Omnidirectional Video Capturing, Multiple People Tracking and Identification for Meeting Monitoring.

Fernando De la Torre Carlos Vallespi Paul E. Rybski Manuela Veloso Takeo Kanade

CMU-RI-TR-05-04

January 2005

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Meetings are a very important part of every days life for professionals working in universities, companies or governmental institutions. In fact, it is estimated that a mid-level manager or professional spends around 35% of his/her time in meetings. We have designed a physical awareness system called CAMEO (Camera Assisted Meeting Event Observer), a hardware/software component to record and monitor people's activities in meetings.

CAMEO captures the audio and a high resolution omnidirectional view of the meeting by stitching images coming from almost concentric cameras. Besides recording capabilities, CAMEO automatically detects people and automatically learns a person-specific facial appearance model (PSFAM) for each of the participants. The PSFAMs allow more robust, reliable and faster tracking and identification. Several novelties in the video capturing device, multiple person identification and tracking are proposed. The effectiveness and robustness of the proposed system is demonstrated over several real time experiments and a large data set of videos.

Contents

1 Introduction **1**

2 An omnidirectional view of the meeting **2**
2.1 Real-time mosaicing 2
2.2 Geometric and photometric autocalibration 4
2.3 Software specifications 6

3 Multiple People Tracking **7**
3.1 Detecting regions of interest. 7
3.2 Learning person-specific facial appearance models (PSFAM) 7
3.2.1 Off-line learning 8
3.2.2 On-line learning and adaptation 9
3.3 Efficient subspace tracking 10
3.4 Solving for correspondence 11

4 Multiple Face Recognition **12**
4.1 Preprocessing 13
4.2 Multimodal Oriented Discriminant Analysis 14
4.3 Improving performance 16
4.3.1 Pattern rejection and verification 16
4.3.2 Temporal consistency 18
4.3.3 Multiple face recognition 19

5 Experiments **20**
5.1 Depth estimation 20
5.2 Multiple face recognition 20
5.3 Multiple people tracking 21
5.4 On-line learning and adaptation 21
5.5 Learning PSFAM from long video sequences 21

6 Discussion and future work **23**

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 [2, 17]. 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 [17]. 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 fact, many companies now use devices to transcribe events (such as who spoke and what was discussed) into a digital form that can be searched and analyzed. This could be a very labor intensive and a tedious task for a human to do. Using new technologies such as LCD projectors, optical character recognition, voice-to-text converters, cameras or interactive whiteboards will allow for the sharing of data and the automatically saving of information. This is a preliminary step toward implementing collaborative technology in the meeting room.

In this paper we develop CAMEO (Camera Assistant Meeting Event Observer) a hardware/software system that is able to record and process audio-visual information as a first step towards understanding human interactions in meetings. CAMEO is part of a larger effort to develop an enduring personalized cognitive assistant that is capable of helping humans handle the many daily business or personal events that they engage in. This larger project, CALO (Cognitive Agent that Learns and Organizes) aims to build a personalized computational resource that will be able to handle routine tasks/events, anticipate predictable user needs and prepare for them appropriately, and assist the user in handling unexpected events. A CAMEO device is brought into a meeting and simply placed in the center of the room without requiring special manual calibration or instrumented rooms. Apart from omnidirectional audio-video recording capabilities, CAMEO will be able to determine (real time) who is in the meeting, the placement of each of the participants, and when some events occur. This is a first step towards understanding human activity in meetings. Figure 1 shows the hardware and a block diagram of the software capabilities of CAMEO.

The CAMEO hardware is composed of 4 daisy-chained web-cameras and an omnidirectional microphone, see fig. 1. There are two main benefits in the hardware design, a 360 degrees of field of view and a portable device, so there is no need for instrumented rooms to record meetings. A mosaic is constructed by stitching the images coming from the four cameras. Once the mosaic is built, CAMEO is able to track and identify multiple people in the omnidirectional video in real time. The tracker and recognition systems are effective and robust because they are based on a set of learned person-specific facial appearance models (PSFAM). The PSFAM can be learned on-line or off-line, and the model is usually adaptive to compensate for environmental changes. In this paper several novelties in the hardware design, tracking and recognition are introduced.

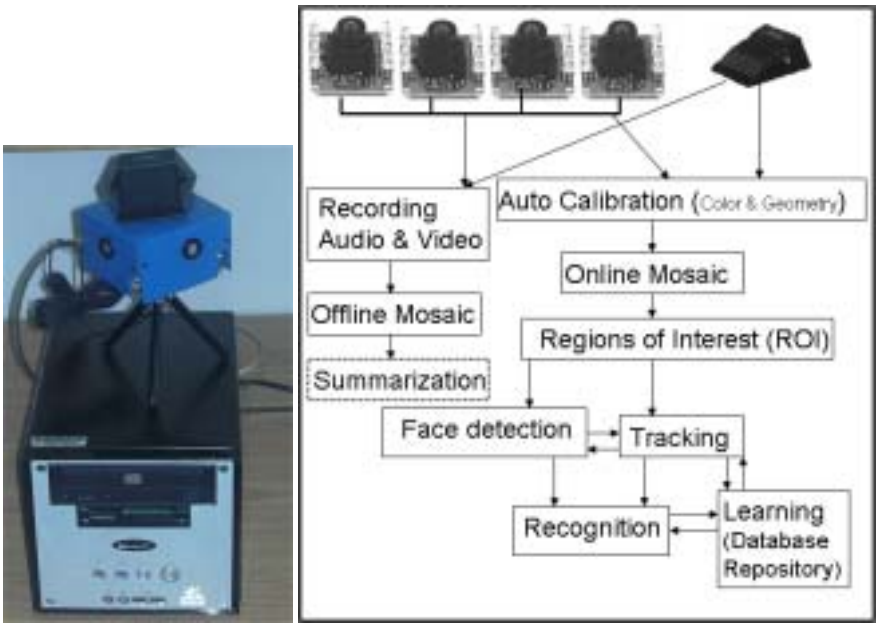

Figure 1: CAMEO a) Hardware b) Software

2 An omnidirectional view of the meeting

Meeting understanding has been an active research topic during the last few years and several groups have proposed intelligent rooms [15, 18], concentric cameras [3, 6, 29] and all sort of devices [21] to record human activity in meetings. Instead of having instrumented meeting rooms, CAMEO is intended to be a portable and self-calibrating device. In order to have a compact representation of the meeting, CAMEO records a panoramic view of the meeting room. Many techniques have been researched to construct panoramic images from real-world scenes. Mirrored pyramids and parabolic mirrors [23] could be used to capture the images directly; however, in order to capture high resolution images expensive equipment is needed and several defocusing problems may result in low quality video. Similar to previous work [3, 6, 29], CAMEO will integrate images coming from almost concentric images into a single mosaic. In this section the software/hardware details for constructing the camera device are explained. Preliminary results have been presented at [25].

2.1 Real-time mosaicing

CAMEO is composed of four inexpensive web cameras that have been daisy-chained and just one firewire cable is necessary to transmit the signal and power, similar to [3, 6]. In order to reduce the number of cameras, wide angle lenses with $1.7mm$ of focal length and approximately $110^{\mathbf{o}}$ of field of view are used. This guarantees a slightly overlap between the field of view of two cameras (further than 30 cm). A small focal

Figure 2: a) Original Image b) Corrected image. Straight lines in the world are map to straight lines in the image.

length yields large depth of field and thus all objects are in focus from a distance of few centimeters to infinity, eliminating the need for autofocus. However, these lenses introduce big radial and tangential distortion in the images. The first step towards stitching the images is to compute an estimate of the intrinsic camera parameters, computed with a standard calibration toolbox ($http://www.vision.caltech.edu/bouguetj/calibk\_doc/$). The intrinsic parameters include effective focal length $f_x,\ f_y$, the image center or principal point $x_o,\ y_o$, and the ones to correct the radial/tangential distortion $k_1,\ k_2,\ k_3,\ k_4$[1]. The projection model, taking into account the distortion model, has the following expression [10]:

$$
\begin{aligned}
x_n = \tfrac{X}{Z} \quad y_n = \tfrac{Y}{Z} \quad r^2 = x_n^2 + y_n^2 \\
u_p = (1 + k_1 r^2 + k_2 r^4) x_n + 2k_3 x_n y_n + k_4 (r^2 + 2x_n^2) \\
v_p = (1 + k_1 r^2 + k_2 r^4) y_n + 2k_3 x_n y_n + k_4 (r^2 + 2y_n^2) \\
x_p = f_x u_p + x_o \quad y_p = f_y v_p + y_o
\end{aligned}
$$

where X, Y, Z are the 3D coordinates and x_p, y_p are the pixels positions in the image. Figure (2.a) shows the image before and after (2.b) correcting for radial/tangential distortion. As we can see in figure (2.b) after the radial/tangential distortion is corrected, straight lines into the world will be mapped to straight lines into the image.

