Learning to Track Multiple People in Omnidirectional Video. [*]

Fernando De la Torre Carlos Vallespi Paul E. Rybski Manuela Veloso Takeo Kanade
Robotics Institute. Carnegie Mellon University. 5000 Forbes Ave.Pittsburgh, PA 15213
ftorre@cs.cmu.edu, cvavlles@cs.cmu.edu, prybski@cs.cmu.edu, veloso@cs.cmu.edu ,tk@cs.cmu.edu

***Abstract*— Meetings are a very important part of everyday life for professionals working in universities, companies or governmental institutions. We have designed a physical awareness system called CAMEO (Camera Assisted Meeting Event Observer), a hardware/software system to record and monitor people's activities in meetings. CAMEO captures a high resolution omnidirectional view of the meeting by stitching images coming from almost concentric cameras. Besides recording capability, CAMEO automatically detects people and learns a person-specific facial appearance model (PSFAM) for each of the participants. The PSFAMs allow more robust/reliable tracking and identification. In this paper, we describe the video-capturing device, photometric/geometric autocalibration process, and the multiple people tracking system. The effectiveness and robustness of the proposed system is demonstrated over several real-time experiments and a large data set of videos.**

***Index Terms*— Omnidirectional-video capturing, Multiple people tracking, Subspace methods, Meeting understanding, Person-specific models.**

I. Introduction

Meetings are an integral part of business life. A mid-level manager or professional spends approximately 35% of his or her time in meetings. 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 a meeting, 39% have dozed during a meeting, and many of those attending a meeting need to clarify miscommunications [12]. Having systems that help to review/share meetings can help to correct 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 is a preliminary step towards 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 and personal tasks that they engage in. The goal of the larger project CALO (Cognitive Agent that Learns and Organizes) is to build a personalized computational resource that will be able to handle routine tasks/events, anticipate predictable user needs and appropriately prepare for them, including the handling of unexpected events.

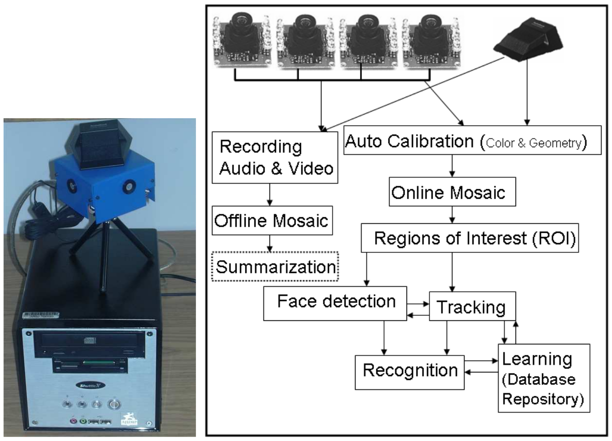

Fig. 1. CAMEO a) Hardware b) Software

Instead of instrumenting meeting rooms, CAMEO is intended to be used more as a speaker phone for a conference call. That is, a CAMEO device will be brought into a meeting and simply placed in the center of the room without requiring special manual calibration. As such, CAMEO is designed to be used in environments where those who are participating in the meetings agree to and welcome the use of such an electronic assistant. Apart from omnidirectional audio-video recording capabilities, CAMEO will be able to answer (real time) who is in the meeting, where each of the participants is, and when some events occur. This is a first step towards understanding human activity in meetings [18]. Figure 1 shows the hardware and a block diagram of the software capabilities of CAMEO.

The hardware is composed of 4 daisy-chained web-cameras and an omnidirectional microphone. There are just two cables, one firewire for the cameras and one for the microphone, connected to a shuttle PC or a laptop. There are two main benefits of the hardware design, a 360 degrees of field of view and a portable device, so that there is no need for instrumented rooms to record meetings. The software component is divided into two parts: one that operates in real time and the other off-line. In the real time processing, CAMEO will take the images coming from the four cameras and construct an approximate mosaic. Once the mosaic is constructed, CAMEO will be able track and

[*] This work has been partially supported by National Business Center of the Department of the Interior under a subcontract from SRI International and U.S. Department of Defense contract N41756-03-C4024, NIMH Grant R01 MH51435.

recognize multiple people in the omnidirectional video. The tracker and recognition systems are effective and robust because they are based on a set of learned person-specific appearance models. The off-line part records the audio and the video stream coming from the 4 cameras, and in later processing builds a more accurate mosaic [3].

