EVENT RECOGNITION FOR MEANINGFUL HUMAN-COMPUTER INTERACTION IN A SMART ENVIRONMENT

Ramon Morros [1], *Albert Ali Salah* [2], *Ben Schouten* [2], *Carlos Segura Perales* [1], *Jordi Luque Serrano* [1], *Onkar Ambekar* [2], *Ceren Kayalar* [3], *Cem Keskin* [4], *Lale Akarun* [4]

[1] Technical University of Catalonia, Barcelona, Spain
[2] Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands
[3] Computer Graphics Laboratory (CGLAB), Sabanci University, Turkey
[4] Perceptual Intelligence Laboratory, Department of Computer Engineering, Boğaziçi University, İstanbul, Turkey

ABSTRACT

The aim of this project is to monitor a room for the purposes of analysing the interactions and identities of a small set of individuals. We work with multiple uncalibrated sensors that observe a single environment and generate multimodal data streams. These streams are processed with the help of a generic client-server middleware called SmartFlow. Modules for visual motion detection, visual face tracking, visual face identification, visual opportunistic sensing, audio-based localization, and audio-based identification are implemented and integrated under SmartFlow to work in coordination across different platforms.

KEYWORDS

Pattern recognition – Image sensors – Image motion analysis – Audio systems

1. INTRODUCTION

The localization and recognition of spatio-temporal events are problems of great theoretical and practical interest. Two specific scenarios are nowadays of special interest: home environment and smart rooms. In these scenarios, context awareness is based on technologies like gesture and motion segmentation, unsupervised learning of human actions, determination of the focus of attention or intelligent allocation of computational resources to different modalities. All these technologies pose interesting and difficult research questions, especially when higher levels of semantic processing is taken into account for complex interaction with the users of the environment (See Fig. 1).

It is possible to envision a useful sensor-based system even without an elaborate hierarchical reasoning structure in the home environment, where low-cost sensor equipment connected to a home computer can be used for activity monitoring and in helping the user in various settings. In smart rooms, more sophisticated equipment is usually used, allowing for more robust applications. Furthermore, wearable sensors, sensor-instrumented objects and furniture are also used frequently [4].

This paper inspects some of the problems that arise from the use of non-calibrated, low-cost sensors for observing the room. Through the observation and subsequent processing, we try to determine some basic facts about the persons in the room: identity, spatial position and interactions between people. Using this knowledge, the application can also aim at higher level goals, such as controlling a door.

In the proposed application, a set of cameras and microphones continuously monitor a room. Each person entering the room is identified (by means of a camera aimed at the entrance door and also using the microphones inside the room). Then, the persons are tracked inside the room so that its spatial position is always known. Interactions between people can be detected by using their positions, ID, head orientations and information about if they are speaking or not. Tracking can help determining if a given person is approaching some specific part of the room (i.e. the door) so that a convenient action can be taken. Tracking need not to be continuously and people can be out of sight of camera's and/ or microphone. The challenge is to still have a good knowledge about the whereabouts.

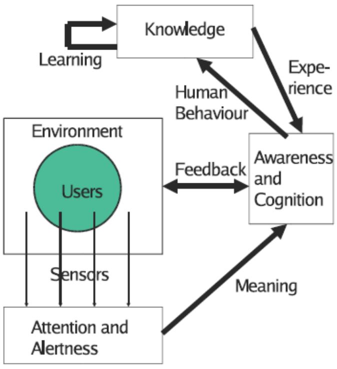


Figure 1: *The conceptual design of a human-centered smart environment (From [31]).*

There are many possible applications of such an approach. Tracking babies, kids, or elderly people for particular events, intrusion detection, gesture or speech based controlling of environmental parameters (e.g. lights, audio volume of the TV set, etc.) can be implemented. The aim of the project is to implement the tools as black-box modules that would allow straightforward application to flexible scenarios. Some possible technologies needed to fulfill these goals are motion detection (visual), person identification (visual, acoustic), tracking (visual, acoustic), speech detection, head pose estimation (visual, acoustic), gesture recognition (visual) and gait analysis.

This paper is organized as follows. In Section 2, we describe the setup of the room, and the low-cost sensors we have employed. We also briefly describe the SmartFlow middleware developed by NIST for cross-platform sensor communication in Section 2.3. Section 3 describes the separate software modules we have implemented and interfaced to SmartFlow. A discussion and our conclusions follow in Section 4.