Once the radial/tangential distortions have been corrected, several techniques exist for obtaining mosaic images coming from several cameras [19, 3, 6, 30, 20]. However, most of them assume that the camera is panning or that only rotation exists between cameras' optical center. When the motion between the optical centers of the cameras is just rotational, it is easy to show that a homography can relate the geometry of the images [9]. In our case, the cameras do not share a common center of projection (see fig. 1), and parallax effects occur due to the translational component between the optical centers of the cameras. Having translational motion between cameras' optical centers, the geometric transformation that relates two images becomes depth dependent

[1]Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d_j}$ represents the j column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{D}$ and the scalar i-th element of a column vector $\mathbf{d}_j$. All non-bold letters will represent variables of scalar nature. $diag$ is an operator which transforms a vector to a diagonal matrix. $\mathbf{I}_k \in \Re^{k \times k}$ is the identity matrix. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$. $||\mathbf{A}||_F = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of a matrix. $N_d(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a d-dimensional Gaussian on the variable $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. $\odot$ Hadamard (point wise) product.

Figure 3: a) Original images. b) Mosaic image.

(the parallax effect becomes more evident at shorter distances). One possible solution will involve computing depth for each point [19]; however, this approach will be very expensive for real-time applications. With the topology of the camera, if the objects are approximately 2 m far away from the camera, parallax effects can be ignored. To minimize this effect and because of easy construction, cylindrical panoramas are commonly used [3, 29, 30]. Each image is corrected and warped into cylindrical coordinates ($\theta = atan(\frac{X}{Z}), v = \frac{Y}{\sqrt{(X^2+Z^2)}}$. In order to speed up the process, a look up table (LUT) is constructed to correct the radial/tangential distortion and the cylindrical mapping. Once the images coming from the cameras are corrected and warped into cylindrical coordinates, constructing the mosaic is a translational estimation problem (assuming almost concentric cameras). CAMEO automatically searches for the translation that produces the best match between adjacent cameras. A constrained (by the epipolar geometry between the cameras) normalized template matching is computed to search for the optimal translation. Despite the possibility of using gradient descent type of methods [30], parallax effects and the large change in viewpoint make them too sensitive to local minima. Finally, a weighted (more weight to the image that is closer) blending procedure is used to merge both images. Figure 3.a shows four original images and how they are merged into the mosaic one 3.b.

2.2 Geometric and photometric autocalibration

When CAMEO starts, it loads all the camera parameters, LUTs, and begins the geometric/photometric calibration process. Because the mosaic is constructed from different cameras, all of them should be similarly color-calibrated to ensure that the cameras look alike. In the beginning one camera is taken as the reference camera and the chromatic characteristics are recorded and propagated to the other cameras (no automatic settings are used). In order to adjust different lighting and CCD properties, an affine transformation is computed using overlapping regions. That is, given a set of matching points between two images, an affine transformation $(\mathbf{A}, \mathbf{b})$, which minimizes the error among matched points $\min_{\mathbf{A},\mathbf{b}} \sum_i ||\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2 + \mathbf{b}||$, is computed. Finally, the camera drivers are used to change the hue and saturation values of each camera and correct the color in a hardware efficient way.

A useful feature of CAMEO is to know the relative position of a person/pattern with respect to one of the cameras. This will allow us to calibrate between several CAMEO's, to estimate the depth of a person, and to know the position with respect to other devices. Having the internal camera parameters, and assuming a planar calibration pattern, it is relatively simple to estimate the relative orientation with respect to the camera (e.g [35]). In order to simplify the scenario, it is assumed that the planar pattern has just two rotational degrees of freedom, one angle θ, which describes the in-plane rotation, and γ for the tilt. The pattern is composed of 3 colors (see fig. 4.a). CAMEO automatically detects it with a high resolution image using normalized template matching at different scales. Without loss of generality, it is assumed that the left corner of the pattern is the world coordinate system and the axes are aligned with the pattern (the pattern is in the plane Z=0). Under these assumptions, recovering the rotational angles and translational components $\theta, \gamma,\ t_x,\ t_y,\ t_z$, is achieved by minimizing:

$$\begin{aligned} E(t_x, t_y, t_z, \theta, \gamma) &= \textstyle\sum_{i=1}^{N} ((x_p^i\ y_p^i)^T - \mathbf{P}\big(\mathbf{R}_2(\theta)\mathbf{R}_1(\gamma)(X_i,\ Y_i,\ Z_i)^T + (t_x\ t_y\ t_z)^T)\big)^2 \\ &= \textstyle\sum_{i=1}^{N} (x_n^i - \frac{X_i cos(\theta)cos(\gamma) - Y_i sin(\theta) + tx}{X_i sin(\gamma) + t_z})^2 \\ &\quad + (y_n^i - \frac{X_i \sin(\theta)cos(\gamma) + Y_i cos(\theta) + ty}{X_i sin(\gamma) + t_z})^2 \end{aligned} \tag{1}$$

where $(x_p^i,\ y_p^i)$ are the pixel coordinates of the pattern for the point i and $(X_i,\ Y_i,\ Z_i)$ are the 3D coordinates in the global reference frame (the pattern one). $\mathbf{R}_1(\gamma) = \begin{pmatrix} cos(\gamma) & 0 & -sin(\gamma) \\ 0 & 1 & 0 \\ sin(\gamma) & 0 & cos(\gamma) \end{pmatrix}$, $\mathbf{R}_2(\theta) = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and $\mathbf{P}$ is a non-linear projection operator that takes into account the internal camera parameters (radial distortion, focal length, principal point). Optimizing eq. 1 involves a non-linear optimization and may be difficult to solve due to multiple local minima. Rather than applying gradient descent type of methods starting from different initial points, a multigrid search is followed. There are two sources of non-linearity in eq. 1, one due to the angles and the other due to the quotient (easily solved by multiplying). The angle space is sampled for $\theta \in [0..2\pi]$ and $\gamma \in [0..2\pi]$, and for each value of θ, γ the following linear system of equations is solved:

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} -1 & 0 & x_n^1 \\ \cdots & & \\ -1 & 0 & x_n^N \\ 0 & -1 & y_n^1 \\ \cdots & & \\ 0 & -1 & y_n^N \end{bmatrix}^{\dagger} \begin{bmatrix} X_1 cos(\theta)cos(\gamma) - Y_1 sin(\theta) - x_n^1(X_1 sin(\gamma)) \\ \cdots \\ X_N cos(\theta)cos(\gamma) - Y_N sin(\theta) - x_n^N(X_N sin(\gamma)) \\ X_1 sin(\theta)cos(\gamma) + Y_1 cos(\theta) - y_n^1(X_1 sin(\gamma)) \\ \cdots \\ X_N sin(\theta)cos(\gamma) + Y_N cos(\theta) - y_n^N(X_N sin(\gamma)) \end{bmatrix} \tag{2}$$

$\dagger$ indicates the pseudo-inverse, which is computed just once. In order to make the search efficient, first the minimum of a $10^{\mathbf{0}}$ grid is searched and later one of $1^{\mathbf{0}}$ is computed. Figure 4.b shows the value of the energy function for several values of θ and γ. In this particular case, there are two valid solutions with the same energy value (due to planar ambiguity). The parameters which give positive depth are chosen.

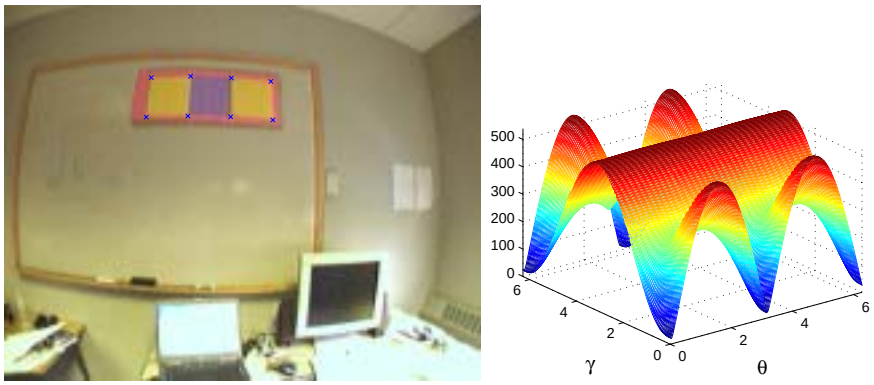

Figure 4: a) Calibration pattern. b) Surface error.

