A HIERARCHICAL FRAMEWORK FOR FACE TRACKING USING STATE VECTOR FUSION FOR COMPRESSED VIDEO

Jun Wang, Radhakrishna Achanta, Mohan Kankanhalli, Philippe Mulhem
National University of Singapore
School of Computing
Singapore 117543
{wangj, achanta, mohan, mulhem}@comp.nus.edu.sg

ABSTRACT

Faces usually are the most interesting objects in certain categories of video like home videos and news clips. In this paper a novel sensor fusion based face tracking system is presented that tracks faces in compressed video, and aids automatic video indexing. Tracking is done by fusing the measurements from three independent sensors – motion and colour based trackers (derived from [2]) and a face detector (presented in [1]) using a novel hierarchical framework based on Kalman filter state vector fusion. The tracking results show that the fused results are better than those of any individual sensors or their mean.

1. INTRODUCTION

In most videos, visual features related to human activities are usually the most important content descriptors. The work presented here, talks about an automatic indexing tool, a face tracker, which is a part of the Digital Image and Video Album (http://diva.comp.nus.edu.sg:8080) project. There are many face tracking approaches in literature ([3],[4]) focused at applications like object orientated image coding, surveillance/security, expression recognition and man-machine interaction. Assumptions like constant background or presence of a single face often render them ineffective in other genres of video. Also, there is glaring absence of the use of information available due to the added dimension of time in current video based face detectors [5]. The system presented in this paper avoids these assumptions and takes a sensor fusion approach to tracking faces. It does not rely on any single sensor, as it could be inaccurate. Instead, the premise that multiple sensors (one from face detection and two from tracking) with their respective inaccuracies can produce a good overall result is relied upon. Here advantage is taken both of spatial information coming from face detector and the temporal correlation obtained from object tracker. In addition, this approach uses a feedback loop where the output of the previous tracking result helps the next one.

2. SYSTEM DESIGN

Current face detectors may not robust enough for tracking face(s) in an unconstrained scene of a real-life video because the temporal correlation of face regions is not taken into consideration. Face trackers similarly might be handicapped because of the possible absence of a robust face modeling (or detection) method (sometimes merely skin colour detection is used). The mutual limitations are overcome in this work using a face detector and a tracker (with two tracking components, here) helping each other. A Kalman filtering framework is used for the purpose of *estimation* in non-intracoded frames and for *sensor fusion* in I frames, to achieve this. Before this framework is explained, a brief introduction to three sensors is in order.

2.1. Visual Sensors

There are three visual sensors: neural network based face detector, a motion vector based object tracker and a color based object tracker (MOT and COT hereafter), all operating in the compressed domain. The output of each sensor is a rectangle, which locates the position of the face. In the beginning, the object trackers use the output of the face detector as the starting point.

2.1.1. Face detecting sensor
In the algorithm [1], a statistical skin region filter is initially used to filter out skin colour regions in I frames using chrominance DC information. A compressed domain neural network based face detector scans these regions to classify face and non-face regions using the luminance DCT coefficients.

2.1.2. Motion vector based object tracking sensor
Tracking is done within the Group of Pictures (GOP) using forward motion vectors of P and B pictures. Since I frames do not have motion vectors, to cross over to the GOP boundary starting with a new I frame, the backward motion vectors of the last B frame are used. Details of the algorithm are available in [2].

2.1.3 Color based object tracking sensor

This is a colour based matching method. Normalized histograms with sixteen bins for the DC values and the first eight AC values for all the blocks comprising the face region of the starting I frame are matched with similar histograms for the regions in a search area in each following I frame, to find the best matching face region in them (refer to [2]). This method is also able to take into consideration changes in size of the tracked object.

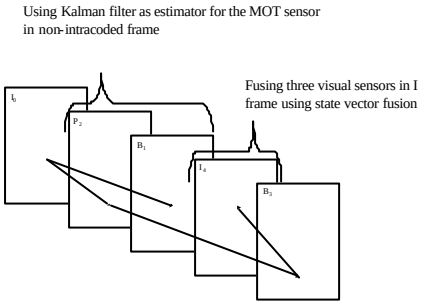


Figure 1. Kalman Filter Based Face Tracking Framework In Compressed Video

2.2. Tracking Framework

The measurements from the three sensors are not always available in each frame (measurements from MOT are available in each frame, but the measurements of face detector and COT are available only in I frames). Therefore in the framework shown in figure 1, Kalman filter based estimation is applied to smoothen the measurement of MOT in non-intracoded frames, and then *state vector fusion* approach is used in I frames to fuse three noisy sensor measurements to get optimal results.