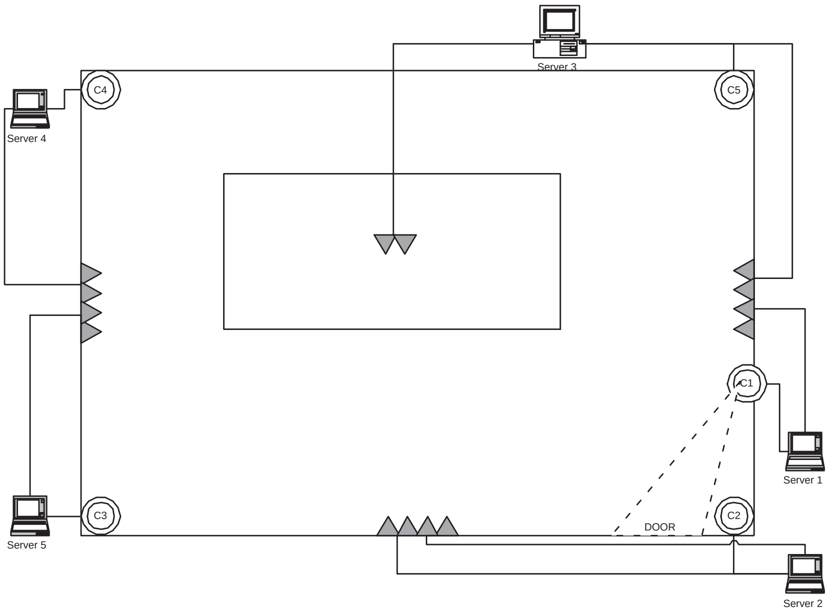


Figure 2: *The design of the room. The ceiling cameras are shown as circles at the corners of the room, and one camera that is roughly placed at eye-level is facing the door. The microphones are placed on the walls and on the table in the centre of the room.*

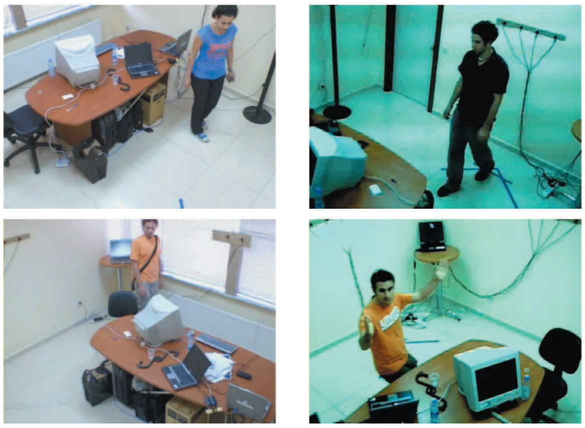


Figure 3: *Sample recordings from the ceiling cameras. The illumination conditions and quality of images are very different, also because the cameras were of different models.*

2. THE BU SMART ROOM

The BU smart room is established for the whole duration of the Enterface Workshop in Boğaziçi University. Since there is no sensor calibration, the actual installation was fast. We have used five cameras and fourteen microphones. The sensor equipment is low-cost, but five computers are used to drive the sensors in the absence of dedicated cheaper hardware. The sensors are connected to the computers via USB interface, and the computers are connected to an Ethernet switch for communication.

The design of our smart room is given in Fig. 1. There are four ceiling cameras in the corners of the room, and one camera facing the door. The microphones are placed in three groups of four microphones on three walls, plus a group of two microphones in the middle of the room. The room has windows on one side, which change the illumination conditions for each of the cameras. The ceiling illumination is fluorescent, and there are reflections from the ground tiles. The focus and resolutions of the cameras are minimally adjusted.

2.1. Visual Sensors

We have used two Philips 900 NC cameras, two Philips 700 NC cameras, and a Logitech UltraVision camera. The Philips cameras are placed at the ceiling corners, and the Logitech camera faces the door. All cameras have narrow angles of view, and this equipment costs less than 400$. Fig. 3 shows images acquired from these cameras at a given time. As it is evident from these images, the illumination, saturation and resolution properties are not consistent across the cameras.