2.3 Software specifications

In order to ensure software stability, the software is divided into 4 main modules:

- **Video acquisition**: Acquires raw data from four cameras using Microsoft Direct Show. This module supports different resolutions and different frame rates.

- **Mosaic generation**: Builds a mosaic from 4 camera streams. This module is optimized using IPP Intel libraries, and has two sub-modules: a. Correction of radial distortion and cylindrical mapping: LUT and interpolation; b. Mosaic calibration/tuning: Computes the translational error between overlapping areas of adjacent cameras, keeps track of the overlapping error and recomputes the translation if needed.

- **Recording system (optional)**: Microsoft Windows Media 9 Series (MWM) is used to record the output of the acquired mosaic. MWM provides a set of features that are very convenient such as: audio and video synchronization, real time compression, streaming, and the possibility of adding metadata in the stream.

- **Processing module**: Uses the remaining CPU processing time to detect, track and recognize faces. To make this possible, we first need to determine the amount of CPU time remaining for processing each image. Then we assign quotes of this time to each sub-module to ensure real-time processing. If the system runs out of time, it then jumps to the next task or module. Most of the routines use OpenCV functions, that are highly optimized for INTEL processors.

The bandwidth of the Fire-Wire bus is up to 400 Mbps and to reduce the amount of data transmitted CAMEO acquires the images using YUV format. However, there are some limitations due to the Fire-Wire bandwidth, and there exists a trade-off between resolution and the frame rate. Table 2.3 shows the bandwidth and CPU times required for each configuration to build the mosaic. Finally, each meeting takes about 0.5G/hour to store (high quality video), most of it the video signal (also the compression is proportional to the number of people/movement).

Cameras	Resolution	Frame rate	Bandwidth	CPU time
4	320x240	30	221 Mbps	50%
4	320x240	15	111 Mbps	25%
4	320x240	7.5	56 Mbps	15%
4	640x480	7.5	277 Mbps	45%
4	640x480	3.75	138 Mbps	25%

Table 1: Measured with Pentium-M CPU at 1.7 Ghz and 1 Gbyte of RAM.

3 Multiple People Tracking

Real time robust localization and tracking of faces from the omnidirectional video is a key aspect of CAMEO towards understanding human activity. Knowing people's position is helpful to extract high level information in order to infer activity [24]. However, tracking multiple people is a challenging problem due to significant occlusion caused by interaction among people, deep changes in pose, and fast motions. Moreover, in the CAMEO scenario low quality video, low contrast and varying illumination conditions difficult the tracking process. This section describes the use of person-specific facial appearance models (PSFAM) for tracking multiple people.

3.1 Detecting regions of interest.

In order to detect/track people efficiently, a first step is to detect regions where potentially there can be moving objects. The regions of interest (ROI) are computed using a simple but effective background subtraction algorithm.

When CAMEO starts, it computes an estimate of the background by averaging the incoming mosaic images during several seconds (no camera motion). Once an estimation of the background is given, a multi-resolution version of the mosaic image from the estimated background is subtracted; this will provide a first estimate of the ROI. The difference is thresholded and an opening (morphological operator) with an structural element of 3×3 is used to eliminate spurious noise. At this point, a binary image with blobs is created. The blobs correspond to compact regions with graylevel changes, and the previously detected/tracked faces' area is added to enforce temporal consistency (in case the person does not move the head). In order to label the blobs, the image is projected into its x and y coordinates. From segmenting the x-projection, it is easy to estimate the y component and build a bounding box around the area of interest. Finally, the background regions that do not belong to the ROI are used to update the estimation of the background. Figure 5 illustrates the ROI algorithm.

3.2 Learning person-specific facial appearance models (PSFAM)

Since most of the people remain seated during the meeting, we have focused our efforts on developing head trackers that are able to track the head from profile to profile. On the other hand, most of the people who assist at the meetings are the same. Taking this fact into account, CAMEO will automatically learn (off-line and on-line) Person-

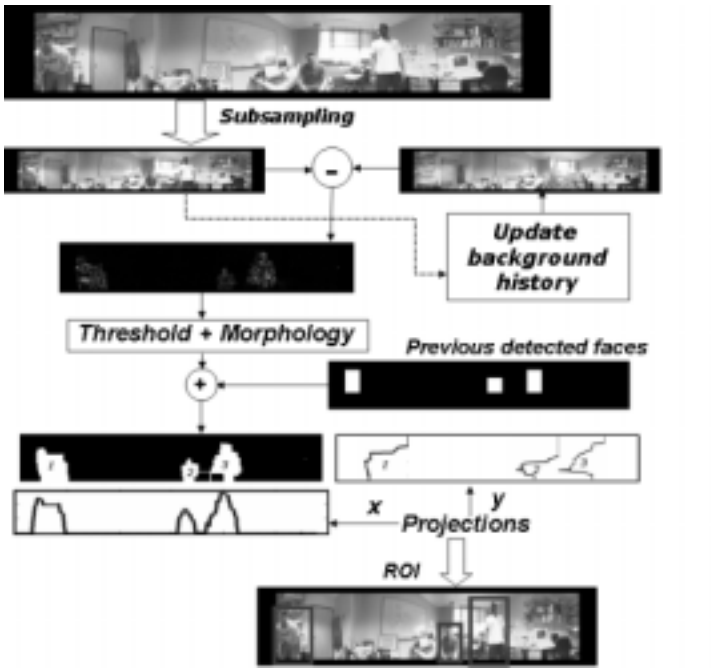

Figure 5: Algorithm to detect regions of interest.

Specific Facial Appearance Models (PSFAM), which will allow more robust and faster tracking. Given a new video, CAMEO will automatically detect the people and identify them (see next section). If the person is recognized, CAMEO will use his/her person-specific facial appearance model to track his/her head. The PSFAM will be adapted on-line to take into account changes in appearance and environmental conditions. If the person is not recognized, CAMEO will build the PSFAM on-line.

3.2.1 Off-line learning

In order to construct a statistical model of facial-appearance variation of a person, a person sits in front of a CAMEO's camera and he/she performs different facial expressions under several pose/illumination changes (approximately 1 minute of video is recorded). After the ROI's are computed, the Scheniderman face detector [26, 27] is used to detect frontal/profile faces. Figure (8.a) shows some original frontal faces gathered (60×60 pixels); approximately around 800 frontal faces are gathered.

The face detector occasionally gives some false positives. In order to filter these potential outliers, a simple scheme based on color [33] is used. Using the normalized r, g components ($r = \frac{R}{R+G+B}$, $g = \frac{G}{R+G+B}$), a Gaussian distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is fitted to a set of training faces, where $\boldsymbol{\mu} \in \Re^{2\times 1}$ and $\boldsymbol{\Sigma} \in \Re^{2\times 2}$ are the mean and covariance that approximate the skin color. Given a new face, CAMEO computes the percentage of skin color pixels inside the patch containing the face, and the percentage

Figure 6: a) Set of templates. b) First eigenbasis

of skin color pixels in the surrounding area. In an ideal face, the ratio of skin pixels in the area containing the face versus the percentage of skin pixels in the surrounding area should be big. If this ratio is below a certain threshold the sample face is discarded.

A possible way of constructing a PSFAM will consist of selecting several prototypes (different scales and profiles). To track, a normalized template matching would be computed for each prototype and the one with minimum error would be selected. However, as the number of templates increases, it becomes impractical to find the best match with respect to each of the templates, and in our scenario a more efficient and robust matching approach is necessary. To exploit the spatial redundancy existing between the templates, to filter noisy data and to average clutter from the background, a subspace $\mathbf{B}^i$ for subject i is computed by means of the Principal Component Analysis (PCA) [8]. In order to get a better estimate of the subspace, the images have to be perfectly geometrically aligned with respect to the subspace; parameterized component analysis [4] is used to achieve geometric (translation, rotation, scale) invariant learning. After the data have been registered with respect to the subspace, which preserves 85% of the energy, the data is clustered in approximately 150 prototypes in order to avoid the principal components being biased towards specific facial expressions/poses that are more common. Furthermore, if the number of profile faces is lower than the frontal ones, some samples are duplicated in order to give the same weight to all the facial expressions. In figure 6.(b), the set of eigentemplates at one scale are displayed after applying parameterized component analysis. Usually three eigentemplates are constructed at 3 different scales by subsampling the training data.