2.2.1. Estimation in non-intracoded frames

Since there is only the MOT measurements existing in non-intracoded frames, we use a Kalman filter to estimate the true positions of human faces. A discrete-time dynamical system (like the movements of faces in videos) can be presented using graph model (Dynamic Bayesian Networks (DBNs)) as shown in figure 2. If the dynamic system is assumed to be linear and subject to Gaussian noise (arrows are governed by equation 1 and 2) the DBNs becomes a Kalman filter.

$$x(k+1) = \Phi x(k) + w \quad (1)$$

with measurements $z(k)$ at time instant k given by

$$z(k) = Hx(k) + v \quad (2)$$

where, $x(k)$ is state vector at time k, which consists of all parameters that are estimated by the filter (e.g. position, velocity); Φ is the state transition matrix, H is the measurement matrix; w and v are zero-mean normally distributed random variables with covariance matrices Q and R, respectively.

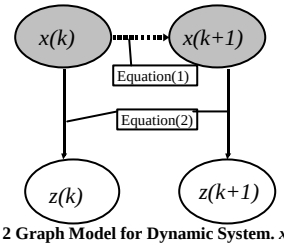


Figure 2 Graph Model for Dynamic System. *x(k)* is state to be estimated in time *k* (can not be observed). *z(k)* is the measurement of *x(k)* at time *k*. This system can be modeled by Kalman filters when arrows are governed by equation (1) and (2).

In non-intracoded frames (where only the MOT measurements are available), $x(k)$ provides the human face positions in video (represented as locations or velocity of movements) in frame k, and $z(k)$ gives the sensor measurement of $x(k)$ from the MOT sensor. Utilizing the measurement $z(k)$ from the MOT sensor, the optimal true face position $x(k)$ can be estimated as the expectation $E[x(k)|z(0),\ldots,z(k)]$ by the Kalman filtering algorithm.

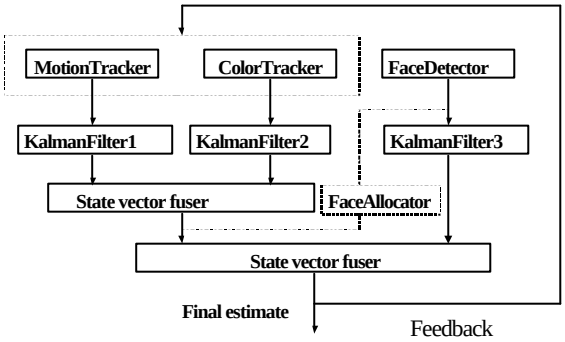


Figure 3. Two-stage fusion framework in I frames

Feedback :The final fused results are fed back to the tracking sensors as the correction mechanism. It helps to stop the error accumulation of object trackers

FaceAllocator: With the help of the first stage of object tracker sensor fusion result, it allows us to assign each face detected (by the face detector) in the incoming I frame to the one being tracked by the tracking sensors. This is for tracking multiple faces simultaneously.

2.2.2. State vector fusion framework in I frames

As stated before, Kalman filters can be used for sensor fusion apart from the usual estimation problems. To fuse sensors, two methods can be used: *measurement fusion* or *state vector fusion*. *Measurement fusion* needs the assumption that individual sensor measurements are independent, which is not the case for the three sensor measurements being used here. Therefore in this paper, the *state vector fusion* approach is presented. Experiments in [6] have also proven that *state vector fusion* works better for this case.

2.2.2.1. A hierarchical fusion framework

The proposed hierarchical sensor fusion framework is shown in figure 3. In the first stage, the measurements from the tracking sensors (MOT,COT) are fused. In the second stage, the measurements from the face detector sensor are

fused with the output of the first stage to get the final result. There are two advantages to using this two-stage fusion framework:

1. With the help of result of the first stage of sensor fusion, face detected (by the face detector) in the incoming I frame (in a multi-face scenario) is associated to the one being tracked by the tracking sensors (shown in the figure 3 as face allocator, FA). This problem of assigning the newly detected face to the right track cannot be resolved by the face detector itself.
2. The final fused results are fed back to the tracking sensors as the corrective feedback as shown in figure 3. Since this feedback is only for object trackers, not face detector, we call it partial feedback, and it helps prevent error accumulation.

State vector fusion in the first stage of fusion is described in section 2.2.2.2. The fusion mechanism in the second stage is same as in the first stage.

