

# Robust Multi-View Multi-Camera Face Detection inside Smart Rooms Using Spatio-Temporal Dynamic Programming

Zhenqiu Zhang  
Beckman Institute  
University of Illinois  
Urbana, IL 61801, USA  
zzhang6@uiuc.edu

Gerasimos Potamianos  
Human Language Technologies  
IBM T.J. Watson Research Center  
Yorktown, NY 10598, USA  
gpotam@us.ibm.com

Ming Liu      Thomas Huang  
Beckman Institute  
University of Illinois  
Urbana, IL 61801, USA  
{mingliu1,huang}@ifp.uiuc.edu

## Abstract

*Robust face detection presents a difficult problem in real interaction scenarios, that, in order to achieve, most often requires employing additional sources of information. In this paper, we consider two such sources: Temporal information, available in the form of video sequences, and spatial information, available from multiple calibrated cameras with synchronous, overlapping fields of view of the 3D scene of interest. These two sources are exploited jointly, using a novel dynamic programming approach, for a lecture scenario inside appropriately equipped smart rooms, aiming at robust face detection of the lecturer within the available 2D camera views. Experimental results, reported on the CHIL project database, demonstrate that the proposed approach outperforms purely frame-based face detection.*

## 1 Introduction

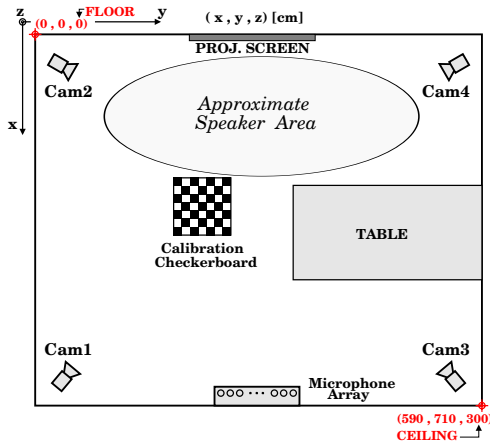
Automatic detection of human faces is a very important and challenging problem, central to human-computer interaction applications. As a result, significant research work has been devoted to it, with machine learning based approaches widely considered as the most effective. Examples of popular such techniques for face detection use neural network [11] or support vector machine (SVM) [6] classifiers, a network of linear units [10], or the AdaBoost approach [14]. These methods can be readily extended to handle detecting faces under varying head pose. For example, in [7], a view-based technique within the appearance-based framework is proposed, whereby difficulties in explicit three-dimensional modeling are avoided. In other works, Ng and Gong [5] study face trajectories in the PCA space as the face rotates, using SVMs for multi-pose face detection and pose estimation, whereas Schneiderman and Kanade [12] consider multiresolution information at different levels of

a wavelet transform, and Li et al. [4] use “FloatBoost”, an AdaBoost variant, for multi-view face detection.

In practice however, real human-computer interaction scenarios present significant challenges to most face detection algorithms, for example partially occluded and low-resolution faces, as well as lighting and head-pose variations. These difficulties can often be successfully addressed, only if additional information is available, such as temporal and spatial knowledge of the scene.

In the first case, video sequences are required that allow exploiting temporal information. While frame-based face detection makes a local decision, detecting faces across time confirms and validates the results. A number of possibilities for incorporating such information are possible, such as independent face detection at each frame, or a combination of face detection with a separate tracking module, followed with appropriate validation. Some recent examples of such approaches can be found in [13], where Verma et al. present a probabilistic framework for detecting multiple faces in a video sequence, in [3], where Han et al. detect and track multiple objects of unknown and varying number using a graph structure that maintains multiple hypotheses, and in [9], where automatic appearance models are built based on appropriate clustering over video segments.

In the second case, overlapping fields of view from multiple calibrated cameras are required, thus allowing to assess the spatial consistency of face detection results from more than one camera views. Typically, this can be achieved using traditional camera calibration techniques, and by placing a threshold on the inter-ray distance that maps two-dimensional image plane coordinates to the three-dimensional space for each combination of face detection results in the available views [1]. In practice, such a scenario is of interest when knowledge of both frame-view and space-level human location is desired (for example, person tracking in the three-dimensional space, followed by person identification).



**Figure 1.** Schematic diagram of the CHIL smart room located at a CHIL project partner site. Four fixed cameras and other sensors are depicted.