3.2.2 On-line learning and adaptation

If CAMEO does not recognize the person during execution time, it will gather faces coming from the face detector in order to build a PSFAM. With a minimum of 8 faces, CAMEO will start building the PSFAM recursively. By recursively adapting the PSFAM, the tracker will be much more robust to illumination and appearance changes. The core of the algorithm is based on the recursive SVD [1, 22].

Let $\mathbf{D} \in \Re^{d \times n}$ be a matrix with some initial data (e.g. 8 images) and its SVD represented by $\mathbf{D} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $k = \min\{d, n\}$, $\mathbf{U} \in \Re^{d \times k}$ and $\mathbf{V} \in \Re^{k \times n}$ are an orthogonal basis that spans the column and row space of $\mathbf{D}$. $\mathbf{\Sigma} \in \Re^{k \times k}$ is a diagonal matrix containing the singular values. If a new set of samples $\mathbf{P}$ are available, computing the SVD of $[\mathbf{D}\ \mathbf{P}]$ can be done recursively [1]. $\mathbf{P}$ can be expressed as $\mathbf{P} = (1 - \mathbf{U}\mathbf{U}^T + \mathbf{U}\mathbf{U}^T)\mathbf{P} = \mathbf{P} - \mathbf{U}\mathbf{U}^T\mathbf{P} + \mathbf{U}\mathbf{U}^T\mathbf{P} = \mathbf{E} + \mathbf{U}\mathbf{L}$, where $\mathbf{L} =$

$\mathbf{U}^T\mathbf{P}$ and $\mathbf{E} = (1 - \mathbf{U}\mathbf{U}^T)\mathbf{P}$. Using the QR factorization, $\mathbf{E} = \mathbf{J}\mathbf{K}$, the new data $\mathbf{P}$ can be factorized into a reconstruction with the previous basis $\mathbf{U}$ and its orthogonal complement. The previous equation can be rearranged into:

$$\begin{bmatrix} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{U}\mathbf{\Sigma} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T = \begin{bmatrix} \mathbf{U} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{L} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^T \quad (3)$$

Eq. 3 has much of the desired factorization, except that the central matrix is not diagonal. However, it can be efficiently factorized with the SVD. In the particular case that $\mathbf{P}$ is just a vector, the previous expression can simplified, see [1] for the details.

Registration of new images with respect to the model is a key aspect to ensure compactness and achieve good generalization. In the off-line version, parameterized component analysis [4] has been used; however, it can be computationally expensive to run on real time. Instead, the gathered data is registered (just translation) with respect to a set of clusters using normalized correlation. Later, the clusters are recomputed and the procedure is repeated until convergence. This procedure is not as optimal as [4], but it is suitable for fast on-line registration. In order to select the number of clusters, a measure of the compactness of the data after the registration step is applied. The compactness is measured by computing the energy of a set of bases when the SVD is performed. Table 3.2.2 shows results for different number of clusters; the first row indicates the number of bases, and the second row and beyond the percentage of energy preserved. We compare the percentage of energy rather than the total reconstruction error, since due to interpolation errors the energy of two sets with different registration parameters is not necessarily equal. As it can be seen from the results, registering with

Number of basis 1	2	3	4	5	6
1 cluster(mean)	0.3	0.44	0.45	0.48	0.6
2 clusters	0.31	0.32	0.39	0.59	0.63
4 clusters	0.31	0.32	0.39	0.59	0.63
6 clusters	0.31	0.32	0.39	0.59	0.63
8 clusters	0.31	0.32	0.39	0.59	0.63

Table 2: Energy vs. number of basis for different number of clusters.

respect to two clusters usually results in a good trade-off between speed and quality of the clustering.

3.3 Efficient subspace tracking

Once the person-specific subspace $\mathbf{B}$ is estimated, the problem becomes how to track the face, that is, finding the scale, position and appearance coefficients in the image that best match the model. Given a subspace $\mathbf{B}$ and an image $\mathbf{I}$, CAMEO has to find the position (u, v) in the image $\mathbf{I}$ such that the distance from the subspace is minimum. At a given scale this implies minimizing:

$$E(u, v, \mathbf{c}) = \min_{u,v,\mathbf{c}} ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v) - \mathbf{B}\mathbf{c}||_2^2 \quad (4)$$

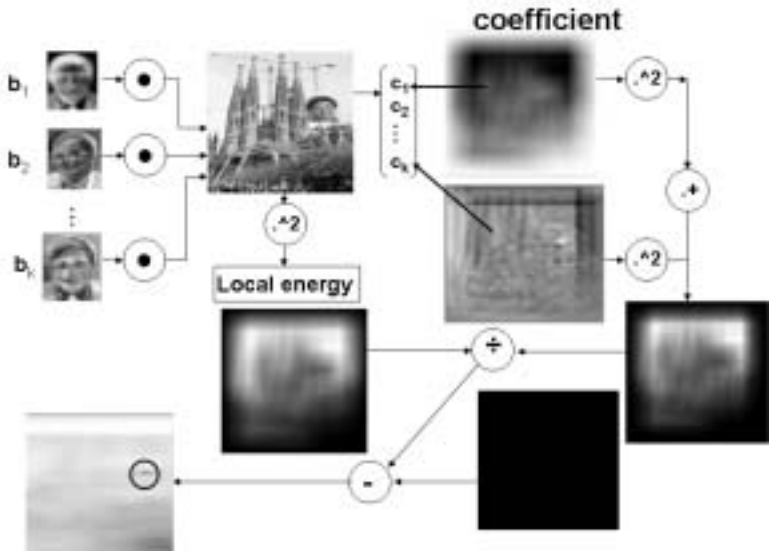

Figure 7: A subspace is constructed with the 3 program chairs of ICRA-2005. One of the program chairs' picture is included next to the Sagrada Familia in Barcelona. After computing the distance from the subspace, the minimum is located where the face is.

where $\mathbf{x}, \mathbf{y}$ are the spatial coordinates of a rectangle of the same size of the subspace images, and u, v are the position of the head to search for. An obvious approach is to compute the reconstruction error for each position (u, v); however, this approach is not efficient either in space or time. Key to developing efficient methods is to observe that the error at a particular position (u, v) can be computed as $E(\mathbf{c}, u, v) = ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v) - \mathbf{B}\mathbf{c}||_2^2 = ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)||_2^2 - \mathbf{c}^T\mathbf{c}$ [16], where $\mathbf{c} = \mathbf{B}^T\mathbf{I}(\mathbf{x}+u, \mathbf{y}+v)$. Computing the coefficients $\mathbf{c}$ is equivalent to correlating the image with each basis of the subspace and stacking all the values for each pixel. For large regions, this correlation is performed very efficiently in the frequency domain with the Fast Fourier Transform (FFT) (i.e. $c_1 = \mathbf{b}_1^T\mathbf{I} = IFFT(FFT(\mathbf{b}_1) \odot FFT(\mathbf{I}))$), and for small regions the highly highly optimized correlation from OpenCV is used. Finally, the local energy term, $||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)||_2^2$, is computed very efficiently using the integral image [14]. Figure 7 shows an illustration of the subspace correlation method.

3.4 Solving for correspondence

The tracker may get lost either due to abrupt appearance changes or when people cross/occlude one another. Having PSFAM greatly simplifies the data association problem. In the on-line version of the tracker, CAMEO waits until it finds a face by means of the face detection [26, 27] and tracks it using normalized correlation. Later, the face recognition is used to know the identity of the person and acquire his/her PSFAM.

A more interesting situation occurs when learning PSFAM from long video se-

quences. That is, given a video sequence, CAMEO will track and gather all the faces as training data for face recognition. In long videos (e.g. 1 hour), it is likely that the trackers eventually get lost. In this section, we introduce a simple but effective manner of solving the correspondence between several sets of tracked faces.

The correspondence problem is posed as a clustering one. In particular, we make use of recent advances in spectral graph methods [34]. Given several sets of images tracked over different periods of time, an affinity matrix is computed as follows. First the principal components for each set of images (many containing the same person at different time instances) is computed. Given 2 sets of images and its principal components $\mathbf{B}^1, \mathbf{B}^2$, several distance measures between sets of data are explored: principal angles [8], average cosine angles [12] and the average reconstruction error.

