

A Joint System for Person Tracking and Face Detection

Zhenqiu Zhang*, Gerasimos Potamianos, Andrew Senior,
Stephen Chu, and Thomas S. Huang*

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA

zzhang6@uiuc.edu, {gpotam, aws, schu}@us.ibm.com, huang@ifp.uiuc.edu

Abstract. Visual detection and tracking of humans in complex scenes is a challenging problem with a wide range of applications, for example surveillance and human-computer interaction. In many such applications, time-synchronous views from multiple calibrated cameras are available, and both frame-view and space-level human location information is desired. In such scenarios, efficiently combining the strengths of face detection and person tracking is a viable approach that can provide both levels of information required and improve robustness. In this paper, we propose a novel vision system that detects and tracks human faces automatically, using input from multiple calibrated cameras. The method uses an Adaboost algorithm variant combined with mean shift tracking applied on single camera views for face detection and tracking, and fuses the results on multiple camera views to check for consistency and obtain the three-dimensional head estimate. We apply the proposed system to a lecture scenario in a smart room, on a corpus collected as part of the CHIL European Union integrated project. We report results on both frame-level face detection and three-dimensional head tracking. For the latter, the proposed algorithm achieves similar results with the IBM “PeopleVision” system.

1 Introduction

Visual detection and tracking of humans in complex scenes is a very interesting and challenging problem. Often, input from multiple calibrated cameras with overlapping fields of view is available synchronously, and information about both the frame-view and space-level human location is desired. One such scenario of interest, considered in this paper, is human-computer interaction in smart rooms, where a speaker is presenting a seminar in front of an audience. The scenario is of central interest within the CHIL European Union integrated project, “Computers in the Human Interaction Loop” [1]. In data collected as part of the CHIL project, four fixed calibrated cameras located at the corners of a smart room capture video data, with the goal of locating and identifying the seminar presenter. Hence, both three-dimensional head position estimation, as well as

* Zhenqiu Zhang and Prof. Thomas Huang are with the Beckman Institute, University of Illinois, Urbana, IL 61801, USA. This work was performed while Zhenqiu Zhang was on a summer internship with the Human Language Technologies Department at the IBM Thomas J. Watson Research Center, and was supported by the European Commission under the integrated project CHIL, “Computers in the Human Interaction Loop”, contract number 506909.

face detection at the available frame views is required. The information can be further utilized to obtain close-up views of the presenter, based on steerable pan-tilt-zoom cameras, in the seminar indexing and annotation, etc. Clearly therefore, in such a scenario, a visual system that combines face detection, tracking, and multicamera processing is feasible and desirable.

Significant research work has been devoted to the problems of face detection, single-camera and multicamera tracking. For face detection, a machine learning based approach has proved the most effective. For example, Rowley et al. [2] present a face detection system based on retinally connected neural networks (NNs), accepting as input the preprocessed image pixel values directly. Post-processing is then performed by appropriately combining the NN outputs by “and”/“or” operators, or by using an additional neural network to arbitrate them. Roth et al. [3] use a network of linear units. The SNoW learning architecture is specifically tailored for learning in the presence of a very large number of features. The system of Viola and Jones [4] makes a successful application of AdaBoost to face detection. The resulting system is an efficient, real-time frontal-view face detector.

The ability to detect faces under varying head pose is important in many real applications. A reasonable treatment for multiview face detection is a view-based method within the appearance-based framework. In the system of Schneiderman and Kanade [5], multiresolution information is used for different levels of a wavelet transform. The algorithm consists of an array of five face detectors in the view-based framework. Each is constructed using statistics of products of histograms computed from examples of the respective view. In general, while great success has been achieved for frontal-view face detection, much engineering work is needed for real-time multiview face detection.

For tracking, there also exist many successful algorithms proposed in recent years. In [6], Comaniciu et al. introduce an algorithm for real-time tracking of non-rigid objects seen from a moving camera. The central computational module is based on the mean shift iterations and finds the most probable target position in the current frame. In [7], Isard and Blake propose a new algorithm, called “condensation”. The algorithm uses “factored sampling”, previously applied to the interpretation of static images, in which the probability distribution of possible interpretations is represented by a randomly generated set. The method uses learned dynamical models, together with visual observations, to propagate the random set over time. The result is highly robust tracking of agile motion that runs in near real-time.