In this paper, we consider a scenario where both sources of information, temporal and spatial, are available to benefit face detection. In particular, we are interested in human-computer interaction inside smart rooms, where a speaker is presenting a seminar in front of an audience, a scenario that constitutes the focus of the European Union Integrated Project CHIL [2]. The CHIL smart room is equipped with multiple fixed cameras providing overlapping views of the lecturer (see also Fig. 1), and the aim in this paper is to robustly detect the lecturer’s face in the available camera views, whenever the front or the side of the face is visible. Under this scenario, the availability of continuous, multi-camera observations of the lecture room space allows various approaches that can improve face detection robustness. In previous work [15], we have presented one such algorithm that employs the available temporal and spatial information to improve a FloatBoost based multi-view face detector [4]. However, the information was used independently of each other, namely first in the temporal domain, by employing face tracking independently at each camera view, and subsequently in the spatial domain, by checking for consistency of the tracked faces, using the camera calibration information. Instead, in this paper, we propose to jointly exploit the two information sources, by introducing a novel approach that integrates temporal and spatial domain information within a single dynamic programming framework, as a means to improve robustness of the FloatBoost based multi-view face detector. The algorithm is applied on the CHIL database, with the experimental results clearly demonstrating a significant improvement over frame-based face detection.

The remainder of this paper is organized as follows: Section 2 briefly discusses FloatBoost, used in this work as the baseline frame-based face detection algorithm. Section 3 presents in detail the proposed dynamic programming

framework for spatio-temporal face detection. Experiments on the CHIL dataset are described in Section 4, and a brief summary is given in Section 5.

## 2 FloatBoost Multi-View Face Detection

As already discussed in the Introduction, various algorithms have been proposed in the literature for frame-based face detection. Among those, in this paper, we use the FloatBoost approach [4], similarly to our prior work in [15].

FloatBoost is a variant of AdaBoost [14], introduced to amend some of its limitations. 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 this assumption is violated. FloatBoost instead incorporates the idea of floating search [8] into AdaBoost to overcome the non-monotonicity problems associated with the latter. The sequential floating search (SFS) method [8] 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 status. As a result, quality improvement of the selected features is obtained at the cost of increased computation due to the extended search. 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.

Briefly, FloatBoost is an iterative procedure involving various steps: In the forward inclusion step, the currently most significant weak classifier is added one at a time, a step identical to AdaBoost. In the conditional exclusion step, FloatBoost removes the least significant weak classifier from the current ensemble, subject to the condition that the removal leads to a lower cost than the one incurred at the previous iteration. The classifiers following the removed one will subsequently need to be re-trained. The above steps are repeated until no more removals can be performed.

In the scenario of interest in this paper, face detection needs to accommodate the lecturer’s varying head pose, as captured in the fixed camera views inside the smart room. Therefore, a multi-view FloatBoost approach is used, where two face detectors are trained: One for nearly frontal views, and the other for the left side view, with the right side face detector obtained by mirroring the latter. Both detectors are trained by the FloatBoost technique.

## 3 Spatio-Temporal Face Detection

An example of frame-level face detection using the FloatBoost algorithm of the previous section in the scenario

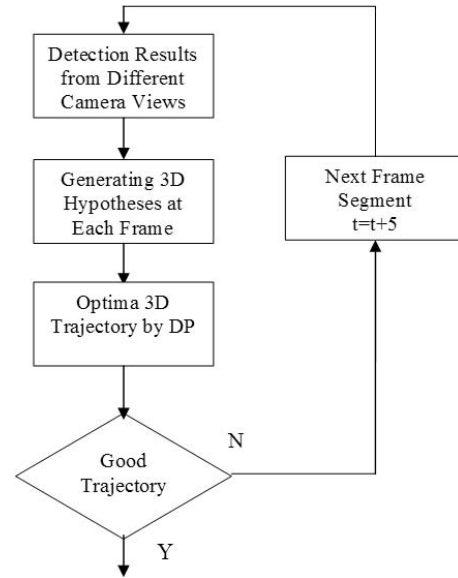


**Figure 2.** Frame-level face detection result on synchronized images from the four available camera views, using the FloatBoost frontal and side face detectors.

of interest to this paper, i.e. on the four available camera views in the CHIL smart room, is depicted in Fig. 2 (see also Fig. 1). Notice that the resolution of the lecturer's face is rather small, often less than 25 pixels in the  $640 \times 480$  pixel images. Pose and illumination variations are also severe.

