AUTOMATIC CLUSTERING OF FACES IN MEETINGS

Carlos Vallespi, Fernando De la Torre, Manuela Veloso and Takeo Kanade

Carnegie Mellon University
Robotics Institute
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

ABSTRACT

Meetings are an integral part of business life for any organization. In previous work, we have developed a physical awareness system called CAMEO (Camera Assisted Meeting Event Observer) to record and process the audio/visual information of a meeting. An important task in meeting understanding is to know who and how many people are attending the meeting. In this paper, we present an automatic approach to detect, track, and cluster people's faces in long video sequences. This is a challenging problem due to the appearance variability of people's faces (illumination, expression, pose, ...). Two main novelties are presented:

- A robust real-time adaptive subspace face tracker which combines color and appearance.
- A temporal subspace clustering algorithm.

The effectiveness and robustness of the proposed system is demonstrated over a data set of long videos (i.e. 1 hour).

Index Terms— Face Detection/Tracking, Clustering, Subspace Methods, Meeting Understanding.

1. INTRODUCTION

Meetings are an integral part of business life. In fact, approximately 11 million business meetings are held every day in the United States [1]. A mid-level manager or professional spends around 35% of his time in meetings, and this percentage increases as a person advances up the company ladder. On the other hand, meetings are not always as productive as expected. Among professionals who meet on a regular basis, 96% miss all or a part of a meeting, 73% have brought other work to the meeting, 39% have dozed during a meeting, and many of those attending a meeting need to clarify miscommunications. Having systems that help to review and share meetings can help to improve these undesirable situations. In previous work, we have proposed CAMEO (Camera Assistant Meeting Event Observer) a hardware/software system to record and process audio-visual information as a first step towards understanding human interactions in meetings [1, 2].

A very important task in meeting understanding is to know who and how many people are assisting to the meeting. Given a long video sequence of a meeting our algorithm will be able to detect, to track multiple people's faces simultaneously and to solve for the correspondence of faces across time. Tracking people's faces is a challenging task because of appearance changes due to pose, illumination, expression,... . Likewise, solving for the correspondence of people's faces is a hard problem due to failures in the tracker and variability of their appearance.

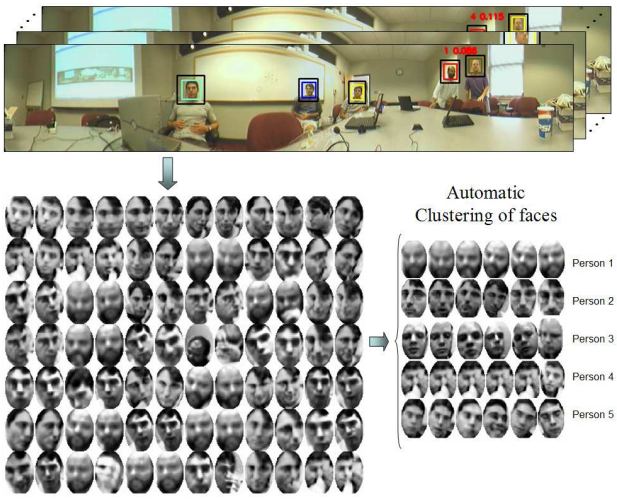

Fig. 1. Detection, tracking and automatic clustering of faces from a long video sequence.

This paper proposes two main novelties, an adaptive subspace tracker and a robust clustering algorithm able to automatically determine the number of clusters. The appearance of people's faces is modeled with a subspace. The tracker dynamically adapts the subspace to account for unseen examples. Additionally, the tracker combines appearance features with color to provide a more robust tracking against untrain changes. The tracker works in almost real-time (15 frames/sec) in 1160×260 color images. To cluster faces, we use a modification of standard spectral graph methods [1, 9] and we automatically estimate the number of people assisting to the meeting. Moreover, we define a new metric between image sequences to construct the affinity matrix.