The principal angle [8] is the minimum angle between the columns of $\mathbf{B}^1, \mathbf{B}^2$ and can be used as a measure of distance between two subspaces. However, this measure can be very sensitive to outliers. The average cosine angles [12] is a more robust measure and it is computed as follows; one of the subspaces is considered to be "fixed" and we find the best-matching set of orthogonal axes in the second subspace. It can be shown that this is equivalent to compute [12]:

$$\mathbf{S} = (\mathbf{B}^1)^T\mathbf{B}^2(\mathbf{B}^2)^T\mathbf{B}^1 \quad 0 < d(\mathbf{B}^1, \mathbf{B}^2) = tr(\mathbf{S}) = \sum_{i=1}^{k}\sum_{j=1}^{k} cos^2(\theta_{ij}) = \sum_i \lambda_i < k \quad (5)$$

However, the measure which performs the best in our experiments was an average of the reconstruction error:

$$1/n_1||\mathbf{D}^1 - \mathbf{B}^2(\mathbf{B}^2)^T\mathbf{D}1||_F + 1/n_2||\mathbf{D}^2 - \mathbf{B}^1(\mathbf{B}^1)^T\mathbf{D}2||_F \quad (6)$$

where $\mathbf{D}^1 \in \Re^{d\times n_1}$ denotes the first set of data and $\mathbf{D}^2$ the second one. If two sets overlap over time, they can not belong to the same class. In order to enforce this constraint, a bigger than the average reconstruction error is given as an input in the affinity matrix. Once the affinity matrix is created, recent advances in Normalized Cuts [34] are used to solve the clustering problem.

4 Multiple Face Recognition

A very important task in meeting understanding is to know who is attending to the meeting, and CAMEO's task is to infer people's identity from the mosaic images. Face recognition from images/video is quite a complex problem, which suffers from misalignment, high dimensionality of the visual data, occlusions, facial expression changes and illumination variations. Due to its difficulty and usefulness as a biometric, there exists a huge literature and there are many available techniques for face recognition from images (see [36] for a review). In this section, a new dimensionality reduction technique, Multimodal Oriented Discriminant Analysis (MODA) [5], is used to optimally reduce the dimension of the data for fast recognition. Moreover, in order to deal with unexpected variation changes, several strategies that exploit spatio-temporal redundancy and are able to recognize several people simultaneously are introduced to improve performance.

Figure 8: a) Original training images (800). b) 40 out of 75 clusters.

4.1 Preprocessing

The first step towards face recognition is to construct a statistical model of facial expression/pose/illumination variations of a person. Using the same automatic way of gathering visual data as for learning PSFAM, CAMEO collects around 800 frontal faces. After a geometric normalization using Robust Parameterized Component Analysis [4], a dimensionality reduction to filter noise and alleviate computational burden is performed. There are two types of "dimensions" that would be interesting to reduce: the number of pixels and the number of samples. Let $\mathbf{D}^i \in \Re^{d \times n_i}$ be a data matrix containing the gathered images for class i, usually $n_i \approx 800$ (off-line). In the n_i images collected for class i, there is not necessarily a uniform sampling over expressions or illuminations, which can bias the posterior classifier. To avoid this problem the p most representative samples ($p << n_i$) are selected by clustering $\mathbf{D}^i$. To cluster high dimensional data, recent advances in multi-way normalized cuts [34] are used. Figure (8.a) shows some images of the original 800 samples and (8.b) represents 40 prototypes out of 75. Observe that the variations are mostly due to facial expression, scale, and illumination changes.

After the selection is done, a data matrix $\mathbf{D} \in \Re^{d \times (p \times c)}$, containing the data from all the classes (c), is created. The redundancy in the column space of $\mathbf{D}$ will be inconvenient for several reasons: firstly any type of discriminative learning (e.g. classifiers) will suffer over-fitting effects with lots of correlated data, and secondly we are interested in reducing the computational burden of the overall algorithm. Each image has been rescaled to be 60×60 pixels (3600-dimensional vector). In order to reduce the dimensions of the column space of $\mathbf{D}$, principal component analysis (PCA) is applied after normalizing the data ($||\mathbf{d}_i|| = 1 \; \forall \; i$ and subtract the mean). PCA projects the data into the subspace spanned by the eigenvectors of the covariance matrix $\mathbf{D}\mathbf{D}^T \in \Re^{d \times d}$; however, for large amounts of data where $d >> n$, it is more numerically convenient to compute the eigenvectors of $\mathbf{D}^T\mathbf{D} \in \Re^{n \times n}$ [31]. $\mathbf{D}\mathbf{D}^T$ has the same eigenvalues as $\mathbf{D}^T\mathbf{D}$ and the eigenvectors are related by $\mathbf{D}$. Observe that by projecting onto the principal components, some discriminatory power could be lost; however, it is worth to point out a couple of aspects. Firstly, by projecting onto the principal components, the generalization performance in the case of $d >> n$ probably will be better, and secondly, the true dimensionality of $\mathbf{D} \in \mathbf{R}^{d \times n}$ when $d >> n$ is n, so that by projecting into the first k eigenvectors whose eigenvalues are different from zero, no discriminatory power is lost. We project onto the PCs which preserve 99% of the energy.

4.2 Multimodal Oriented Discriminant Analysis

Given a detected/tracked face, CAMEO will classify it into one of the c classes. One naive solution would be to match each test image with each of the prototypes (75) in the class i; however, this nearest neighbor classifier is not very efficient since k dimensions have to be matched. Multimodal Oriented Discriminant Analysis (MODA) [5], a new method that generalizes Linear Discrimiant Analysis (LDA) will be used. MODA will allow to perform fast matching ($m << k$), avoid overfitting, and improve recognition performance with respect to LDA.

For each class, there are around 800 original images and 75 prototypes. The original 800 images are clustered ([34]) into s clusters, typically between 2 and 5, which mostly take into account scale changes. Each of these clusters r will be modeled as a high-dimensional Gaussian $N(\mathbf{x}; \boldsymbol{\mu}_i^r, \boldsymbol{\Sigma}_i^r)$ for each class i. MODA seeks for a low dimensional projection $\mathbf{B} \in m \times k$, common to all the classes (i.e. $N(\mathbf{B}^T\boldsymbol{\mu}_i^r, \mathbf{B}\boldsymbol{\Sigma}_i^r\mathbf{B}^T)\ \forall i, r$), that maximizes the Kullback-Leibler (KL) divergence [7] between the clusters of different classes in the low dimensional space, but does not impose distance constraints between the clusters of the same class. After some algebraic arrangements, it can be shown that the optimal dimensionality reduction is given by [5]:

$$\sum_i \sum_{r_1 \in C_i} tr\big((\mathbf{B}^T\boldsymbol{\Sigma}_i^{r_1}\mathbf{B})^{-1}(\mathbf{B}^T(\sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T + \boldsymbol{\Sigma}_j^{r_2})\mathbf{B})\big) \quad (7)$$

where $\boldsymbol{\mu}_i^{r_1}$ is the r_1 cluster of class i, and $\sum_{r_1 \in C_i}$ sums over all the clusters belonging to class i. Eq. 7 is quite hard to optimize with respect to $\mathbf{B}$; second-order type of gradient methods (e.g. Newton or conjugate gradient) do not scale well with huge matrices. Moreover, in this particular energy function the second derivative is quite complex. A bound optimization method called iterative majorization [13] is used instead. Iterative majorization (similar to Expectation Maximization type of algorithms) is able to monotonically reduce the value of the energy function. For details of the optimization see [5].