Clearly, robust multi-view face detection in this scenario is really hard. We therefore seek to utilize additional information, in the form of temporal (video sequences) and spatial (multiple camera views) context. In particular, we propose a novel algorithm that integrates both temporal and spatial information from the frame-level detection results, the framework of which is schematically depicted in Fig. 3. In the proposed algorithm, multi-view face detectors are first applied on the four camera views at each time instant, over a segment of ten frames. Based on the spatial consistency of the detection result from these camera views, three-dimensional hypotheses of the presenter's face location are generated using the calibration-based camera models and triangulation. Then, *dynamic programming* (DP) is employed to search for the optimal trajectory of the lecturer's head in the three-dimensional space. Note that the algorithm can be used as an initialization component for tracking: If the optimal trajectory is accepted as a true object, based on some threshold, it can be employed to initialize lower-cost tracking; otherwise, the next 10 frames with a 5 frames overlap with the previous segment are to be examined, until an optimal trajectory is accepted based on the threshold.

Recording images using a camera is equivalent to mapping three-dimensional object space points into two-dimensional points of the image plane. It is possible to directly relate the projected image and the object by a ray through the projection center [1]. In our system, given the



**Figure 3.** Block diagram of the proposed spatio-temporal face detection system.

detection results from any two different camera views, rays are created by connecting the projection center with the detection result center at each camera view. The inter-ray distance is subsequently calculated. If there are  $n_1, n_2, n_3$  and  $n_4$  face candidates at each of the four camera views, the total number of possible three-dimensional position hypotheses at the time instant of interest will be:

$$1/2 \times \sum_{i,j=1,\dots,4} n_i \times n_j \quad (1)$$

For detection results that correspond to the same object in two camera views, the inter-ray distance should be small; otherwise, the distance should be larger, and the detection results would probably belong to different objects. Based on this assumption, we could reject some of the three-dimensional hypotheses with larger inter-ray distances. To distinguish the lecturer from other audience members, we also set a height threshold (for example, 1.5 m) for the lecturer's head center, under the assumption that the lecturer is standing, as opposed to the typically sitting audience members.

After triangulation between any two camera views, multiple local hypotheses  $h_t$  are proposed at every frame  $t$ . In our framework, the local hypothesis contains matched rectangles in two camera views and the three-dimensional position  $(x,y,z)$ , which is calculated by triangulation. Dynamic programming is then performed on these local hypotheses to find the optimal trajectory of the true target. This process contains three major components, discussed next in more detail: local similarity measurement, transition cost mea-

surement, and hypotheses space search.

### 3.1 Local Similarity Measurement

The local similarity measurement exploits the spatial information, i.e., the availability of multiple camera views for the current time instant. It is defined as the similarity between the color histograms of rectangles in the hypotheses of the different camera views. The assumption is that if the candidate hypothesis corresponds to an actual three dimensional object, then the corresponding rectangles in different camera views should have color histograms with high similarity. Such a measurement is based on Bhattacharyya coefficient, defined as

$$\rho(p, q) = \sum_{b=1}^m \sqrt{p(b)q(b)}, \quad (2)$$

where  $p$  and  $q$  are two  $m$ -bin color histograms.

### 3.2 Transition Cost

The transition cost exploits spatio-temporal information, and is used to penalize non-smooth trajectories, being defined as the three-dimensional spatial distance between consecutive hypotheses.

The transition cost is specified using Gaussian diffusion, which means that hypothesis  $h_t = (x_t, y_t, z_t)$  will be evaluated with a Gaussian  $N(h_{t-1}, \Sigma)$ . In this paper, the covariance matrix  $\Sigma$  is set to be a diagonal matrix with elements (100, 100, 100) as the square root of variances of  $x, y$  and  $z$ . With this assumption, the logarithm of the transition cost between hypotheses can be computed as:

$$\log C(h_t|h_{t-1}) = \frac{1}{2} \log |\Sigma| + \frac{3}{2} \log 2\pi + (h_t - h_{t-1})^T \Sigma^{-1} (h_t - h_{t-1}) \quad (3)$$

The transition cost for a new trajectory generation is defined as a constant equal to  $\frac{1}{2} \log |\Sigma| + \frac{3}{2} \log 2\pi$ .

### 3.3 Searching in the Hypothesis Space

The searching scheme in our framework contains two pools of hypotheses, the local hypotheses  $h_t(1), \dots, h_t(m)$  at current frame  $t$  and the active hypotheses  $H_{t-1}(1), \dots, H_{t-1}(N)$  at frame  $t-1$ .