Multiple cameras provide additional information concerning the objects of interest, which could be used to improve the tracking system performance. Many multicamera tracking algorithms have been proposed in recent years. For example, in [8], Black and Ellis present such a method in the context of surveillance. Their approach exploits multicamera views to resolve object occlusion. Moving objects are detected by using background subtraction, and viewpoint correspondence between the detected objects is established by using ground plane homography. In [9], Hampapur et al. use two or more calibrated cameras to triangulate a moving object’s position, originally obtained by background subtraction, and determine the steering parameters for a third, pan-tilt-zoom camera that is calibrated to the same coordinate system. The pan-tilt-zoom camera automatically acquires zoomed-in views of a person’s head, while the person is in mo-

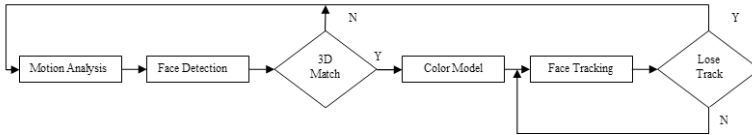


Fig. 1. Block diagram of the proposed multicamera face detection and tracking system

tion within the monitored space. An extended version of this system is also considered in this paper for the CHIL seminar presenter tracking task.

Even though there exist so many successful algorithms for face detection and tracking, how to efficiently combine their strengths, in order to obtain robust and real-time performance within a multicamera framework, is still an interesting and very important problem. Our work focuses on exactly this problem, in the context of the CHIL seminar task discussed at the beginning of this section. In particular, we propose to solve the problem by taking advantage of multiview face detection, color-based mean shift tracking, motion analysis, and utilizing calibration information for the available camera views. The overview diagram of our approach is depicted in Fig. 1: First, a three-dimensional face detection and tracking system is initialized by combining motion history, calibration information, and multiview face detection [10]. Subsequently, for tracking, a face model is constructed based on the color histogram of the face region in the hue/saturation/value (HSV) color space, and mean shift tracking [6] is applied to track the presenter’s face in different views independently. At each frame, the tracking component is verified by local face detection and the calibration information. If it is determined that the tracking system has lost track, the detection and tracking system is re-initialized and the face model updated.

The paper is organized as follows: Section 2 presents FloatBoost [10] learning, which is applied to multiview face detection. The mean shift iteration algorithm [6] for face tracking is discussed in Section 3, and the proposed, combined vision system for multiview face detection and tracking is described in Section 4. For comparison purposes, an alternative approach based on the IBM “PeopleVision” system is described in section 5. Face detection and tracking experiments are presented in section 6, and conclusions are drawn in Section 7.

2 FloatBoost Learning for MultiView Face Detection

In [4], the AdaBoost algorithm was successfully applied to frontal face detection resulting in the first real-time frontal face detection system. However, if considered as a feature selection algorithm, AdaBoost is a sequential forward search procedure, which suffers from the so-called “nesting effect”. Attempts to prevent the nesting of feature subsets have led to the development of floating search methods. FloatBoost [10] incorporates the idea of “floating search” [11] into AdaBoost to solve the “nesting effect” encountered in the sequential search of AdaBoost. A quality improvement of the selected features is gained at the cost of increased computation due to the extended search. In this paper, the FloatBoost learning algorithm is applied to multiview face detection in the smart room seminar scenario.

2.1 AdaBoost Learning

For two class problems, let us assume that a set of N labeled training examples is available, $(x_1, y_1), \dots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example x_i . For face detection, x_i is an image sub-window of a fixed size (e.g. 20×20) containing an instance of the face ($y_i = +1$), or nonface ($y_i = -1$) pattern. In the notion of RealBoost (a real version of AdaBoost [4] as opposed to the original discrete one), a stronger classifier is a linear combination of M weak classifiers

$$H_M(x) = \sum_{m=1}^M h_m(x), \quad (1)$$

where $h_m(x) \in \mathfrak{R}$ are weak classifiers. The class label for a test x is obtained as $H(x) = \text{sign}\{H_M(x)\}$ (an error occurs when $H(x) \neq y$), while magnitude $|H_M(x)|$ indicates the confidence.