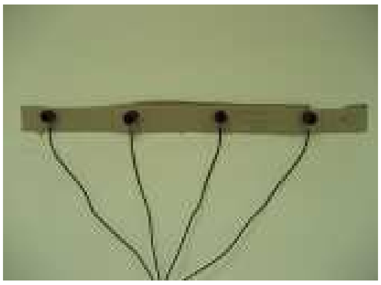


Figure 4: *The home-made low-cost microphone array.*

2.2. Audio Sensors

For the audio equipment, 14 cheap USB microphones were taken, and their protective casings were removed. The microphone arrays are constructed by attaching the stripped microphones to cardboard panels with equal spacing. Fig. 4 shows one such microphone array. The presence of five computers in the room and the large amount of ambient noise makes the audio system very challenging.

2.3. The SmartFlow Middleware

The SmartFlow system, developed by NIST, is a middleware that allows the transportation of large amounts of data from sensors to recognition algorithms running on distributed, networked nodes [20, 23]. The working installations of SmartFlow is reportedly able to support hundreds of sensors [28].

The SmartFlow provides the user with a way of packaging algorithms into processing blocks, where the data are pictured as a distributed flow among these blocks. Sensor data are captured by clients, cast into a standard format, and a flow for each data stream is initiated. The processing blocks are themselves SmartFlow clients, and they can subscribe to one or more flows to receive the data required for their processing. Each client can output one or more flows for the benefit of other clients.

The design of a working system is realized through a graphical user interface, where clients are depicted as blocks and flows as connections. The user can drag and drop client blocks onto a map, connect the clients via flows, and activate these processing blocks.

There are two versions of SmartFlow. Version 1 is only usable under Unix OS, whereas Version 2 is usable with Windows OS. Our final system contains components that were developed under Linux and Windows, therefore we chose Version 2 as our platform. However, some issues are not completely solved in Version 2, and designs with many components suffer.

The synchronization of the clients is achieved by synchronizing the time for each driving computer, and timestamping the flows. The network time protocol (NTP) is used to synchronize the clients with the server time, and this functionality is provided by SmartFlow. A separate client is used to start the processing clients simultaneously. The video streams are not completely in one-to-one correspondence, as clients sometimes drop frames.

2.4. Data Collection

We have used three different datasets to train the speech/non-speech classes. In addition to the RT05 and RT06 datasets [33] a new dataset was collected for adaptation to the BU smart room acoustics. This dataset contains segments of silence (i.e. ambient noise) and a host of non-silence events. Non-silence events include sounds of chairs moving, keyboard typing, claps, coughs, switching on/off the light, knocking on the door, laughs, and steps. A total of 10 minutes of 14 different microphone streams were recorded, leading to a total of $840,000$ frames at a rate of 10ms. for each of the speech/non-speech classes.

For person localization, we have recorded a long sequence of frames (about eight minutes) from the four ceiling cameras simultaneously. During this sequence, the group members entered the room one by one, and walked on a pre-determined path through the room, visiting each location and stopping briefly on marker areas (centres of floor tiles). While walking, the members were asked to talk continuously to provide ground truth data for the audio based localization system, as well as the speaker identification system.

For gesture recognition, we have collaborated with Enterface Group 12, and a dataset of 90 persons was collected through their efforts. Each session contains a hand gesture part, where four gestures are repeated three times in sequence. These gestures are typically associated with turn-on (clacking the finger), turn off (cutting off the head), volume up (increase indication) and volume down (decrease indication) events. This dataset also contains sequences where the head was moved from right to left during ordinary speech.

3. SOFTWARE MODULES

3.1. Motion Detection Module

The motion detection module attempts to separate the foreground from the background for its operation. Foreground detection using background modeling is a common computer vision task particularly in the field of surveillance [29]. The method is based on detecting moving objects under the assumption that images of a scene without moving object show regular behavior which can be modeled using statistical methods.

In a practical environment like our smart room illumination could change according to the need of the user or it could also change due to gradual sun lighting change. In order to adapt changes we can update the training set by adding new samples. Each pixel observation consist of color measurement. At any time, t, pixel value at pixel i can be written as $X_{i,t} = [R_{i,t}, G_{i,t}, B_{i,t}]$.