In the experimental section, we demonstrate the effectiveness of the algorithm with data gathered from long meetings

(approx. 1 hour). We are able to effectively extract the number of people in the video sequence and to cluster them successfully. Figure 1 shows the main aim of the paper.

2. PERSON SPECIFIC TRACKER

Appearance based tracking of people's faces has been extensively used by many researchers (e.g. [6, 7, 8]). However, appearance based methods are not necessarily robust to untrain situations such as changes in expression or illumination changes. In this section, we describe two strategies to deal with untrained cases by dynamically adapt the model and by using color cues.

The appearance-based tracking algorithm has a set of bases such that a linear combination reconstruct the tracked face. These eigenbases are incrementally updated on-line (as new faces arrive) using hte incremental Singular Value Decomposition (SVD), see [3, 4] for more details. The faces are adapted if the reconstruction error is high and the face detector finds it is a face.

In order to track the face, we search for the optimal scale and translation parameters that minimize:

$$\min_{u,v} \|Im\left(s_x(x-u), s_y(y-v)\right) - UU^T Im\left(s_x(x-u), s_y(y-v)\right)\|_2^2 \quad (1)$$

where $Im(x, y)$ is a patch of the image and U contains the bases of the face model. We use a sliding window (u, v) and different scales (s_x, s_y) in a search region. After normalizing the image for its energy, Eq. 1 can be rewritten in terms of correlations operations as:

$$T(u, v, s_x, s_y) = \frac{1 - \sum_i corr(U_i, Im)^2}{corr\left(Im^2, ones\left(sp\right)\right)} \quad (2)$$

Where, U_i is a column vector of U, $corr$ is the correlation function between to images, sp is the size of the patch considered, $ones$ is a function that returns an image all 1's, and 1 is a vector all ones. The minimum value in T will denote the most likely position of the face in the image.

To achieve robustness against untrained situations, we also use color cues to help to segment the face from the background. A Gaussian model is used to dynamically model the skin color ($N(u_s, \Sigma)$, where $u_s = (R, G)$). We compute the ratio of the average probability of pixels inside the face region versus the surrounding regions, that is:

$$P(u, v, s_x, s_y) = \frac{\left(1 + \frac{1}{A}\sum_{i,j \in A} a_{i,j}\right)^2}{1 + \frac{1}{B}\sum_{i,j \in B} b_{i,j}} \quad (3)$$

Where, $a_{i,j}$ and $b_{i,j}$ are probabilities computed using the Gaussian model, A and B are the number of pixels of the inner region and the outer region respectively, u, v are the position in the image in which the equation is applied and s_x, s_y are the scale. Note the normalization term 1 , added to avoid 0's and ∞'s.

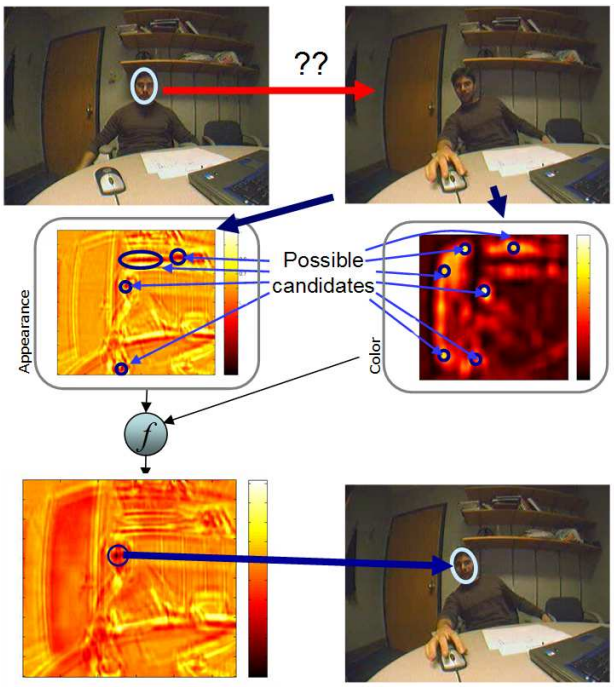

Fig. 2. Full face tracking algorithm. Desired tracker output (top left). Original image (top right) Appearance and color tracker outputs (middle). Combination of trackers (bottom left). Note that the number of candidates is reduced by combining color and appearance models. (Use color for best visualization).