In boosting learning [12, 13], each example x_i is associated with a weight w_i , and the weights are updated dynamically using a multiplicative rule according to the errors in previous learning, so that more emphasis is placed on the examples that are erroneously classified by the weak classifiers learned previously. This way, the new weak classifiers will “focus more attention” to those examples. The stronger classifier is obtained as a proper linear combination of the weak classifiers.

The “margin” of an example (x, y) achieved by $H(x)$ (a single or a combination of weak classifiers) on the training examples can be defined as $yH(x)$. This can be considered as a measure of the confidence of the h 's prediction. The following criterion measures the bound on classification error [13]

$$J(H(x)) = E_w(e^{-yH(x)}) = \sum_{i=1}^N e^{-y_i H(x)}, \quad (2)$$

where $E_w(\bullet)$ stands for the mathematical expectation with respect to weight w over the examples (x_i, y_i) .

The AdaBoost method constructs $h(x)$ by stage-wise minimization of (2). Given the current $H_{M-1}(x) = \sum_{m=1}^{M-1} h_m(x)$, the best $h_M(x)$ for the new strong classifier $H_M(x) = H_{M-1}(x) + h_M(x)$ is the one which leads to the minimum cost

$$h_M = \text{arg min}_{h^+} J(H_{M-1}(x) + h^+(x)). \quad (3)$$

2.2 FloatBoost Learning

AdaBoost is a sequential forward search procedure using the greedy selection strategy. Its heuristic assumption is the monotonicity. The premise offered by the sequential procedure can be broken down when the assumption is violated. FloatBoost [10] incorporates the idea of floating search [11] into AdaBoost to overcome the non-monotonicity problems associated with AdaBoost. The sequential floating search (SFS) method [11] allows the number of backtracking steps to be controlled instead of being fixed beforehand. Specifically, it adds or deletes $l = 1$ feature and then backtracks r steps, where

r depends on the current situation. It is such a flexibility that amends limitations due to the non-monotonicity problem. A quality improvement of the selected features is obtained at the cost of increased computation due to the extended search. The SFS algorithm performs very well in several applications [14]. The idea of floating search is further developed in [15], by allowing more flexibility for the determination of l .

These feature selection methods, however, do not address the problem of (sub-) optimal classifier design based on the selected features. FloatBoost combines them into AdaBoost for both effective feature selection and classifier design.

Let $H_M = \{h_1, \dots, h_M\}$ be the so far best set of M weak classifiers; $J(H_M)$ be the criterion which measures the overall cost of the classification function $H_M(x) = \sum_{m=1}^M h_m(x)$ build on H_M ; J_m^{\min} be the minimum cost achieved so far with a linear combination of m weak classifiers for $m = 1, \dots, M_{\max}$, where M_{\max} denotes the maximum number of features (iterations) allowed in the boosting algorithm, initially set to a large value before the iteration starts. The FloatBoost learning [10] procedure involves training inputs, initialization, forward inclusion, conditional exclusion and output.

In the forward inclusion step, the currently most significant weak classifier is added one at a time, which is identical to AdaBoost. In the conditional exclusion step, FloatBoost removes the least significant weak classifier from H_M , subject to the condition that the removal leads to a lower cost than J_{M-1}^{\min} . Supposing that the removed weak classifier was the m' -th in H_M , then $h_{m'}, \dots, h_M$ will be re-learned. The above steps are repeated until no more removals can be performed.

3 Mean Shift Tracking

Computational complexity is critical to most tracking applications, and therefore, exhaustive search in the neighborhood of the predicted target location for the best candidate is in most cases prohibitive. As a solution to this problem, mean shift tracking has been proposed [6]. Mean shift tracking is a real-time algorithm that aims to maximize the correlation between two statistical distributions.

The correlation or similarity between two distributions is expressed as a measurement derived from the Bhattacharyya coefficient [6]. Statistical distributions can be built using any characteristic discriminating to a particular object of interest. A model might use color, texture, or include both. In this paper, we model the target using the H channel in the HSV color space.