II. 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 [10], [13], concentric cameras devices [1], [4], [22] and various instruments [16] to record human activity in meetings. In order to have a global view of the meeting, CAMEO will capture a $360^{\mathbf{o}}$ degrees of field of view with an omnidirectional camera. Many techniques have been researched for constructing panoramic images from real-world scenes. Mirrored pyramids and parabolic mirrors [17] could be used to capture the images directly; however, in order to capture high resolution images, expensive equipment is needed and potential defocusing problems may result in low quality video. In our case, we are interested in minimizing the amount of cables and designing an inexpensive portable device. Similar to previous work on panoramic images for meeting recording [1], [4], [22], CAMEO will integrate images coming from almost concentric images into a single mosaic.

There exist several techniques to stitch images coming from several cameras [14], [1], [4], [23], [15]. However, most of them assume that the camera is panning, or that only rotation exist between cameras' optical centers. In the case that 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 [6]. However, in our case the cameras do not share a common center of projection, and parallax effects occur due to the translational component between the optical centers of the cameras. Having translational motion between cameras, the geometric transformation that relates two images becomes depth dependent (the parallax effect becomes more evident at shorter distances). One possible solution will involve computing depth for each point [14]; 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, the parallax effects can be ignored. In this section we explain the software/hardware details for constructing the camera device able to produce high resolution video sequences by stitching images coming from almost concentric cameras. Preliminary work has been presented at [19].

A. Real-time mosaicing

CAMEO is composed of 4 inexpensive web-cameras that have been daisy-chained, and just one firewire cable to transmit the signal and power, similar to [1], [4]. 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 slight overlap between the field of view of two cameras (further than 30 cm). A small focal length yields large depth of field, and as all objects are in focus from a distance of a few centimeters to infinity, autofocus is not required. However, these lenses introduce big radial and tangential distortion in the images. The first step toward 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 [7]:

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

where X, Y, Z are the 3D coordinates and x_p, y_p are the pixels position in the image.

Despite the fact that the cameras are mounted as close together as is practical, they do not share a common center of projection and the translational component introduces parallax effects (complicating the matching process). To minimize this effect and because of easy construction, cylindrical panoramas are commonly used [1], [22], [23]. 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, we construct a look up table (LUT) to correct for the distortion and the cylindrical mapping, using very efficient Intel Performance Primitives (IPP) functions. 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). In the on-line version, we ignore the translational component between the optical centers of the cameras and search for the translation that produces the best match between adjacent cameras. A constrained normalized template matching is computed to search for the optimal translational. Although gradient descent type of methods are possible [23], parallax effects and large change in viewpoint make them too sensitive to local minima. Finally, a weighted (more weight to the image which is closer) blending procedure is used to merge both images. Figure 2.a shows four original images and how they are merged into the mosaic 2.b. More details of the mosaic construction can be checked at [19], [3].

B. 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

[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 $\mathbf{D}$ All non-bold letters will represent variables of scalar nature. $\odot$ denotes Hadamard (point wise) product.

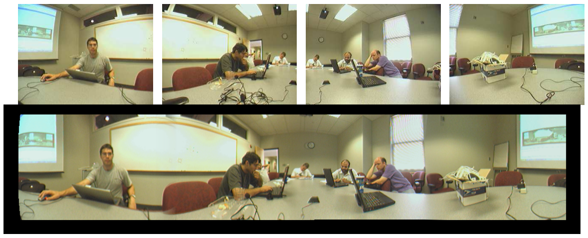

Fig. 2. a) Original images. b) Mosaic image

cameras, all of them should be mutually color-calibrated to ensure that they look alike. At the start, 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). Because of different lighting and CCD properties, we compute an affine transformation using overlapping regions. That is, given a set of matching points between 2 images, we compute the affine transformation $(\mathbf{A}, \mathbf{b})$ which minimize the error among matched points $\min_{\mathbf{A},\mathbf{b}} \sum_i ||\mathbf{x}_1 - \mathbf{A}\mathbf{x}_2 - \mathbf{b}||$. Finally, we use the camera drivers to access and change the hue and saturation values of each camera according to this in order to correct the color, which is hardware efficient.