Finally, the output of the color and appearance model is combined together to create the face tracking algorithm using equation 4.

$$\min_{u,v,s_x,s_y} \frac{T(u, v, s_x, s_y)}{\sqrt{P(u, v, s_x, s_y)}} \quad (4)$$

Where $T(u, v, s_x, s_y)$ and $P(u, v, s_x, s_y)$ are the outputs of the appearance tracker and the color tracker. The system uses highly optimized Intel IPP (Intel Integrated Performance Primitives) to achieve almost real time processing.

3. CLUSTERING FACES

Several trackers are instanced when a recorded video of a meeting is processed. Every instanced tracker dumps its model and tracked patches to a *folder* in the disk when is finished. In ideal conditions, there would be as many *folders* as people in the video, and one would not need this clustering step. However, in practice there are more *folders* than people, typically because the tracker gets lost (*i.e.* occlusions, a person leaves the meeting room, a person turns back, ...).

Fig. 3. Faces before the registration process (left). Faces after correcting the illumination and removing background pixels (right).

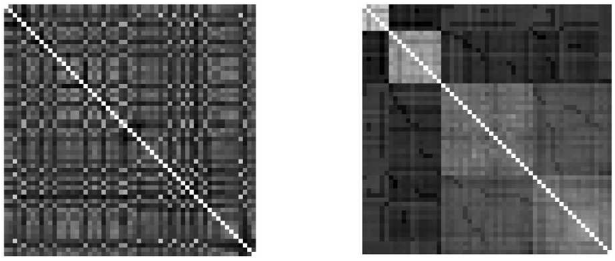

Fig. 4. Affinity matrix before and after clustering.

The algorithm Corrected Clustering with Normalized Cuts (**CCNC**), described in this section, groups together all detected faces of the same person in one cluster, and returns as many clusters as different people are in the video. It uses Normalized Cuts [9] to find the best partition of an affinity matrix W and then it refines its output.

First, in a pre-processing step, all faces are loaded, registered, equalized and normalized. Also, the background is removed using a mask. Illumination is corrected with a linear transformation of the intensity of the pixels (figure 3).

Then, we build an affinity matrix W (figure 4) which contains the similarity of all *folders* of faces. *Folders* of faces are compared using equation 5, which uses a normalized version of the reconstruction error of one *folder* of faces into other. Every face k in *folder* i is put into vector form in matrix M_i. U_i are the first eigenbases of M_i.

$$d_{i,j} = \frac{\left\|M_j - U_i U_i^T M_j\right\|_2^2}{\|M_j\|_2^2} + \frac{\left\|M_i - U_j U_j^T M_i\right\|_2^2}{\|M_i\|_2^2} \quad (5)$$

The affinity matrix W is created using equation 6:

$$w_{i,j} = e^{-\frac{d_{i,j}}{\sigma_D^2}} \quad (6)$$

where σ_D^2 is the standard deviation of $d_{i,j}$ coefficients.