The recent history of every pixel within an image is stacked which can be represented as $[X_{i,1}, X_{i,2}, ..., X_{i,t-1}]$ and is modeled as a set of Gaussian distributions. Now the probability of the current observation at a pixel i can be estimated using the model built from previous observations.

$$P(X_{i,t}|X_{i,1},...,X_{i,t-1}) = \sum w_{i,t-1} * \eta(X_{i,t}, \mu_{i,t-1}, \sigma^2_{i,t-1}) \tag{1}$$

where η is the Gaussian probability density function. $\mu_{i,t-1}$ and $\sigma^2_{i,t-1}$ are mean and covariance matrix of the Gaussian. $w_{i,t-1}$ is the weight associated with the Gaussian. To make the process on-line, a matching process is carried out; a new pixel is considered to be background if it matches with the current Gaussian component, i.e. if the distance between the pixel and the mean of the Gaussian in question is less than ϵ. In this study we have chosen $\epsilon = 2 * \sigma$. If a current pixel doesn't match the mean of the given distribution, then the parameters of the distribution are updated with a higher weight, otherwise it is updated with a lower weight $w_(i,t)$. The adaptation procedure is as follows:

$$w_{i,t} = (1-\alpha)w_{i,t-1} + \alpha M_{i,t} \tag{2}$$

where α is learning rate, $\alpha \in [0,1]$ and $1/\alpha$ determines the speed of the adaptation process. And $M_{i,t} = 1$ if the current pixel matches a model, otherwise it is 0 for rest of the models. In a similar vein μ and σ are updated as follows:

$$\mu_{i,t} = (1-\lambda)\mu_{i,t-1} + \lambda X_{i,t} \tag{3}$$

$$\sigma^2_{i,t} = (1-\lambda)\sigma^2_{i,t-1} + \lambda(X_{i,t} - \mu_{i,t})^T(X_{i,t} - \mu_{i,t}) \tag{4}$$

where

$$\lambda = \alpha\eta(X_t|\mu_{i,t-1}, \sigma_{i,t-1}) \tag{5}$$

One significant advantage of this technique is, as the new values are allowed to be part of the model, the old model is not completely discarded. If the new values become prominent overtime, the weighting changes accordingly, and new values tend to have more weight as older values become less important. Thus, if there is a movement of furniture with the room, the background model is updated rather quickly; and the same is true for lighting changes.

Fig. 5 shows a sample frame and the detected moving foreground.

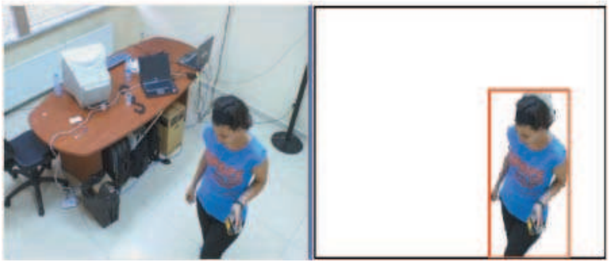


Figure 5: *A frame with motion, and the detected moving foreground.*

3.2. Face Detection Module

Face detection is needed for face identification and opportunistic sensing modules. In this module, the face of each person present in the scene must be detected roughly (i.e. a bounding box around a face will be the output of this module). We use the OpenCV face detection module that relies on the adaboosted cascade of Haar features, i.e. the Viola-Jones algorithm [35]. The client that performs face detection receives a video flow from a client that in its turn directly receives its input from one of the cameras, and outputs a flow that contains the bounding box of the detected face.

3.3. Face Identification Module

The face recognition module is semi-automatic, in that it takes motion tracking and face detection for granted. This module therefore subscribes to the face detection flow that indicates face locations, and to the video flow to analyze the visual input to a camera. A technique for face recognition in smart environments [16, 34] is used. The technique takes advantage of the continuous monitoring of the environment and combines the information of several images to perform the recognition. Models for all individuals in the database are created off-line.

The system works with groups of face images of the same individual. For each test segment, face images of the same individual are gathered into a group. Then, for each group, the system compares these images with the model of the person. We first describe the procedure for combining the information provided by a face recognition algorithm when it is applied to a group of images of the same person in order to, globally, improve the recognition results. Note that this approach is independent of the distance measure adopted by the specific face recognition algorithm.