For each local hypothesis  $h_t(i)$ , there is a local similarity measurement  $s_t(i)$  corresponding to it that specifies its likelihood based on local observation alone. For each active hypothesis  $H_{t-1}(j)$ , there exists an associated global score  $g_{t-1}(j)$  specifying its likelihood for all observations until frame  $t-1$ . The optimal transition for each  $h_t(i)$  is computed as in the following algorithm.

Given every local hypothesis  $h_t(i)$ , the optimal transition was obtained through transitions from active hypotheses  $H_{t-1}(1), \dots, H_{t-1}(N)$  and a new trajectory transition. If the transition from  $H_{t-1}(k)$  is optimal for  $h_t(i)$ ,  $H_{t-1}(k)$  is called the predecessor of  $h_t(i)$ .

---

**Algorithm 3.1:** DYNAMIC\_PROG( $H_{t-1}, N_{t-1}, h_t$ )

---

```

numH ← 0;
for i ← 1 to nt
  {
  comment: nt is the number of local hypotheses at t
  do dmax ← logC0; maxp ← 0
  comment: logC0 is the new trajectory transition cost
  for j ← 1 to Nt-1
    {
    do gs ← gt-1(j) - logC(ht(i)|Ht-1(j))
    if gs > dmax
      then dmax ← gs; maxp ← j
    numH ← numH + 1
    Ht(numH) ← ht(i)
    gt(numH) ← dmax
  }
  }

```

---

The active hypotheses  $H_{t-1}(j)$  which have not been chosen as predecessors of any local hypotheses will have a constant penalty  $P_0$ . These hypotheses together with the new  $H_t(i)$  form the active hypotheses pool at frame  $t$ . Note that missing detection at some frames is allowed in this framework. Those  $H_{t-1}(j)$  are called un-updated active hypotheses at frame  $t$ . To avoid exponential expanding of hypotheses, pruning is performed by allowing at most 6 hypotheses to be kept as active at each frame  $t$ . At the 10th frame, the global optimal trajectory is obtained by choosing the optimal  $H_t$  at frame 10. The optimal global score is subsequently compared with a fixed threshold  $T_0$ , as a means to eliminate bad trajectories. The threshold is determined empirically based on the detection and false alarm rates.

## 4 Experiments on the CHIL corpus

The meeting room considered in this paper corresponds to the smart room located at one of the CHIL project partners [2]. A number of sensors are installed in the room which include the four fixed cameras, providing the data used in this paper (see also Fig. 1). The cameras capture color data at a 640×480 pixel resolution and at 15 frames per second, are synchronized and calibrated by the calibration toolbox.

The data contain a total of 12 video sequences of speakers holding seminars in front of audiences inside the CHIL smart room. For every seminar, segments are allocated to the development and evaluation sets, each containing approximately 9,000 frames for each of the four available

**Table 1. Rejection, false alarm, and detection rates on the evaluation set of the CHIL seminar corpus. Results are computed over 10,646 time instants for three threshold values of the proposed spatio-temporal dynamic programming algorithm.**

Thresh	RT	RR(%)	FT	FR(%)	DT	DR(%)
-66	64	0.6	755	7.1	9827	92.9
-54	213	2.0	425	4.1	10008	95.9
-43	1278	12.0	93	1.0	9275	99.0

camera views per seminar. Ground truth labels of the two-dimensional 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 (defined as when the nose is visible). Three-dimensional ground truth labels are also provided, whenever two-dimensional labels are available in at least two camera views (to allow triangulation).

For multi-view face detection, two face detectors are trained on the development data: 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 [4]. 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 typical process of spatio-temporal face detection is shown in Fig. 4. Given ten consecutive frames of detection results, the top row of the figure shows the four camera view results for frames 1 and 5. We can easily see that several false alarms are returned by the FloatBoost algorithm. To reject the false alarms and obtain the true target, the proposed spatio-temporal face detection scheme is applied, and the optimal trajectory within the 10 frames is found in the three-dimensional space. The final face detection result on frames 1 and 5 is depicted in the bottom row of Fig. 4. All false alarms have been successfully deleted after the proposed spatio-temporal search.