The discrete density q is estimated from the m -bin H-channel histogram of the HSV color in the face region, and p is estimated at a given location y from the m -bin histogram of the face candidate. The sample estimate of the Bhattacharyya coefficient is given by

$$\hat{\rho}(y) \equiv \rho(\hat{p}(y), \hat{q}) = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u}. \quad (4)$$

The distance between the two distributions can then be defined as

$$d(y) = \sqrt{1 - \rho(\hat{p}(y), \hat{q})}. \quad (5)$$

Starting at the predicted location y of the target computed by Kalman filtering [17], we search for the new target location in the current frame using mean shift iterations.

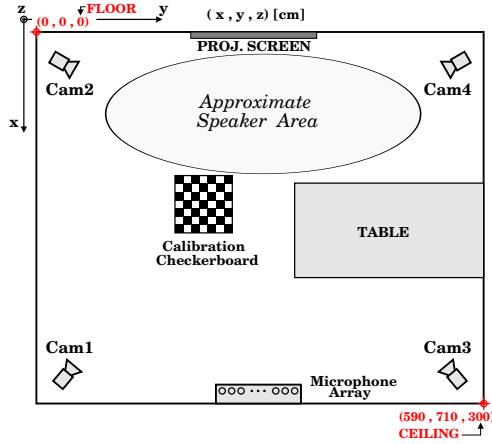


Fig. 2. Schematic diagram of the audio-visual sensors installed in the smart room at the University of Karlsruhe, Germany, as part of the CHIL project [1]. Data recorded by the four fixed cameras cam1–cam4 are used in our experiments

To minimize the distance (5), or, equivalently, maximize the Bhattacharyya coefficient, the Taylor expansion of p is used around the value of the coefficient at the target predicted position y_o . This yields

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{y}_o) \hat{q}_u} + \frac{1}{2} \sum_{u=1}^m \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_o)}}. \quad (6)$$

Position y_o can be just set to the current object position, or can be determined by Kalman filtering, as in our system.

The first term in (6) is independent of y , and the second term can be efficiently achieved based on mean shift iterations. This iterative optimization maximizes the value of Bhattacharyya coefficient in equation (4). At each iteration, a new location of the object is derived based on the mean shift vector [6]. Compared with exhaustive search, this iterative optimization algorithm is very efficient, with typically only several iterations needed to find the optimal target location.

4 Joint MultiCamera Face Detection and Tracking

As discussed in the Introduction, in this paper, we consider a particular scenario, where multiple calibrated cameras are set up in a smart room. The goal is to detect and track the presenter's face in the camera views (whenever visible), and to track the position of the presenter's head in the three-dimensional smart room space.

The setup of the room is as depicted in Fig. 2. This corresponds to the smart room located at the University of Karlsruhe, Germany, one of the CHIL project partners [1, 19]. Similar rooms are being set up in a number of CHIL project partners [1]. The room dimensions are 5.90×7.10 m, with a ceiling height of 3 m. A number of sensors are installed in the room which include the four fixed cameras, providing the data used in



Fig. 3. Example frames captured by the four fixed cameras of the CHIL smart room

this paper. The cameras are located at the corners of the room, at about 2.7 m height (see also Fig. 2). They are SONY DFW-V500, and capture color data at a 640×480 pixel resolution and 15 frames per second through a firewire interface. For storage purposes and ease of access, JPEG compression is used for each frame. The cameras are synchronized and calibrated by the calibration toolbox [18], hence their relative position and orientation are known. A sample of synchronized images from the four camera views is shown in Fig. 3.

To detect and track the location of the presenter in the seminar room, three components are integrated in the proposed visual system: Initialization, tracking, and a reinitialization decision (see also Fig. 1). Notice that in the currently implemented algorithm, only two camera views are utilized. The three components, based on processing the two views, are described in more details in the following.

4.1 Initialization

In the initialization stage, we use three primary vision modules: Motion analysis, the camera model (based on calibration information), and face detection.