3.3.1. Combining groups of images

Let $\{x\}_i = \{x_1, x_2, ..., x_P\}$ be a group of P probe images of the same person, and let $\{C\}_j = \{C_1, C_2, ..., C_S\}$ be the different models or classes stored in the (local or global) model database. S is the number of individual models. Each model C_j contains N_j images, $\{y_n\}^j = \{{y_1}^j, {y_2}^j, ..., {y_{N_j}}^j\}$ where N_j may be different for every class. Moreover, let

$$d(x_i, {y_n}^j) : \mathcal{R}^Q x \mathcal{R}^Q \quad \rightarrow \quad \mathcal{R} \tag{6}$$

be a certain decision function that applies to one element of $\{x\}_i$ and one element of $\{y\}_n^j$, where Q is the dimension of x_i and y_n^j. It represents the decision function of any face recognition algorithm. It measures the similarity of a probe image x_i to a test image y_n^j. We fix a decision threshold R_d so that x_i and y_n^j represent the same person if $d(x_i, y_n^j) < R_d$. If, for a given x_i the decision function is applied to every $y_n^j \in C_j$, we can define the δ value of x_i relative to a class C_j, δ_{ij} as

$$\delta_{ij} = \#\{y_n^j \in C_j | d(x_i, y_n^j) < R_d\} \tag{7}$$

That is, δ_{ij} counts the number of times that the face recognition algorithm matches x_i with an element of C_j. With this information, the δ-Table is built, and based on this table we define the following concepts:

- Individual Representation of x_i: It measures the representation of sample x_i by class C_j:

$$R(x_i, C_j) = \frac{\delta_{ij}}{N_j} \tag{8}$$

- Total representation of x_i: It is the sum of the individual representations of x_i through all the classes:

$$R(x_i) = \frac{1}{P} \sum_{j=1}^{S} R(x_i, C_j) = \sum_{j=1}^{S} \frac{\delta_{ij}}{N_j} \tag{9}$$

- Reliability of a sample x_i given a class C_j: It measures the relative representation of sample x_i by class C_j considering that sample x_i could be represented by other classes:

$$\rho(x_i, C_j) = \begin{cases} \frac{R(x_i, C_j)}{R(x_i)} = \frac{\delta_{ij}/N_j}{\sum_{k=1}^{S} \frac{\delta_{ik}}{N_k}} & R(x_i) > 0 \\ 1 & R(x_i) = 0 \end{cases} \leq 1$$

- Representation of C_j: It estimates the relative representation of a group of samples $\{x\}_i$ by a class C_j. Weighting is performed to account for the contribution of the group $\{x\}_i$ to other classes:

$$R(C_j) = \frac{1}{P} \sum_{i=1}^{P} \rho_{ij} \delta_{ij} \tag{10}$$

- Match Likelihood M for class C_j: It relates a class representation and its match probability. If $r = R(C_j)$, then:

$$M(C_j) = \frac{1 - e^{\frac{-r^2}{\sigma^2}}}{1 - e^{\frac{-N_j^2}{\sigma^2}}} \tag{11}$$

  where σ adjusts the range of $R(C_j)$ values.

- Relative Match Likelihood for a class C_j: It relates the M of a class C_j and the maximum M of the other classes:

$$RML(C_j) = \begin{cases} \frac{M(C_j)}{\max_{k \neq j}(M(C_k))} & M(C_j) \geq 0.5 \\ 0 & M(C_j) < 0.5 \end{cases} \tag{12}$$

  This measure determines if the selected class (that with the maximum M) is widely separated from other classes. A minimum value of M is required, to avoid analyzing cases with too low M values.

Relying on the previous concepts, the recognition process is defined, in the identification mode, as follows:

- Compute the δ-Table.
- Compute the match likelihood M for every model.
- Compute the RML of the class with the highest $M(C_j)$.

The group is assigned to the class resulting in a highest RML value In this work, a PCA based approach [14] has been used. This way, the decision function is the Euclidean distance between the projections of x_i and y_n^j on the subspace spanned by the first eigenvectors of the training data covariance matrix:

$$d(x_i, y_n^j) = ||W^T x_i - W^T y_n^j|| \tag{13}$$

where W^T is the projection matrix.

The XM2VTS database [18] has been used as training data for estimating the projection matrix and the first 400 eigenvectors have been preserved. Only frontal faces are used for identification. Note that, in our system, models per each person have been automatically generated, without human intervention. All images for a given individual in the training intervals are candidates to form part of the model. Candidate face bounding boxes are projected on the subspace spanned by the first eigenvectors of the training data covariance matrix W^T. The resulting vector is added to the model only if different enough from the vectors already present in the model.

