be grouped into states, which permits the control logic of the system to be designed as a state transition graph. Improved methods are required for the design of such state transition graphs.

Acknowledgements:

This work has been supported by France Telecom CNET (Project COMEDI), based on results from Project ESPRIT EP 8212 "VIsion as Process".

Bibliography

[1] J. L. Crowley, P. Stelmaszyk, T. Skordas and P. Puget, "Measurement and Integration of 3-D Structures By Tracking Edge Lines", International Journal of Computer Vision, Vol 8, No. 2, July 1992.

[2] J. L. Crowley and Y. Demazeau, "Principles and Techniques for Sensor Data Fusion", Signal Processing, Vol 32 Nos 1-2, p5-27, May 1993.

[3] J. L. Crowley and H. I Christensen, Vision as Process, Springer Verlag, Heidelberg, 1994.

[4] J. L. Crowley and J. M. Bedrune, "Integration and Control of Reactive Visual Processes", 1994 European Conference on Computer Vision, (ECCV-'94), Stockholm, may 94.

[5] J. L. Crowley and J. Martin, "Experimental Comparison of Correlation Techniques", IAS-4, International Conference on Intelligent Autonomous Systems, Karlsruhe, March 1995.

[6] J. L. Crowley, F. Bérard and J. Coutaz, "Finger Tracking as an Input Device for Augmented Reality", IWAGFR '95 - International Workshop on Gesture and Face Recognition, Zurich, June 1995.

[7] H. Inoue, T. Tashikawa and M. I. Inaba, "Robot vision system with a correlation chip for real time tracking, optical flow, and depth map generation", The 1992 IEEE Conference on Robotics and Automation, Nice, April 1992.

[8] B. Schiele and A. Waibul, "Gaze Tracking Based on Face Color", IWAGFR '95- International Workshop on Face and Gesture Recognition, Zurich. July 1995.

[9] M. J. Swain and D.H. Ballard, "Color Indexing", International Journal of Computer Vision, Vol 7, No 1, 1991.