Motion History: First, independently for each camera view, motion history is estimated to rapidly determine where movement has occurred. The utilized algorithm is based on work by Davis and Bobick [16]. Obtaining a foreground silhouette is achieved through subtraction between two consecutive frames instead of background subtraction. As the person moves, the most recent foreground silhouette is copied as the highest value in the motion history image. The result is called the “motion history image” (MHI). MHI pixel values that fall below that threshold are set to zero. An example of the algorithm applied to two camera views is depicted in Fig. 4(a).



Fig. 4. (a) Motion history image (MHI) examples in two camera views; motion objects are segmented as foreground (white pixels). (b) Multiview face detection result when the detectors are applied locally around the foreground region. (c) Local multiview face detection in the tracking stage: Faces are detected within windows around the tracking results

Multi View Face Detection: Due to the size of the room, the resolution of the presenter's face is quite small in most of the CHIL data considered (typically, around 20×20 pixels). Robust face detection becomes difficult for such low resolutions, and as a result, the face detector threshold is adjusted to low values in order to keep high detection rate. This of course increases the false alarm rate. To deal with the problem, the multiview face detector is only applied to the foreground region, where motion occurred. Example detection results are depicted in Fig. 4(b). The detection results at each view can then be used to verify whether the detected faces belong to the same person. In particular, our system uses two trained face detectors: One for frontal view and the other for the left side view, since the right side face detector can be easily obtained by mirroring the left side one.

Camera Model: Recording images using a camera is equivalent to mapping three-dimensional object space points to image points in the film plane. For digitization, this recorded image will be projected again to the image in the projection plane. For simplicity, it is possible to directly relate the projected image and the object by a ray through the projection center.

In our system, given the results of face detection, rays are created by connecting the projection center with the head center at each camera view. The inter-ray distance is subsequently calculated. If the detection results are correct in both views, then they belong to the same person, and the inter-ray distance should be small; otherwise, the distance should be larger than a threshold, and the detection result erroneous. Based on this assumption, one can verify the detection results of the previous stage.

4.2 Tracking

Following the initialization component and the successful location of the presenter's face, the algorithm switches into its tracking mode. A color-based face model is first created of the detected face region for tracking in each of the two camera views. In this paper, the HSV color space is used for this purpose. In particular, the face model is created by the one-dimensional histogram of the H component in the HSV color space. The mean shift iteration is subsequently applied in the two view images separately to find the best target candidate.

4.3 Decision to Reinitialize

In our system, the re-initialization decision is based on local face detection and utilizes the camera model. In more detail, at each frame, a multiview face detector is applied around the tracking result to determine whether there is a face object in the local region, as shown in Fig. 4(c) (in our system, this is a 80×80 pixel region). If a face could not be detected in the local region for several frames, a re-initialization decision is taken. Similarly, if the inter-ray distance of the two-camera rays is larger than a predetermined threshold, this indicates that the two tracked results are inconsistent, hence prompting re-initialization. In our system, such decision is taken every 5 to 10 frames.

4.4 System Specifics

The resulting algorithm runs at approximately 5 frames per second on a Pentium four, 2.8 GHz desktop, with 512 MBytes of memory. Nevertheless, the current implementation is not optimal, and we believe that the algorithm speed can be substantially improved.

As already mentioned, the algorithm operates using two camera views, but of course can be readily extended to accommodate more camera inputs. For the CHIL data, it uses inputs from the cam1+cam2 or the cam3+cam4 cameras, depending on which set contains higher percentage of frontal faces, as determined on a development data set (see Section 6).



Fig. 5. Typical operation of the proposed multicamera face detection and tracking system. Depicted in left column: Frame 1: Motion analysis. Frame 5: Tracking initialization. Frame 283: Continued tracking. Right column: Frame 324: Lost tracking detected. Frame 331: Foreground segmented by motion analysis. Frame 334: Tracking re-initialization

An example of the tracking algorithm applied on the CHIL data is depicted in Fig. 5: In frame 5, motion analysis is applied and the foreground object is segmented. The presenter’s face is located by the multiview face detector, and the detection result is verified by the camera model. Subsequently, the face color model is constructed in the detected face region. In the next hundreds of frames, color-based mean shift tracking successfully locates the presenter’s face in the two camera views. However, by frame 324, the tracking has failed. Therefore, the system returns to its initialization component, and at frame 334 re-emerges in the tracking mode, after successfully applying re-initialization.