3.4. Opportunistic Sensing Module

The opportunistic sensing module aims at identifying persons in the room when the face information is not available, or not discriminatory. The primary assumption behind the operation of this module is that the variability in a users appearance for a single camera is relatively low for a single session (this case is termed intra session in [34]), and a user model created on-the-fly can provide us with useful information [32]. We use the following general procedure for this purpose: Whenever the face identification module returns a reliable face identification, cameras with access to the area with the detected face consult the motion detection module, and grab a window from the heart of the motion blob. The pixel intensities within this window are modeled statistically, and this statistical model is then used to produce the likelihood values for every candidate person for which the system stored a mixture model.

The general expression for a *mixture model* is written as

$$p(\boldsymbol{x}) = \sum_{j=1}^{J} p(\boldsymbol{x}|\mathcal{G}_j)P(\mathcal{G}_j) \tag{14}$$

where $\mathcal{G}_j$ stand for the components, $P(\mathcal{G}_j)$ is the prior probability, and $p(\boldsymbol{x}|\mathcal{G}_j)$ is the probability that the data point is generated by component j. In a *mixture of Gaussians* (MoG), the components in Eq. 14 are Gaussian distributions:

$$p(x|\mathcal{G}_j) \sim \mathcal{N}(\mu_j, \boldsymbol{\Sigma}_j) \tag{15}$$

In a MoG, the number of parameters determine the complexity of the model, and the number of training samples required for robust training increases proportionally to the complexity of the model. Introducing more components means that the ensuing mixture will be more complex, and more difficult to train on the one hand, but potentially more powerful in explaining the data, as complex models usually go.

One way of controlling the complexity is to state some assumptions regarding the shape of the covariance matrix. A complete covariance matrix $\boldsymbol{\Sigma}_j$ for a d-dimensional data set has $O(d^2)$ parameters, and specifies a flexible and powerful statistical model. Learning a sample covariance of this shape often creates serious overfitting issues and singularities due to limited training data. Frequently, a diagonal covariance matrix is adopted, which means the covariances between dimensions are discarded, and only the variances are modeled.

To build our statistical models for the users of the smart room, we use a *factor analysis* (FA) approach, which represents a trade-off between model complexity and modeling covariances. In FA, the high dimensional data $\boldsymbol{x}$ are assumed to be generated in a low-dimensional manifold, represented by latent variables $\boldsymbol{z}$. The *factor space* spanned by the latent variables is similar to the principal space in the PCA method, and the relationship is characterized by a *factor loading matrix* Λ, and independent Gaussian noise $\boldsymbol{\epsilon}$:

$$\boldsymbol{x} - \boldsymbol{\mu}_j = \boldsymbol{\Lambda}_j \boldsymbol{z} + \boldsymbol{\epsilon}_j \tag{16}$$

The covariance matrix in the d-dimensional space is then represented by $\boldsymbol{\Sigma}_j = \boldsymbol{\Lambda}_j\boldsymbol{\Lambda}_j^T + \Psi$, where Ψ is a diagonal matrix and $\boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \Psi)$ is the Gaussian noise. We obtain a *mixture of factor analysers* (MoFA) by replacing the Gaussian distribution in Eq. 14 with its FA formulation.

In the opportunistic sensing module, the surface features are modeled with mixture models. A single frame is used to produce a training set for the MoFA model. If the the number of components and the number of factors per component are specified beforehand, the maximum likelihood parameters can be computed with the Expectation-Maximization (EM) algorithm [9]. With no prior information, an incremental algorithm can be used to determine model parameters automatically [26]. The incremental MoFA algorithm (IMoFA) starts with a one-factor, one-component mixture and adds new factors and new components until the likelihood is no longer increased on the validation set.

For component addition, a multivariate kurtosis-based measure is adopted to select components that look least like unimodal Gaussians:

$$\gamma_j = \{b^j_{2,d} - d(d+2)\}\left[\frac{8d(d+2)}{\sum_{t=1}^{N} h_j^t}\right]^{-\frac{1}{2}} \tag{17}$$

$$b^j_{2,d} = \frac{1}{\sum_{l=1}^{N} h_j^l}\sum_{t=1}^{N} h_j^t\left[(\boldsymbol{x}^t - \boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}^t - \boldsymbol{\mu}_j)\right]^2 \tag{18}$$

with $h_j^t \equiv E[\mathcal{G}_j|\boldsymbol{x}^t]$, and the component with greatest γ_j is selected for splitting.