2.2.2.2. State vector fusion in first stage

In the case of two sensor measurements coming from MOT and COT, the *state vector fusion* can be shown as figure 4. There are two Kalman filters used for the two sensors. In this graph, there are two more types of hidden units (*xc(k)* and *xc(k+1)* for the state variable of the COT sensor in time k and k+1 respectively; *xm(k)* and *xm(k+1)* for the state variable of the MOT sensor in time *k* and *k+1* respectively). Hidden units *x(k)* and *x(k+1)* represent the final true state, which can mainly be inferred from *xc* and *xm.* The aim here is to get the estimate of *x(k)* at time *k*. Using these two Kalman filters, hidden state *xc(k)* can be estimated as expectation *xc(k|k)* (i.e. E[*xc(k)*| *zc(0),…, zc(k)*]) utilizing measurements *zc* from the COT sensor, and hidden state *xm(k)* can be also estimated as the expectation *xm(k|k)* (i.e.E[*xm(k)*| *zm(0),…,zm(k)*]) utilizing measurements *zm* from the MOT sensor. By using maximum likelihood as a fusion strategy, the best estimated fused data *x(k|k)*(i.e.E[*x(k)*|*zc(0),…,z(k); zm(0, … , zm(k)*)] is derived from equation (3).

$$x(k \mid k) = \frac{P_m(k \mid k)xc(k \mid k) + P_c(k \mid k)xm(k \mid k)}{P_c(k \mid k) + P_m(k \mid k)} \qquad (3)$$

$$P(k \mid k) = \frac{P_c(k \mid k)P_m(k \mid k)}{P_c(k \mid k) + P_m(k \mid k)} \qquad (4)$$

where, *x(k|k)* is the best estimated fused data at time *k*, *Pc* and *Pm* are estimated state vector's covariance matrices for COT and MOT, respectively, at time *k*, and *xc(k|k)* and *xm(k|k)* are the estimated state vectors of Kalman filters for COT and MOT, respectively. The covariance matrix of fused results *P(k|k)* can be obtained using equation (4).

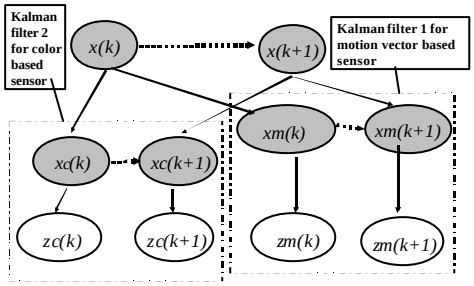


Figure 4 Graph Model for State Vector Fusion

2.2.2.3. Noise Modeling in Kalman filers

Both the positions of the faces *X*=[*x(0)…x(N)*] and their measurements *Y*=[*y(0)…y(N)*] are available from training data. Therefore the process and measurement noise (R,Q) can be modeled using the ML algorithm by maximizing $L(X,Y,q) =$

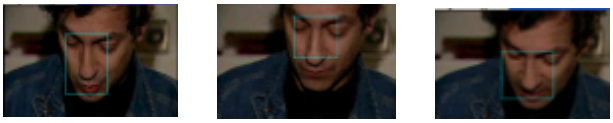
$$-\sum_{k=1}^{N}\{\log|Q|+[x(k)-\Phi x(k-1)]^T Q^{-1}[x(k)-\Phi x(k-1)]\}$$

$$-\sum_{k=0}^{N}\{\log|R|+[y(k)-Hx(k)]^T R^{-1}[y(k)-Hx(k)]\}+C \qquad (5)$$

Here q is the parameter needing to be estimated; *N* is the number of the frames in the training data; *C* is a constant.

3. EXPERIMENTS AND RESULTS

Tracking of faces was done in each of the test video clips using two-stage sensor fusion framework in I frame and Kalman filter estimation in non-intracoded frames. Figure 5 shows some test results on MPEG 7 test set clips.



Test video 1. (MPEG 1 formatted; Picture size 352x288)

Test video 1. (MPEG 1 formatted; Picture size 352x288)

Figure 5 Sample frames with test results

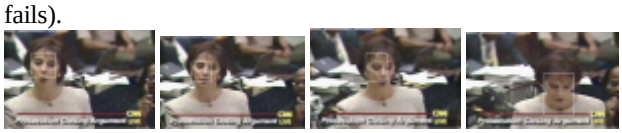
In order to compare with other approaches in compressed domain [5], the clip "Marcia"* (CNN news clip; MPEG-1 352x240 pixels frame size, 556 frames, 38 I-frames) is used.