Normalized cuts uses W and a number of clusters to find the best partition of W that maximizes the similarity of points belonging to the same cluster and the dissimilarity of points belonging to other clusters. To estimate the number of clusters, we start with a large number of clusters and we reduce it at every iteration if there exists overlap between clusters. The overlap value $C_{i,j}$ between cluster i and cluster j is computed by measuring how good the projection of cluster j into a model (U_i, Σ_i, $\overline{u}_i$) of cluster i is (equation 7).

$$C_{i,j} = \frac{1}{N_j} \sum_k (U_i^T (m_j^k - \overline{u}_i))^T \Sigma_i^{-1} (U_i^T (m_j^k - \overline{u}_i)) \quad (7)$$

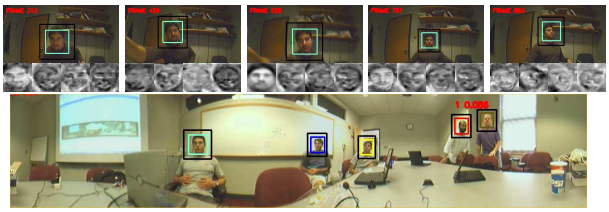

Fig. 5. The inner color box is the output of the face tracker. The black box defines where the face tracker runs.

m_j^k is the vector k of M_j and N_j is the number of images for cluster j. In order to construct the model for each cluster, outliers are rejected. A face image patch is considered an outlier if it is more than p times away from the mean: $\sqrt{\sum_{i,j} (Im(i,j) - \overline{u}(i,j))^2} < p\overline{\sigma}$, where Im is a face image, $\overline{u}$ is the mean image of a cluster, $\overline{\sigma}$ is the standard deviation image of a cluster and p is a scalar. Then, U_i are the eigenvectors of the remaining samples in the cluster i and Σ_i are the eigenvalues. The bigger $C_{i,j}$ is the bigger is the overlap. We stop reducing the number of clusters when there is no overlap between any of the clusters.

Finally, we assign every *folder* in the database to a cluster using $C_{i,j}$. Note that $C_{i,j}$ defines a score of similarity of model i to *folder* j.

4. EXPERIMENTS

We ran our system in 1 hour CAMEO videos (color images of 1160×260). Five instances of the tracker managed to ran at 15fps in a single CPU P4 at 3.06Ghz. In smaller sized videos, the tracker could handle several faces at 30 fps. Figure 5 shows the bounding box of the tracker in two videos of different size (a 320×240 sized video and a mosaic sized video). Some examples of the real time tracker can be downloaded from:

```
http://www.cs.cmu.edu/~cvalles/ICIP/Demo001.asf
http://www.cs.cmu.edu/~cvalles/ICIP/Demo002.asf
http://www.cs.cmu.edu/~cvalles/ICIP/Demo003.asf
```

To show the effectiveness of the clustering algorithm, we applied it in real data from recorded meetings. Faces were tracked during the video and recorded into folders. Later, the clustering algorithm was used to automatically estimate the number of people and to solve for correspondence between the extracted folders of faces. Table 1 shows the real number of people who attended the meeting, and the estimated one. Note that the algorithm failed in two of the videos, in one case because the sparsity of the data and in the other because one of the people went away from the working range of the camera (from 30 cm to 5 meters).

We also compare the results of the clustering algorithm with the normalized cuts (using the same metric). Table 2 shows the resulting comparison. The criteria used is the percentage of good classified faces over the total. Corrected

Video	# people	Estimated	Time	Total faces
1	3	3	18s	700
2	4	4	22s	3.400
3	5	**4**	51s	5.200
4	5	5	59s	5.400
5	5	**6**	122s	10.400

Table 1. *# people* is the number of people in the meeting. *Estimated* is the number of people estimated by our algorithm. *Time* is the time needed to converge in seconds. *Total faces* is the number of faces grabbed from the video.

	Corrected Clustering			Normalized Cuts		
	wc	nc	acc.	wc	nc	acc.
1	100%	0.00%	**100%**	100%	0%	**100%**
2	100%	4.5%	**95.5%**	94.1%	0%	94.1%
3	98.5%	2.63%	**95.9%**	91.2%	0%	91.2%
4	100%	5.1%	**94.8%**	87.5%	0%	87.5%
5	99.1%	6.2%	**93.0%**	89.2%	0%	89.2%

Table 2. *wc* and *nc* are the percentage of cases well and not classified. *acc* is the accuracy.