For factor addition the difference between sample covariance and modeled covariance is monitored and the component with the largest difference is selected for factor addition. The new factor is the principal axis of the residual error vectors.

There are currently no accuracy results for the opportunistic sensing module, since the recorded data is not annotated with ground truth. However, visual inspection shows that this model is successful under two conditions:

- The training frame should be discriminative.
- The overall image intensity conditions during operation should be consistent with the training conditions.

In Fig. 6 three sample training frames are shown. To ensure fast operation, only a portion of these frames is used for training. 5×5 pixels with RGB values are concatenated to form $75-$dimensional vectors, which are then modeled with IMoFA mixtures.

Fig. 9 shows the detection results for each of the three people in the system. Since we use generative models, an increase in the number of subjects does not decrease the localization accuracy of the system, but considering additional candidates brings a computational cost. The client that performs likelihood computation pre-loads the mixture parameters, computes inverse covariance and determinant parameters, and waits for data packets. The real-time operation is ensured by dropping data frames.

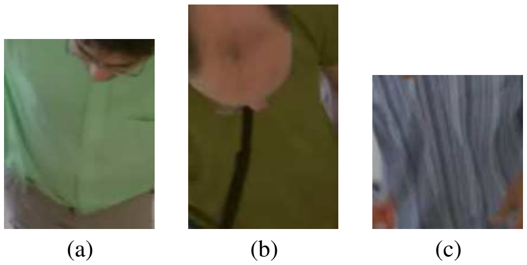


Figure 6: *Training frames for a) Albert, b) Ramon, c) Onkar.*

Fig. 7 shows the effect of a non-discriminative frame. The presence of generic intensities results in a too general mixture, and many data packets produce high likelihoods.

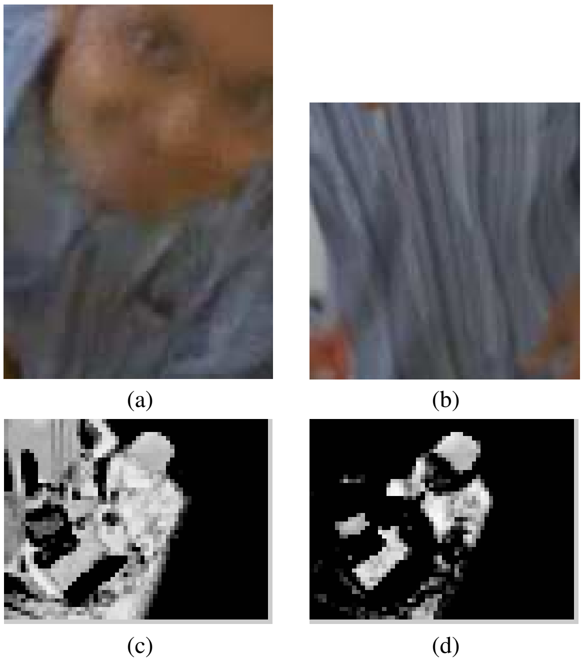


Figure 7: *a) Non-discriminative training frame for class Onkar. b) A more discriminative training frame for class Onkar. c) Detection of Onkar results in many false-positives with the non-discriminative training frame. d) The discriminative frame works fine.*

3.5. Speaker Identification Module

The Speaker ID module (SID) segments the audio data, identifying the silences and the speaker changes. Furthermore, the algorithm is able to identify different speakers with the help of prior knowledge. With this purpose, a small database of ten persons was collected in the BU smart room. The speaker models are computed off-line.

The speaker identification system is composed of several modules developed in the SmartFlow framework, and it can be inspected in three groups. The Speech Activity Detection (SAD) is in charge of detecting the speech activity in the room and discriminating it from non-speech events. The Acoustic SEgmentation (ASE) module retrieves speech information from the SAD and provides the Speaker IDentification (SID) stage with homogenous acoustic data from a single identity. All three stages work in parallel, continuously computing the high level information, speech/non-speech segmentation, acoustic change detection and frame-score accumulation for the identity labels, thereby monitoring the room constantly in real time. However, an identity label is only provided in the case that the acoustical information is sufficient, and homogenously collected from a single person, and the speech data is sufficient for discrimination.