Despite reducing the dimensionality of the data from d to k with PCA, fitting discriminative models such as MODA can easily suffer from over-fitting problems and lack of generalization. In order to be able to generalize better and not suffer from storage and computational requirements, the covariance matrices are approximated as the sum of outer products plus a scaled identity matrix $\boldsymbol{\Sigma}_i \approx \mathbf{U}_i\boldsymbol{\Lambda}_i\mathbf{U}_i^T + \sigma_i^2\mathbf{I}_k$ where $\mathbf{U}_i \in \Re^{k \times l}$ and $\boldsymbol{\Lambda}_i \in \Re^{l \times l}$ is a diagonal matrix. In order to estimate the parameters $\sigma_i^2,\ \mathbf{U}_i,\ \boldsymbol{\Lambda}_i$, a fitting approach is followed by minimizing $E_c(\mathbf{U}_i, \boldsymbol{\Lambda}_i, \sigma_i^2) = ||\boldsymbol{\Sigma}_i - \mathbf{U}_i\boldsymbol{\Lambda}_i\mathbf{U}_i^T - \sigma_i^2\mathbf{I}_k||_F$. It can be shown that the optimal solution satisfies $\mathbf{U}_i\boldsymbol{\Sigma}_i = \mathbf{U}_i\hat{\boldsymbol{\Lambda}}_i$, $\sigma_i^2 = tr(\boldsymbol{\Sigma}_i - \mathbf{U}_i\hat{\boldsymbol{\Lambda}}_i\mathbf{U}_i^T)/k - l$, $\boldsymbol{\Lambda}_i = \hat{\boldsymbol{\Lambda}}_i - \sigma_i^2\mathbf{I}_k$ [5].

It is worthwhile to point out two important aspects of the previous factorization. It is an efficient (in space and time) manner to deal with the small sample case, since we can compute $\boldsymbol{\Sigma}_i\mathbf{B} = \mathbf{U}_i\boldsymbol{\Lambda}_i(\mathbf{U}_i^T\mathbf{B}) + \sigma_i^2\mathbf{B}$. In this case we do not need to explicitly have the full covariance matrix. On the other hand, observe that the original covariance has $k(k+1)/2$ free parameters, and with the factorized matrices ($\mathbf{U}_i, \boldsymbol{\Lambda}_i, \sigma_i^2$) the number of parameters is reduced to $l(2k - l + 1)/2$ (assuming orthonormality of $\mathbf{U}_i$), so we need much less data to estimate these parameters and hence it is not so prone to over-fitting.

Figure 9: Training data.

Due to the lack of public databases for face recognition from video, we have collected one to test our approach. A database of 23 people has been recorded over two different days (two weeks apart) under different illumination conditions. Figure 9 shows some images from people in the database, variations are mostly due to facial expression, pose, scale and illumination conditions. The training set consist of the data gathered during the first day, under three different illumination conditions (varying lights in the recording room), scale, and expression changes. The testing data consist of the recording of the second day (a couple of weeks later) under similar conditions. Figure 10 illustrates the recognition performance using PCA, LDA and MODA, similarly table 4.2 gives some detailed numerical values for the different number of bases.

Basis	2	5	10	15	20	25	30	40	50	60
PCA	0.12	0.26	0.43	0.50	0.55	0.56	0.58	0.59	0.59	0.60
LDA	0.21	0.36	0.48	0.54	0.56	NA	NA	NA	NA	NA
MODA	0.23	0.38	0.50	0.57	0.59	0.60	0.61	0.62	0.63	0.63

Table 3: Recognition performance of PCA/LDA/MODA (23 classes)

In this experiment, each class has been clustered into two clusters to estimate $\mathbf{B}$. Once $\mathbf{B}$ is calculated, the Euclidean distance for the nearest neighborhood is used. Several metrics have been tested (e.g. Mahalanobis, Euclidean, Cosine, etc) and the Euclidian distance (with 75 prototypes) is the one which performs the best in our experiments. For the same number of bases, MODA outperforms PCA/LDA. Also, observe that LDA can extract just classes-1 features (22 features), whereas MODA can

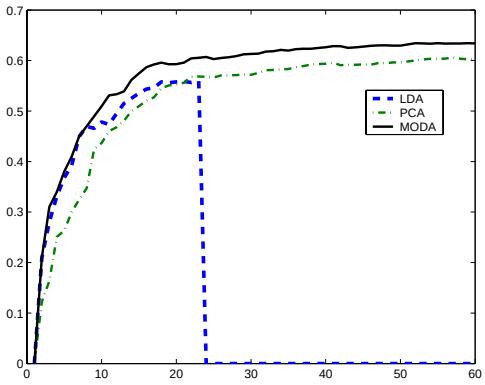

Figure 10: PCA/LDA/MODA

extract many more features. By matching just 20-dimensional vector rather than the original 3600-dimensional one, MODA is able to get more accurate better recognition performance with less computational complexity.

4.3 Improving performance

Although MODA is the optimal linear dimensionality reduction technique that preserves discriminability power among classes, it does not handle changes in the statistical properties of the training data due to undesirable noise very well (e.g. different illumination, hair configuration, misregistration). Three strategies have been tested in order to handle such unexpected situations:

- Use a verification step to reject samples (outlier detection).
- Integrate temporal information into the classification process.
- Recognize multiple people simultaneously.

4.3.1 Pattern rejection and verification

Face recognition from video has the advantage of having a lot of temporal redundancy that can be exploited. In order to construct robust systems against surprise or untrained situations, not all of the adquiered samples are going to be classified, only the ones that are reliable. In order to decide if a sample should be used for classification or not, a simple outlier detection strategy is used.

When CAMEO detects the face of a person, the first step it performs is to determine if the person is in the database or not. Two thresholds (one generative and the other discriminative) are used in order to determine if the sample belongs to any of the classes. Both thresholds are computed from the covariance of the projected data. That is, the training data have been projected onto the discriminative or generative subspace (i.e. $\mathbf{C} = \mathbf{B}^T\mathbf{D}$), and a threshold is established by the variance of this distribution, that

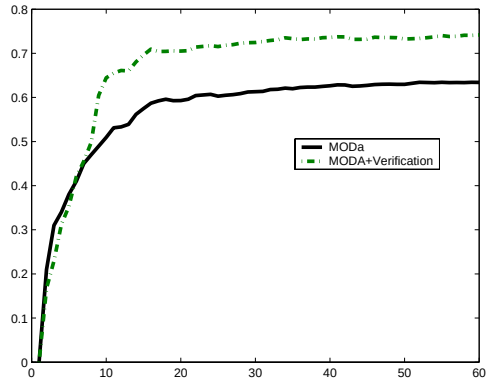

Figure 11: MODA and MODA + a verification step

is $\mathbf{C}\mathbf{C}^T$. The covariance naturally becomes diagonal (due to a first PCA step). Outliers will be considered the ones that are 4 times far away from the standard distribution.

If the first global step is not an outlier, the second one uses a verification method to check if the data belong to a local subspace. For each class i the principal components that preserve 80% of energy are computed, $\mathbf{U} \in \Re^{d \times k_i}$. The average error ($\mu_i$) and the variance ($\sigma_i$) of the distance from the subspace (DFS) ($||\mathbf{I} - \mathbf{U}\mathbf{c}||_2^2 = ||\mathbf{I}||_2^2 - \mathbf{c}^T\mathbf{c}$) for class i, are computed, as well as the average error (μ_o) and the average variance (σ_o) of the distance from the subspace for all the classes but class i. Once the mean and variance for the inter class DFS of class i, and the mean and variance for the intra class DFS for class i are obtained, a quadratic classifier that minimizes the classification error is calculated. That is, an optimal threshold (T), which reduces the classification error ($\int_T^\infty P_i p_i(x)dx = \int_{-\infty}^T P_o p_o(x)dx$), where P_i and P_o, are the a priori probabilities of each class. Assuming that $p_i(x)$ and $p_o(x)$ are Gaussian, the optimal threshold is given by the solution of the following second order equation:

$$T^2(\frac{1}{\sigma_i^2} - \frac{1}{\sigma_o^2}) + 2T(\frac{\mu_i}{\sigma_i^2} - \frac{\mu_o}{\sigma_o^2}) + (\frac{\mu_i^2}{\sigma_i^2} - \frac{\mu_o^2}{\sigma_o^2} - 2ln(\frac{P_i\sigma_o}{P_o\sigma_i})) \tag{8}$$

Basis	2	5	10	15	20	25	30	40	50
MODA+verification	0.16	0.35	0.64	0.69	0.70	0.71	0.72	0.73	0.73
Percentage(%)	45%	44%	49%	51%	50%	51%	51%	52%	50%

Table 4: Recognition performance of MODA and MODA+outliers. The $\%$ indicates the percentage of inlier data.