A useful feature of CAMEO is to know the relative position of a person/pattern w.r.t. one of the cameras, because it allows us to calibrate between several CAMEO's, to estimate the depth of a person and to know the position w.r.t other devices. By knowing the internal camera parameters, and assuming a planar calibration pattern, it is relatively simple to estimate the relative orientation w.r.t the camera (e.g [24]). In order to simplify the scenario, we will assume 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. 3.a) and CAMEO automatically detects (in high resolution images) which camera sees the pattern and extracts the coordinates of the corners using normalized template matching at different scales. Without loss of generality, we assume 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 we are interested in recovering the rotational angles and translational components $\theta, \gamma,\ t_x,\ t_y,\ t_z$, by minimizing:

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

where $\mathbf{X}_i = (X_i,\ Y_i,\ Z_i)^T$, $\mathbf{T} = (t_x\ t_y\ t_z)^T$ and $(x_p^i,\ y_p^i)$ 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). $\mathbf{R}_1(\gamma) = \begin{pmatrix} cos(\gamma) & 0 & -sin(\gamma) \\ 0 & 1 & 0 \\ sin(\gamma) & 0 & cos(\gamma) \end{pmatrix}$, $\mathbf{R}_2(\theta) = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$, 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, we use a two step approach. 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). We sample the angle space for $\theta \in [0..2\pi]$ and $\gamma \in [0..2\pi]$, and for each value of θ, γ we solve the following linear system of equations:

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

$\dagger$ indicates the pseudo-inverse which is computed just once. In order to make the search efficient, we start sampling every $10^{\mathbf{o}}$, and when we have the minimum we make another local search, but with $1^{\mathbf{o}}$ resolution. Figure 3.b shows the error energy function for several values of θ and γ; in this case we have two valid solutions with the same energy value (due to planar ambiguity). We choose the parameters that give positive depth.

C. Software specifications

In order to ensure software stability, we divide the software in 4 main modules:

- **Video acquisition**: Acquires raw data from 4 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 library, and has two sub-modules: a. Correction of radial/tangential distortion and cylindrical mapping: LUT and bilinear interpolation. b. Mosaic calibration/tuning: Computes the translational error between overlapping areas of adjacent cameras. It keeps track of the overlapping error and recomputes the translation if needed.

- **Recording system (optional)**: Records the output of the acquired mosaic with Microsoft Windows Media 9 Series, which provides a set of features very convenient such as: audio and video synchronization, real time compression, streaming, 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, 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

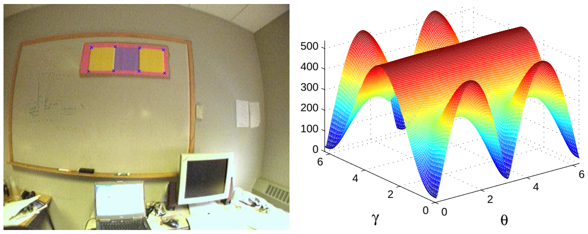

Fig. 3. a) Calibration pattern. b) Surface error.

a trade-off exists between resolution and frame rate. Table II-C shows the bandwidth and CPU times required for each configuration to build the mosaic. Finally, each meeting takes about 1.5G/hour to store (high quality video), most of which is 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 I

MEASURED WITH PENTIUM-M CPU AT 1.7 GHZ AND 1 GBYTE OF RAM.

III. 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. However, tracking multiple people is a challenging problem due to significant occlusion caused by interaction among people, deep changes in pose, and rapid motions. Moreover, in the CAMEO scenario, low quality video, low contrast, and varying illumination conditions complicate the tracking process. In this section we will describe the use of person-specific facial appearance models (PSFAM) for tracking multiple people.

A. *Learning person-specific facial appearance models*

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. Rather than use generic models, CAMEO will automatically learn PSFAM, which will allow a more robust and faster tracker. 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, otherwise it will learn the model on-line.

In the off-line version to learn PSFAM, a person sits in front of one of CAMEO cameras and performs different facial expressions under several pose/scale/illumination changes; approximately 1 minute of video is recorded. Given this video sequence, the frontal and profile faces are automatically detected using Scheiderman face detection algorithm [20], [21]. Figure 4.a shows some of the gathered images (64×64 pixels). One possible way of constructing a PSFAM will consist of selecting several prototypes (different scales and profiles). Once a set of prototypes for each person are selected, tracking is achieved by performing template matching with each of them and selecting the position with minimum error. However, as the number of templates increases, it becomes impractical to find the best match w.r.t each of the templates, and in our scenario, a more efficient and robust matching approach is necessary. To exploit the spatial redundancy existing in 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 Singular Value Decomposition (SVD) [5]. In order to get a better estimate of the subspace, the images have to be perfectly geometrically aligned w.r.t. the subspace. Parameterized component analysis [2] is used to achieve geometric (translation, rotation, scale) invariant learning. After the data has been registered w.r.t. the subspace which preserves 85% of the energy, it is clustered in approximately 120 prototypes in order to avoid the principal components being biased towards specific facial expressions/poses that are more common. Usually, the number of profile-gathered faces is lower than the frontal ones; this can bias the construction of a subspace. To avoid this situation, we first cluster each of the profile faces into 30 prototypes and the frontal faces into 50 prototypes. With these prototypes we construct the PSFAM. In figure 4.(b) the set of eigentemplates at one scale is displayed after applying parameterized component analysis. Usually three eigentemplates are constructed at 3 different scales by subsampling the training data. Also, CAMEO can learn the PSFAM on-line. More details are given in [3].