5 A Background Subtraction Tracker

For comparison purposes, we briefly present an alternative head tracking system, based on the IBM “PeopleVision” system [9], properly modified for use on the CHIL data. The system uses background subtraction based object detection, that utilizes a multiple Gaussian color model at each pixel, and object tracking based on the tracking method described in [20].

The system is first applied on each of the four camera views independently. At each frame, the 2D tracker is applied, and the resulting 2D probabilistic models are used to determine the top of the presenter’s head. These 2D object points are then considered as hypotheses of the presenter’s head top in the 3D space, based on camera calibration information and a minimum threshold for inter-ray distances between all candidate pairs in the four views. The best resulting 3D candidates at each frame are then considered to obtain the “optimal” temporal sequence / track of the presenter’s head, using a Viterbi decoding approach.

6 Experiments on the CHIL Corpus

For our experiments, we use the CHIL seminar database collected at the University of Karlsruhe (UKA). There are a total of 12 video sequences of speakers holding seminars in front of audiences inside the UKA smart room, with the first 7 seminars collected in 2003 (referred to as the Sem03 data), and the remaining 5 recorded in 2004 (Sem04 data). For every seminar, segments are allocated to the development and the test sets, each containing approximately 8,000 frames for each of the four available camera views per seminar. Ground truth labels of the 2D face center coordinates (a point of center of the head) in individual camera images are provided every 10 frames, but only when faces are visible (i.e., when the nose is visible). For the 2004 seminars, a bounding box of each visible face is also provided.

For multiview face detection, and as mentioned in Section 4, two face detectors are trained: One for the frontal view and the other for the left side view (the right side face detector is obtained by mirroring the latter). A number of frontal and left-side view face images are cropped from selected images in the development set for this purpose. In addition, non-face training samples are cropped from an image database that does not include faces. The two face detectors are trained using the FloatBoost approach described in Section 2. For the frontal face detector, a cascade structure of 15 layers and 576 features is obtained, trained on 1606 annotated images. For the left side view, the cascade structure consists of 30 layers and 4330 features, trained on 1542 annotated images.

The face detection accuracy of the combined detection and tracking scheme is depicted in Table 1. Note, that in order to obtain face detection estimates at the two non-used camera views, the face detection step is also applied to them around the 2D camera view point that corresponds to the 3D estimate of the presenter’s head. Notice that the face detection accuracy is defined as the percentage of frames that contain detected faces leading to errors between the face and the label centers less than half of the labeled face size. This is readily available for the labeled Sem04 data, but for the Sem03 data it is set to a default value of 30 pixels.

Table 1. Face detection accuracy of the proposed algorithm on the CHIL seminar data

Data	Sem03	Sem04
Face detection accuracy	76.16%	51.21%

The proposed algorithm also returns the 3D location of presenter’s head based on the estimated 2D locations in the two calibrated cameras, using triangulation. The resulting estimates are compared to the ground truth, that is available every 10 frames (66 msec), for the cases where the presenter’s head is visible by at least two cameras (i.e., has been labeled at two or more camera views, and thus can be obtained through triangulation). Four metrics are depicted in Table 2 for evaluating the performance of the system:

3D error: Mean Euclidean 3D distance in millimeters between the estimated and the ground truth position of the head center in 3D coordinates. In addition, “% err < 300” is the percentage of time instants, where the 3D error is smaller than 300 mm.

Table 2. Head location performance by the proposed algorithm versus the background subtraction system (BGS) of IBM “PeopleVision”, using the metrics discussed in Section 6

Data	Sem03		Sem04	
Method	Proposed	BGS	Proposed	BGS
3D error (mm)	253.9	278.4	467.4	480.3
3D err < 300	84.6%	81.2%	78.9%	47.7%
2D err (mm)	228.3	204.7	441.1	436.9
2D err < 300	85.3%	84.1%	80.7%	57.1%

2D error: Mean of Euclidean 2D distance in millimeters between the projection on the smart room floor of the estimated 3D head center and that of the 2D ground truth. Furthermore, “% err < 300” is the percentage of time instants, where the 2D error is smaller than 300 mm.