Clustering with Normalized Cuts (**CCNC**) performs better than Normalized Cuts in the videos tested, by getting a better overall accuracy. Furthermore, the reliability is increased by getting in all cases almost **0%** of error.

5. SUMMARY AND CONCLUSIONS

In this paper, we have proposed a robust algorithm to automatically extract people's faces in long video sequences and cluster them. The presented face tracking algorithm handles video sequences of recorded meetings in real time and is robust to moderate changes in the pose (from frontal to profile faces), illumination and rotation (up to 30 degrees). Its output enhances the performance of the clustering algorithm by imposing timing constraints (it is not possible to see the same face in two different locations at the same time -assuming there are no mirrors in the room).

CCNC compares favorably to Normalized Cuts algorithm for clustering of people's faces since it removes outliers and gives a good automatic estimate of the number of clusters.

In the future, we plan to improve the tracking algorithm by using the best model from a pre-stored database of models every time it is instanced, instead of initializing every new instance of the tracker to the mean face model. Furthermore, with a weighting mask this tracker would be more robust to partial occlusions. **CCNC** is not real time due to the iterative algorithm to estimate the number of clusters. Using the information provided by the models of the tracker it is possible to start with a better estimate and reduce the number of iterations needed to converge.

Acknowledgments

Thanks to Paul Rybski for helpful comments and discussions. This work has been partially supported by the National Business Center under contract no. NBCHD030010 and SRI International under subcontract no. 03-000211. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

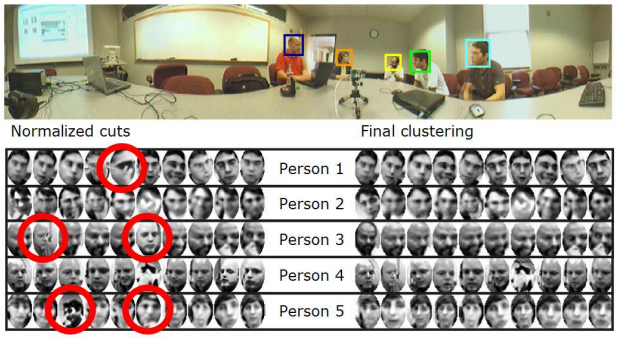

Fig. 6. Normalized Cuts (bottom left). Outliers corrected with CCNC (bottom right).

6. REFERENCES

[1] F. De la torre, C. Vallespi, P.E. Rybski, M. Veloso, T. Kanade. Omnidirectional Video Capturing, Multiple People Tracking and Identification for Meeting Monitoring *Tech. report CMU-RI-TR-05-04, RI, CMU, January, 2005.*

[2] P.E. Rybski, F. De la Torre Frade, R. Patil, C. Vallespi, M. Veloso, and B. Browning. CAMEO: Camera Assisted Meeting Event Observer. *Tech. report CMU-RI-TR-04-07, RI, CMU, January, 2004.*

[3] M. Brand. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. *ECCV 2002.*

[4] D. Ross, J. Lim, M.H. Yang. Adaptive Probabilistic Visual Tracking with Incremental Subspace Update. *ECCV04(Vol II: 470-482)*

[5] M.J. Black, A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using view-based representation. In Buxton, B., Cipolla, R. *Proceedings of the Fourth European Conference on Computer Vision. LNCS 1064, Springer Verlag (1996) 329-342.*

[6] T. F. Cootes and G. J. Edwards and C. J. Taylor. Active Appearance Models. *Lecture Notes in Computer Science, volume 1407, pages 484–??, year 1998.*

[7] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience, volume 3, pages 71-86, year 1991.*

[8] B. Champagne, Q.G. Liu. Plane rotation-based EVD updating schemes for efficient subspaces tracking. *IEEE Transactions on Signal Processing 46 (1998) 1886-1900.*

[9] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 888-905, August 2000.*