In the 12 testing video sequences, there exist a total of 10,646 time instants, where the three-dimensional ground truth is given. Based on the baseline approach of using multi-view face detection, there are about 4 to 5 false alarms per camera view at each time instant in the raw detection result. Hence, totally, there exist about 200,000 false alarms for the instants where the three-dimensional ground truth is available. We then apply the spatio-temporal analysis

for each instant where the three-dimensional ground truth is available, placing each instant in the center of 10 consecutive time instants. Hundreds of hypotheses of presenter's 3D face location are generated at each of these 10 consecutive time instants, based on the raw detection result. An optimal trajectory of the estimated face location in 3D space is then obtained using dynamic programming. Rejection (no 3D estimation) could happen in two cases: the whole optimal trajectory is rejected, or no active 3D hypotheses are present in the center time instant. We adjust the global threshold  $T_0$  to obtain different rejection and detection rates as shown in Table 1. There, **RT** denotes the number of rejected time instants (totally 10,646 time instants with ground truth), **RR** is the rejection rate, **FT** denotes the number of time instants, when the estimated 3D face location are false alarms, **FR** is the false alarm rate (compared to the non-rejected time instants), **DT** the correctly detected time instants, and **DR** the detection rate (compared with time instants not rejected). From this table, we could observe that if we adjust the rejection rate higher (for example, 12.0%), the proposed algorithm is sufficiently robust to localize the 3D location of presenter's face and would be used as automatic initialization for subsequent tracking.

At this stage, the algorithm runs at approximately 4 frames per second on a Pentium four, 2.8 GHz desktop, with 512 MBytes of memory. However, since we know roughly the size of faces in the images, we could constrain the size of the face detector searching template from about  $20 \times 20$  to  $50 \times 50$  pixels. The resulting algorithm is then sped-up and could run at approximately 20 frames per second.

## 5 Summary

In this paper, we proposed a novel system for multi-view face detection in video sequences with input from multiple calibrated cameras. This system integrates temporal and spatial information of the detection results in a novel dynamic programming framework. We applied the algorithm on the CHIL seminar database, and our experimental results showed a clear improvement over frame-based detection.

## Acknowledgements

This work has been partially funded by the EU under contract 506909 within the project CHIL [2]. Work has been conducted during Z. Zhang's summer 2005 internship at the IBM T.J. Watson Research Center.

## References

- [1] J.-Y. Bouguet. Camera calibration toolbox. In <http://www.vision.caltech.edu>.



**Figure 4. Spatio-temporal face detection on two instants for all four camera views. Upper-row: Based on frame-level FloatBoost face detection. Lower-row: After the proposed dynamic programming.**

- [2] CHIL. Computers in the Human Interaction Loop. In <http://chil.server.de>.
- [3] M. Han, A. Sethi, and Y. Gong. A detection-based multiple object tracking method. In *Proc. Int. Conf. Image Process. (ICIP)*, pages 3065–3068, Singapore, Oct. 2004.
- [4] S. Z. Li and Z. Zhang. FloatBoost learning and statistical face detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(9):1112–1123, Sept. 2004.
- [5] J. Ng and S. Gong. Performing multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *Proc. IEEE Int. Work. on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 14–21, Corfu, Greece, Sept. 1999.
- [6] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. Conf. Computer Vision Pattern Recog. (CVPR)*, pages 130–136, San Juan, Puerto Rico, June 1997.
- [7] A. P. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. Conf. Computer Vision Pattern Recog. (CVPR)*, pages 84–91, Seattle, WA, June 1994.
- [8] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recog. Lett.*, 15:1119–1125, 1994.
- [9] D. Ramanan and D. Forsyth. Using temporal coherence to build models of animals. In *Proc. Int. Conf. Comp. Vision (ICCV)*, pages 338–346, Nice, France, Oct. 2003.
- [10] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In *Proc. Advances in Neural Information Process. Systems (NIPS)*, Denver, CO, Nov. 2000.
- [11] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(1):23–28, Jan. 1998.
- [12] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. Conf. Computer Vision Pattern Recog. (CVPR)*, Hilton Head, SC, June 2000.
- [13] R. Verma, C. Schmid, and K. Mikolajczyk. Face detection and tracking in a video by propagating detection probabilities. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(10):1215–1228, Oct. 2003.
- [14] P. Viola and M. Jones. Robust real time object detection. In *Proc. IEEE ICCV Works. on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 2001.
- [15] Z. Zhang, G. Potamianos, A. Senior, S. Chu, and T. Huang. A joint system for person tracking and face detection. In *Proc. Int. Works. Human-Computer Interaction (HCI)*, Beijing, China, Oct. 2005.