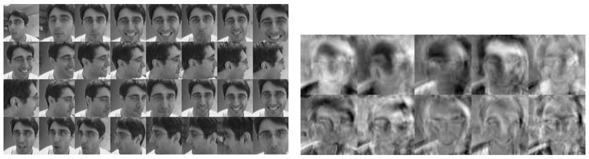

Fig. 4. a) Set of templates. b) First eigenbasis

B. *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:

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

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

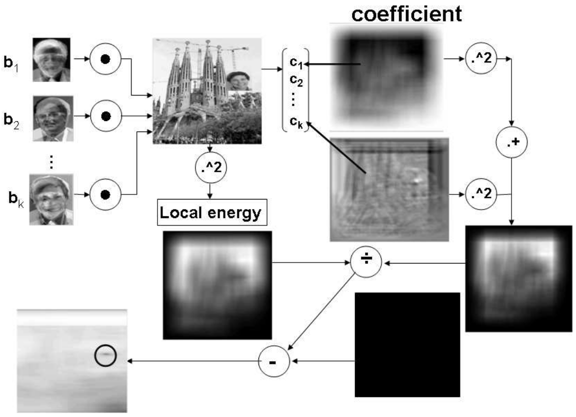

Fig. 5. A subspace is constructed with the 3 program chairs of ICRA-2005. One of the program chairs picture is included next to the Sagrada Familia in Barcelona. After computing the distance from the subspace, it can be seen that the minimum is where the face is located.

or time. A key observation in order to develop efficient methods is to observe that the error at a particular position (u, v) can be computed as $E(\mathbf{c}, u, v) = ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v) - \mathbf{B}\mathbf{c}||_2^2 = ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)||_2^2 - \mathbf{c}^T\mathbf{c}$ [11], where $\mathbf{c} = \mathbf{B}^T\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)$. Computing the coefficients $\mathbf{c}$ is equivalent to correlate the image with each basis of the subspace and stack all the values for each pixel. For big 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 we use the highly optimized OpenCV function for correlation. Finally the local energy term, $||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)||_2^2$, is computed very efficiently using the integral image [9]. In order to deal with local illumination changes, we normalize $\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)$, dividing by the square root of the energy, hence the total error can be expressed as $E(\mathbf{c}, u, v) = ||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v) - \mathbf{B}\mathbf{c}||_2^2 = 1 - \mathbf{c}^T\mathbf{c}/||\mathbf{I}(\mathbf{x} + u, \mathbf{y} + v)||_2^2$. Figure 5 shows an illustration of the subspace correlation method.

It is possible that during the off-line or on-line learning, some pose or facial expression is not captured by the model. If the error in the tracking exceeds some threshold, the face detector [20], [21] is run and a new face is gathered. The new face ($\mathbf{d}_t$) is added to the subspace by re-computing the set of basis using incremental SVD [**?**]. See [3] for a more detailed explanation.

C. Solving for correspondence and depth

When people cross or occlude each other it may happen that both trackers get lost or confused. Having PSFAM simplifies the data association problem greatly. Once two closer trackers are lost, CAMEO waits until it finds two faces again by means of the face detection [20], [21] and tracks them using normalized correlation. In order to assign which face belongs to which person, we use as a measure of closeness the directions in the subspace. One possibility is to measure some weighted cosines between pairs of eigenvectors corresponding to subspaces $\mathbf{B}^1, \mathbf{B}^2$; however these angles may be quite large even though the two subspaces are different. Following [8], we consider one of the subspaces ”fixed” and we find the best-matching set of orthogonal axes in the second subspace. It can be shown [8], that this is equivalent to compute the distance between two subspaces $\mathbf{B}^1 \in \Re^{d \times k_1}$ and $\mathbf{B}^2 \in \Re^{d \times k_2}$, $d(\mathbf{B}^1, \mathbf{B}^2)$ as: $0 < d(\mathbf{B}^1, \mathbf{B}^2) = tr(\mathbf{S}) = \sum_{i=1}^k \sum_{j=1}^k cos^2(\theta_{ij}) = \sum_i \lambda_i < k$, where $\mathbf{S} = (\mathbf{B}^1)^T\mathbf{B}^2(\mathbf{B}^2)^T\mathbf{B}^1$. Other possible measures such as the principal angle between subspaces [5] can be computed; however, principal angles can be very sensitive to outliers and we found that the measure proposed at [8] is much more robust and reliable.