Figure 11 shows the comparison between MODA and MODA plus a generative verification step. Table 4.3.1 shows more detailed values for some basis and the percentage of data that has not been discarded. Discarding approximately half of the data greatly improves the recognition performance. In the real time implementation, all the

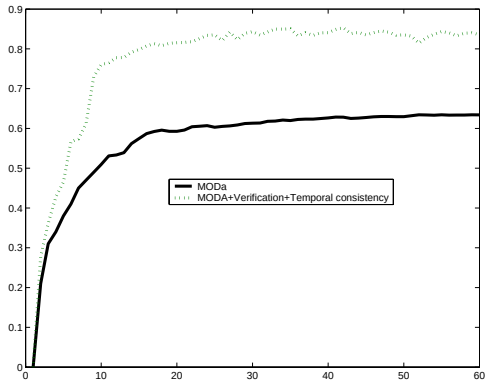

Figure 12: MODA and MODA + Verification step + Temporal voting

thresholds, the global and local ones, vary proportionally to one parameter, which is adapted depending on the amount of data that has been classified.

4.3.2 Temporal consistency

Up to now, CAMEO uses one single image to determine classification; however, a key aspect to improve recognition performance consist of integrating all the evidence coming from the video stream. Preliminary work has been done to match sets of images rather than individual images. Yamaguchi et al. [32] propose to compute the principal angle as a distance between two sets. However, this measure is too sensitive to possible outliers, since it is based on computing the minimum correlation between two vector (one on each subspace). Shakhmarovich et al. [28] took into account a more probabilistic approach, modeling both sets as a high dimensional Gaussian, and then computing the Kullback-Leibler distance between distributions. In this paper, a simple voting scheme is used. The data is first projected with the basis obtained with MODA, and the class with less error is chosen. After the verification step, if the reconstruction error for this class falls within a range, the sample is used for classification. The same procedure is followed with the rest of the samples, and when 10 samples are collected CAMEO assigns the set of images to the class where more samples have been classified. Figure 12 compares the recognition performance w.r.t. just using MODA. Table 4.3.2

Basis	2	5	10	15	20	25	30	40	50
MODA+verification +voting	0.27	0.46	0.76	0.79	0.81	0.83	0.84	0.84	0.84

Table 5: Recognition peformance using MODA + Verification step + Temporal voting.

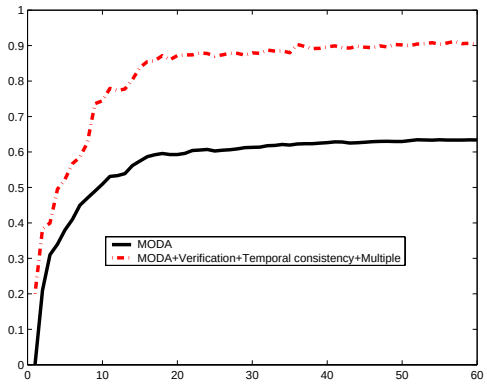

Figure 13: MODA and MODA + Verification + Temporal consistency + Multiple face Recognition.

4.3.3 Multiple face recognition

Recognizing several people simultaneously greatly improves the recognition performance, since the constraint that two people can not have the same identity is implicitly imposed. To incorporate these constraints into the classification problem, the multiple face recognition problem is posed as the classical assignment (transport) problem.

A matrix $\mathbf{A}$ with m rows (number of people tested) and n columns (number of people in the database) is created, where each position a_{ij} correspond to the average error of the last 10 samples for person i w.r.t class j. The error is computed as the average of the minimum error from the prototypes of class i. Once this matrix is created, the assignment problem is the task of finding a permutation $\pi[0]\,,\,\cdots\,,\,\pi[n-1]$ of $\{0,\cdots,n-1\}$ such that $\sum_{k=0}^{n-1} a_{k\pi[k]}$ is minimized [11]. This is typically solved when $(m=n)$ with the Hungarian algorithm. In our case, $m<n$, and $n-m$ dummy rows with zero value are added, which do not affect the result of the algorithm. The code for the Hungarian algorithm has been copied from [11].

Fig. 13 shows the recognition performance exploiting spatio-temporal information and outlier detection vs. using just MODA. As we can see, around 40% of increase in performance is achieved. Imposing constraints in the multiple face recognition improves on average 5% of recognition performance and it can be time consuming for real time applications.

Basis	2	5	10	15	20	25	30	40	50
MODA+ Verification +Temporal+ Multiple	0.38	0.52	0.74	0.83	0.87	0.87	0.87	0.89	0.90

Table 6: Multiple face recognition with a window of 10 samples.

Figure 14: Some images illustrating the depth estimation. In the upper black box the depth(Z) and the (Y) coordinates can be observed.

5 Experiments

5.1 Depth estimation

The first experiment tests CAMEO's ability to infer distances from a video sequence. Knowing the relative position of one person with respect to a CAMEO's camera is an important feature for meeting understanding. Once a face is detected, an average size of the head of the person is assumed ($12cm$ wide and $17cm$ high). Using eq. 2 and considering that the face is a plane oriented towards the camera (rotational angles 0), the translational components are straight forward to compute. Figure 14 shows some results for the depth estimation. In the black box, Z indicates the estimated depth and Y the altitude with respect to the ground. A video with ground truth can be downloaded from www.salleURL.edu/~ftorre/distance.avi . CAMEO is able to estimate the depth with a factor error of less than 4%.

5.2 Multiple face recognition

A four people meeting has been recorded and can be downloaded from ftp://brim.coral.cs.cmu.edu/pub/calo/presentations/icra05/recognition.avi. The duration of the video is approximately one minute. CAMEO has pre-stored facial models for three of the people attending the meeting. After a few frames CAMEO starts recognizing people and using PSFAM to track; however, the third person starting from the left is not recognized. CAMEO gathers a few frames of the unknown person (≈ 50) and asks the person if he/she wants to be introduced into the database. In this case, the user agrees by pressing a key and the model is stored in the database. Once the model is added to the database, CAMEO successfully recognizes the new person. Observe that at the end of the video CAMEO does not recognize the new person, probably due to big scale changes; however, it detects that the new person is in the database and the label 'Need Data' is shown.

The multiple face recognition constraint has not been used in this video, since it only increases the recognition performance 5% (in our database experiments) and it can be time consuming. If two users have the same identity, the one with less error is chosen. Although not the optimal solution, it is a reasonable approximation for real-

Figure 15: Tracking multiple faces.

time applications.

5.3 Multiple people tracking

A four people meeting scenario is recorded and the video can be downloaded from www.salleURL.edu/ ~ftorre/tracking.asf. In figure 15, one frame of this meeting is shown. In the first frames, CAMEO automatically identifies the people and assigns them his/her PSFAM (previously learned in similar environmental conditions). CAMEO is able to track multiple heads using PSFAM, despite fast head motion, partial occlusion, and crossing people. Occasionally, the head tracker is lost due to very fast motion, motion blur, or frames with different training conditions; when this situation occurs the face detector is executed again (red square). The PSFAMs use between 5-7 bases and run at 5-6 fps. Observe that in the mosaic construction, there exist some blurring effects in the overlapping area between the cameras. This is due to the parallax existing between the cameras; however, the automatic adjustment reduces this effect. A similar tracking in a meeting scenario of 5 people can be downloaded from www.salleURL.edu/ ~ftorre/tracking2.asf.

5.4 On-line learning and adaptation

In figure 16 some images of the online method for learning PSFAM are shown. At the beginning, the face is tracked with normalized correlation until several images are gathered (8). Each of the added samples are first equalized, normalized, and resized to 28x36 pixels. Later, the set of bases is adapted to compensate for appearance changes. The subspace tracker is faster, more accurate and robust than a simple template matching. The video can be downloaded from $http://www.salleURL.edu/\tilde{}ftorre/online\_adaptation.asf$.

5.5 Learning PSFAM from long video sequences

We have recorded 1 hour video of a meeting with five attendees. The adaptive subspace tracker runs automatically for each person's face until it gets lost (usually after several minutes). Each of the trackers runs the frontal face detection in the tracked area in order to gather frontal people's faces. Figure 17 shows an example of set of faces collected by CAMEO. Each tracker starts or stops independently. CAMEO outputs 90 folders, where each folder contains more than 10 frontal faces of the same person. Some of the sets overlap over time and the ultimate goal is to put in correspondence all the folders. That is, the folders containing the same person should be grouped. As

Figure 16: Online learning and Adaptation. 4 of the 9 basis are shown in the lower left.