Fused results #1 | Estimated by MOT #2 | Estimated by MOT#3 | Fused results #4

Figure 6 Sample frames for video "Marcia" with tracking

* Courtesy of S. F. Chang and H. S. Wang, Columbia University

Figure 6 shows the scenario when the face detector helps the object trackers. Note that in the I frames (#1 and #4) of figure 6, the result of tracking is after the stage two fusion where the measurement from the three sensors have been fused. In the non-intracoded frames (#2,#3 of figure 6), the tracking results are obtained only from MOT using Kalman filters for the estimation purpose. Any error accumulated by the MOT results can be corrected by the fused result in the following I frame as shown in last sample frame (#4) of figure 6. For a typical MPEG video (30 frames/second, 15 frame GOP), the partial feedback mechanism in our framework is able to correct the accumulated error, if any. Figure 7 shows a case when object trackers locate the face when the face detector actually fails to do so (frame #4)(i.e. fused results from stage one only when the face detector fails).



Fused results #1 | Estimated by MOT #2 | Estimated by MOT #3 | Recovered by object trackers in I frame #4

Figure 7 Sample frames for video "Marcia" with tracking

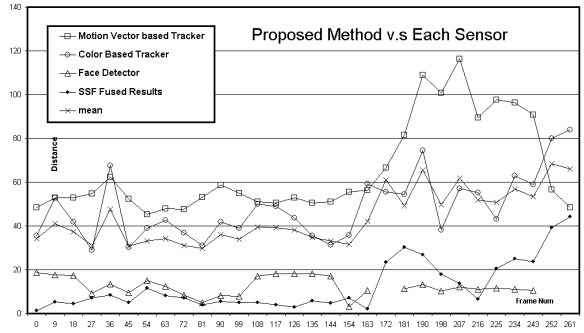


Figure 8 Tracking errors (distance) of each sensor and fused

In order to analyze the performance of the fusion results (figure 8) with respect to each individual sensor and to observe how the limitations of each sensor are overcome, the distance between the centroid of the rectangles of the three individual sensors and those of the fused output are plotted against the centroid of the ground truth in the test video 1 (the screen shots shown in figure 5). The sensor fusion based results outperform MOT and COT as well as their mean. It is also evident in figure 8, that a good result can be obtained by object trackers alone when face detector fails in a certain I frame (the gap of face detector sensor in figure 8). The performance of the system in the test video 2 (the screen shots in figure 5) is shown by plotting *y* positions of ground truth and fusion results in figure 9.

4. CONCLUSIONS

The results from our proposed framework are better than the results of the face detector and tracker independently. The trackers and face detector assist each other to mutually compensate for their respective shortcomings. There are several things that can be done to improve the performance of the system in the future: the performance of the individual trackers itself can be improved; the face dynamics can be modeled non-linearly; other sensor fusion approaches like Bayesian networks can be used in place of Kalman filters.

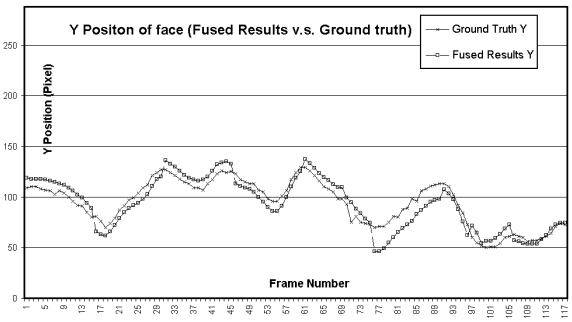


Figure 9 Fused results v.s. Ground truth (Y position)

5. REFERENCES

[1] J. Wang, M. S. Kankanhalli, P. Mulhem, and H. Abdulredha, "Face Detection Using DCT Coefficients in MPEG video," In *proceedings of International Workshop on Advanced Image Technology*, at Hualien, Taiwan, January 2002.

[2] R. Achanta, M. Kankanhalli, and P. Mulhem, "Compressed Domain Object Tracking for Automatic Indexing of Objects in MPEG Home Video," *IEEE Intl. Conf. on Multimedia and Expo*, Lusanne, Switzerland, August 2002.

[3] K. Toyama, "Prolegomena for Robust Face Tracking," *Workshop on Automatic Facial Image and Analysis and Recognition Technology (ECCV 98)*, 1998.

[4] S. Spors, and R. Rabenstein, "A Real-Time Face Tracker For Color Video," *IEEE Int. Conf. On Acoustics, Speech & Signal Processing (ICASSP)*, Utah, USA, May 2001.

[5] H. S. Wang, and S. F. Chang, "FaceTrack: Tracking and Summarizing Faces from Compressed Video," *SPIE Multimedia Storage and Archiving Systems IV*, 19-22 Sept, Boston, MA, 1999.

[6] J. Wang, "Detecting and Tracking Human Faces in Compressed Domain for Content Based Video Indexing," Master Thesis, School of Computing, National University of Singapore, 2002.