In addition to the proposed system, the performance of the modified background subtraction based system (BGS) developed from the IBM “PeopleVision” system is also depicted. As it becomes clear, the two approaches achieve similar results on the CHIL seminar database test sets. Notably, the performance on the Sem04 dataset is worse. This is partially due to the more challenging nature of this set in terms of lighting and motion. For example, when the presenter moves in or outside the area near the screen, the face region skin color changes abruptly due to the projector illumination on the presenter’s face. In this case, the proposed algorithm fails, due to the use of color based mean shift tracking and has to “wait” for the decision to re-initialize tracking with multi-view face detection. An additional reason for the poor performance of the proposed system on the Sem04 set is that, due to lack of time, the multiview face detectors have only been trained on Sem03 data. Nevertheless, the proposed system is slightly more consistent than the BGS approach on the Sem04 set, as it achieves a higher percentage of instants where the tracking error is less than 300 mm.

7 Summary

In this paper, we proposed a novel system for joint face detection and head tracking in camera views and the three-dimensional space using multicamera input. The system combines the strengths of FloatBoost multiview face detection and mean shift tracking, with camera calibration information that is used to initialize and verify the returned results based on two camera views. We applied the algorithm on the CHIL seminar database, and we compared the system performance to that of a background subtraction based tracker. In future work, we will extend the system to deal with four cameras, and improve tracking by seeking additional cues such as joint contour and appearance information.

References

1. CHIL project web-site: <http://chil.server.de>
2. H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, 20(1):23–28, 1998.
3. D. Roth, M.-H. Yang and N. Ahuja, "A SNoW-based face detector," In *Proc. NIPS*, 2000.
4. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," In *Proc. Conf. Computer Vision Pattern Recog.*, 2001.
5. H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," In *Proc. Conf. Computer Vision Pattern Recog.*, 2000.
6. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," In *Proc. Conf. Computer Vision Pattern Recog.*, 2000.
7. M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Int. J. Computer Vision*, 29(1):5–28, 1998.
8. J. Black and T. Ellis, "Multi camera image tracking," In *Proc. IEEE Work. on Performance Evaluation of Tracking and Surveillance*, 2001.
9. A. Hampapur, S. Pankanti, A.W. Senior, Y.-L. Tian, L. Brown, and R. Bolle, "Face cataloger: Multi-scale imaging for relating identity to location," In *Proc. IEEE Conf. Advanced Video Signal Based Surveillance*, pp. 13–20, 2003.
10. Z. Zhang, L. Zhu, and S. Li, "Real time multiview face detection," In *Proc. IEEE Int. Conf. Face Gesture Recog.*, 2002.
11. P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recog. Lett.*, 15:1119–1125, 1994.
12. J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Technical Report*, Dept. Statistics, Stanford University, Palo Alto, CA, 1998.
13. R.E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *J. Machine Learning*, 37(3):297–336, 1999.
14. A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, 19(2): 153–158, 1997.
15. P. Somol, P. Pudil, J. Novovicova, and P. Paclik, "Adaptive floating search methods in feature selection," *Pattern Recog. Lett.*, 20:1157–1163, 1999.
16. A. Bobick and J. Davis, "The representation and recognition of action using temporal templates," *IEEE Trans. Pattern Anal. Machine Intell.*, 23(3):257–267, 2001.
17. G. Welch and G. Bishop, "An introduction to the Kalman Filter," *Technical Report TR 95-041*, Computer Science Dept., Univ. of North Carolina, Chapel Hill, NC, 1995.
18. J.-Y. Bouguet, Camera calibration toolbox, http://www.vision.caltech.edu/bouguetj/calib_doc/.
19. D. Macho, J. Padrell, A. Abad, et al., "Automatic speech activity detection, source localization, and speech recognition on the CHIL seminar corpus," In *Proc. Int. Conf. Multimedia Expo*, 2005.
20. A. Senior, "Tracking with probabilistic appearance models," In *Proc. Int. Work. on Performance Evaluation of Tracking and Surveillance Systems*, 2002.