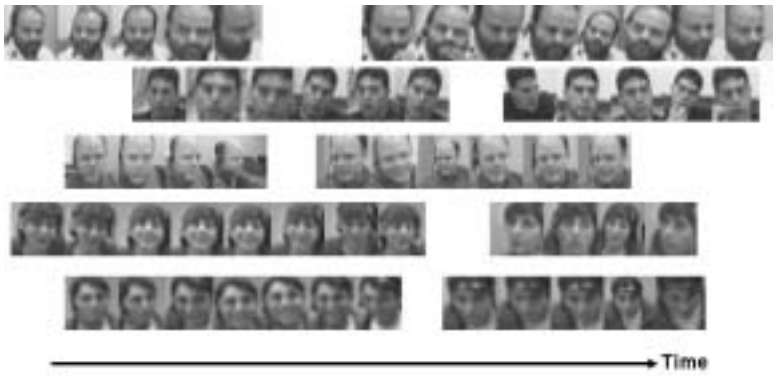

Figure 17: Gathered faces. Each tracker starts and ends independently over time.

explained in section 3, for each set (e.g $\mathbf{D}^1$ and $\mathbf{D}^2$) the principal components ($\mathbf{B}^1$ and $\mathbf{B}^2$) are computed and the distance between the two sets is calculated as the average residual $d(\mathbf{D}^1, \mathbf{D}^2)^S = 1/n_1||\mathbf{D}^1 - \mathbf{B}^2(\mathbf{B}^2)^T\mathbf{D}1)||_F + 1/n_2||\mathbf{D}^2 - \mathbf{B}^1(\mathbf{B}^1)^T\mathbf{D}2)||_F$. If the sets overlap over time they can not belong to the same cluster, hence a bigger distance (e.g. 1.5*$d(\mathbf{D}^1, \mathbf{D}^2)^S$) is introduced. Moreover, in order to take into account spatial information and assuming smoothness of motion, the final distance is computed as the product of the distance between subspaces and a geometric distance. That is, $d(\mathbf{D}^1, \mathbf{D}^2) = d(\mathbf{D}^1, \mathbf{D}^2)^S * d(\mathbf{D}^1, \mathbf{D}^2)^G$, where $d(\mathbf{D}^1, \mathbf{D}^2)^G$ is proportional to the distance between the location of the face in the last frame of the starting set ($\mathbf{D}^1$ or $\mathbf{D}^2$) and the location of the face of the other set.

After creating the similarity matrix, Normalized Cuts [34] is used to cluster the affinity matrix. This procedure links correctly the sets which belong to the same person. Figure 18 shows some images belonging to each of the five clusters over 1 hour of video. Recall, that in the first and second row there are some outliers. Also, observe that in the fourth row, there is an image another person, this was because the tracker jumped

Figure 18: Result from clustering 90 sets containing 5 people.

from one person to another for few frames. Once the faces have been automatically gathered, we can construct the PSFAM easily as explained in section 3.

6 Discussion and future work

In this paper we have introduced CAMEO, a hardware/software component to record and to extract useful visual information for meeting understanding. Several novelties for face identification, tracking, and mosaic generation have been introduced. The use of FSFAM has proven to provide robust, reliable, and fast tracking. On the other hand, the on-line adaptation strategy is necessary to be able to adapt to new environments. In the identification aspect, the combination of generative and discriminative models has proven to be very effective to increase the recognition performance. However, several aspects remain to be researched and extended:

- In order to improve the mosaic generation, better distortion models should be used (e.g. non parametric ones [29]).
- Better algorithms for reducing parallax problem in the matching have to be researched.
- To gather higher resolution data of some meeting events (such as what is being written on the blackboard or gathering higher resolution face images), a pan/tilt/zoom camera should be added.
- Face-to-face meeting encompasses several modalities such as speech, gesture, and handwriting which should be added to CAMEO's capabilities. For instance, recognition could be improved by adding speech.
- Capturing high-quality audio in a meeting room is a challenging problem due to the variety of noises, reverberation, etc., which should be removed. In future versions, we will record directly from a microphone.

Besides the meeting scenario, CAMEO could be used to target applications such as classroom lectures, distance learning, and video conferencing. Also, we are working on

the audio-visual summarization aspects of the meeting. For instance, we are interested in automatically detecting changes in facial expression for all the attendants, detect when everybody tries to speak/laugh, or who wrote in the blackboard. Moreover, more research will be conducted towards temporal segmentation of the meeting into simple events (monologue, discussion, start/end, presentation, etc.).

Acknowledgements Thanks to Jordi Casoliva for the fast energy normalization code and preliminary versions of the face tracker based on normalized correlation. Thanks to B. Browing and B. Ricker for constructing the camera mount. Thanks to R. Patil, Jon White, Yoni Wexler, F. Tamburrino, R. Cutler and R. Swaminathan for helpful comments and discussions.

References

[1] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*, pages 707–720, 2002.

[2] M. Conferencing. Meetings in america: A study of trends, costs and attitudes toward business travel, teleconferencing, and their impact on productivity. In *A network MCI Conferencing White Paper.*, 1998.

[3] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *ACM Multimedia*, 2002.

[4] F. de la Torre and M. J. Black. Robust parameterized component analysis: theory and applications to 2d facial appearance models. *Computer Vision and Image Understanding*, 91:53 – 71, 2003.

[5] F. de la Torre and T. Kanade. Multimodal oriented discriminant analysis. In *tech. report CMU-RI-TR-05-03, Robotics Institute, Carnegie Mellon University, January 2005*.

[6] J. Foote and D. Kimber. Flycam: Practical panoramic video and automatic camera control. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 1419–1422, 2000.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press.Boston, MA, 1990.

[8] G. Golub and C. F. V. Loan. *Matrix Computations*. 2nd ed. The Johns Hopkins University Press, 1989.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press., 2000.

[10] J. Heikkil and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition*, 1997.

[11] D. E. Knuth. *The Standford GraphBase*. Addison-Wesley Publishing Company, 1993.

[12] W. J. Krzanowski. Between-groups comparison of principal components. *Journal of the American Statistical Association*, 47(367):703–707, 1979.

[13] J. D. Leeuw. *Block relaxation algorihtms in statistics*. H.H. Bock, W. Lenski, M. Ritcher eds. Information Systems and Data Analysis. Springer-Verlag., 1994.

[14] J. P. Lewis. Fast normalized cross-correlation. In *Vision Interface*, 1995.

[15] I. Mikic, K. Huang, and M. Trivedi. Activity monitoring and summarization for an intelligent meeting room. In *IEEE Workshop on HUman Motion*, 2000.

[16] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *Pattern Analysis and Machine Intelligence*, 19(7):137–143, July 1997.

[17] R. B. Nelson and P. Economy. Better business meetings. In *McGraw-Hill.*, 1995.

[18] M. Nicolescu and G. Medioni. Globeall: Panoramic video for an intelligent room. In *Proceedings of the International Conference on Pattern Recognition*, pages 823–826, 2000.

[19] P. Peer and F. Solina. Panoramic depth imaging: Single standard camera approach. *International Journal of Computer Vision*, 47:149–160, 2002.

[20] S. Pelg and J. Herman. Panoramic mosaics by manifold projection. 1997.

[21] P. Robertson, R. Laddaga, and M. Van Kleek. Virtual mouse vision based interface. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 177–183. ACM Press, 2004.
[22] D. Ross, J. Lim, and M. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Eighth European Conference on Computer Vision*, 2004.
[23] Y. Rui, A. Gupta, and J. J. Cadiz. Viewing meetings captured by an omni-directional camera. In *ACM-CHI*, 2001.
[24] P. Rybski and M. Veloso. Using sparse visual data to model human activities in meetings. 2004.
[25] P. E. Rybski, F. de la Torre, R. Patil, C. Vallespi, M. Veloso, and B. Browning. Cameo: Camera assisted meeting event observer. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
[26] H. Schneiderman. Feature-centric evaluation for cascaded object detection.. In *CVPR*, 2004.
[27] H. Schneiderman. Learning a restricted bayesian network for object detection. In *CVPR*, 2004.
[28] G. Shakhnarovich, J. W. Fisher, and T. Darrell. Face recognition from long-term observations. In *ECCV*, 2002.
[29] R. Swaminathan and S. K. Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1172–1178, 2000.
[30] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31(Annual Conference Series):251–258, 1997.
[31] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal Cognitive Neuroscience*, 3(1):71–86, 1991.
[32] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *IEEE International Conference on Automatic Face and Gesture Recognition.*, 1998.
[33] J. Yang and A. Waibel. A real-time face tracker. In *Proceedings 3rd IEEE Workshop on Applications of Computer Vision*, pages 142–147, 1996.
[34] S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, 2003.
[35] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, 1999.
[36] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM computing surveys*, 35(4):399–458, 2003.