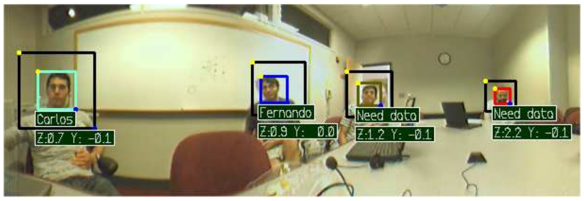

Fig. 6. Some images illustrating the depth estimation. In the upper black box the depth(Z) and the (Y) coordinates can be observed.

Knowing the relative position of one person w.r.t. one of the CAMEO cameras is an important feature for meeting understanding. Once the face is tracked or detected, we will assume an average size of the head of the person ($12cm$ wide and $17cm$ high) and that it is a plane perfectly oriented towards the camera (all rotational angles 0). This simplifies the equation 1, and the translational components are straight forward to compute. Figure 6 shows some results of the depth estimation.

IV. Experiments

In the first experiment we have tested CAMEO’s ability to infer distances from a video sequence. The video can be downloaded from www.salleURL.edu/~ftorre/distance.avi . From the video, we can observe that CAMEO is able to estimate the depth with an error less than 4%.

In the second experiment, we have recorded a meeting scenario with 4 people. The video can be downloaded from www.salleURL.edu/~ftorre/tracking.asf , and we can observe how CAMEO is able to track multiple heads using PSFAM. In figure 7 one frame of this meeting is shown. In the first frames, CAMEO automatically identifies the people and assigns his/her person-specific model that has been previously learned in similar environmental conditions. Observe that CAMEO is able to track people’s faces despite the fast head motion, partial occlusion and crossing between 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) and the new face is updated to the basis (just the last 3). The PSFAMs uses between 5-7 basis and runs at 15 fps. Several times there exist blurring effects in the overlapping area between the cameras due to the parallax effects; however, due to the automatic adjustment this effect will disappear in few frames.

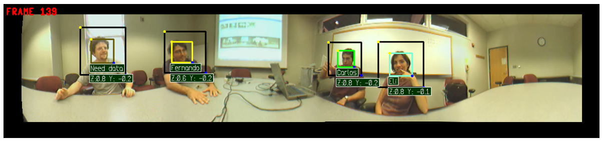

Fig. 7. Tracking multiple faces.

V. Conclusions and Future work

In this paper we have introduced CAMEO, a hardware/software component to record and extract useful visual information for meeting understanding. Several novelties for tracking and mosaic generation have been introduced. The tracking/recognition algorithms work in real time and provide robust, reliable and fast tracking due to the use of learned person-specific facial appearance models. These facial appearance models can be learned on-line or off-line. 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 [22]) so as to have a more flexible and accurate model for radial/tangential distortion.
- To gather higher resolution data of some meeting events (such as what people are writing on the blackboard or gathering higher resolution face images), a pan/tilt/zoom camera should be added.
- Capturing high-quality audio in a meeting room is challenging problem due to a 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 targets applications such as classroom lectures, distance learning, video conferencing, and more research should be done in this aspect. 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 attendees, 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.).

Acknowledgments

Thanks to Jordi Casoliva for the fast energy normalization code and preliminary versions of the face tracker. Thanks to B. Browing and B. Ricker for constructing the camera mount. Thanks to R. Patil, F. Tamburrino, R. Cutler and R. Swaminathan for helpful comments and discussions. Thanks Jon White for helping with the camera building and video recording.

References

[1] 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.

[2] 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.

[3] F. de la Torre, C. Vallespi, P. E. Rybski, M. Veloso, and T. Kanade. Omnidirectional video capturing, multiple people tracking and recognition for meeting understanding. Technical report, Robotics Institute, Carnegie Mellon University, January 2005.

[4] 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.

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

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

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

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

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

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

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

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

[13] 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.

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

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

[16] 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.

[17] Y. Rui, A. Gupta, and J. J. Cadiz. Viewing meetings captured by an omni-directional camera. In *ACM-CHI*, 2001.

[18] P. Rybski and M. Veloso. Using sparse visual data to model human activities in meetings. 2004.

[19] 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.

[20] H. Schneiderman. Feature-centric evaluation for cascaded object detection.. In *CVPR*, 2004.

[21] H. Schneiderman. Learning a restricted bayesian network for object detection. In *CVPR*, 2004.

[22] 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.

[23] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31(Annual Conference Series):251–258, 1997.

[24] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, 1999.