Fig. 8 depicts the SmartFlow map employed in the SID implementation. The frequency downsampling stage produces acoustic signals at a rate of 16.000 KHz for the remaining stages, for a total of 14 microphones. The flow connections indicate the feedback and interactions between the three main modules.

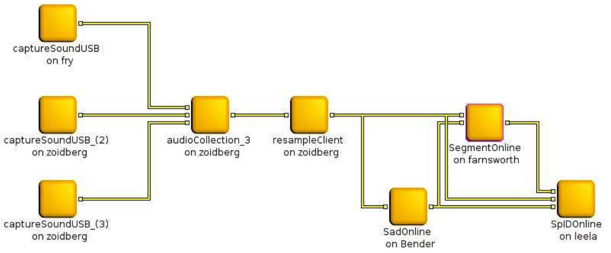

captureSoundUSB on fry
captureSoundUSB_(2) on zoidberg
captureSoundUSB_(3) on zoidberg
audioCollection_3 on zoidberg
resampleClient on zoidberg
SegmentOnline on farnsworth
SadOnline on Bender
SpIDOnline on leela

Figure 8: *Smarflow Map implementation of the SID module. The first stages are related to capturing signal modules and downsampling. The SelectChannel box gathers all the input channels coming from different machines and selects one of them based on a simple energy threshold decision.*

3.5.1. Speech Activity Detection

The SAD module used in this work is based on a support vector machine (SVM) classifier [27]. The performance of this system was shown to be good in the 2006 Rich Transcription SAD Evaluations (RT06) [33]. A GMM-based system that ranked among the best systems in the RT06 evaluation was selected as a baseline for performance comparison [33].

For classical audio and speech processing techniques that involve GMMs, the training data are in the order of several hundred thousand examples. However, SVM training usually involves far less training samples, as the benefit of additional samples is marginal, and the training becomes infeasible under too large training sets. Some effective methods should be applied to reduce the amount of data for training without losing accuracy. Therefore, we employ Proximal SVMs (PSVM) in the same way as proposed in [33].

Proximal Support Vector Machine (PSVM) has been recently introduced in [8] as a result of the substitution of the inequality constraint of a classical SVM $y_i(wx_i + b) \geq 1$ by the equality constraint $y_i(wx_i + b) = 1$, where y_i stands for a label of a vector x_i, w is the norm of the separating hyperplane H_0, and b is the scalar bias of the hyperplane H_0. This simple modification significantly changes the nature of the optimization problem. Unlike conventional SVM, PSVM solves a single square system of linear equations and thus it is very fast to train. As a consequence, it turns out that it is possible to obtain an explicit exact solution to the optimization problem [8].

Depending on the speech activity inside the room, we should penalize the errors from the Speech class more than those from the Non-Speech class, in the case of a meeting for example. It is possible to selectively penalize errors for the SVM in the training stage by introducing different costs for the errors in two classes by introducing different generalization parameters C_- and C_+. This approach will adjust the separating hyperplane H_0, and it will no longer be exactly in the middle of the H_{-1} and H_1 hyperplanes (Fig. 9). It is worth mentioning that favouring a class in the testing stage (after the classifier is trained) is still possible for SVM through the bias b of the separating hyperplane.

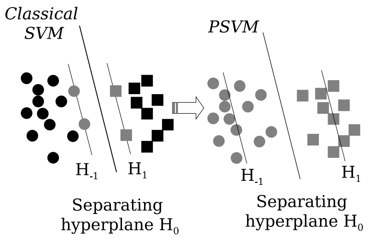


Figure 9: *Proximal Support Vector Machine based SVM.*

3.5.2. Bayesian Information Criterion based Segmentation

Audio segmentation, sometimes referred to as acoustic change detection, consists of exploring an audio file to find acoustically homogeneous segments, detecting any change of speaker, background or channel conditions. It is a pattern recognition problem, since it strives to find the most likely categorization of a sequence of acoustic observations. Audio segmentation becomes useful as a pre-processing step in order to transcribe the speech content in broadcast news and meetings, because regions of different nature can be handled in a different way.

The ALIZE Toolkit was used in this work to implement a XBIC-based segmentation system, making use of the algorithms for the GMM data modeling and the estimation of the parameters. The ALIZE toolkit is a free open tool for speaker recognition from the LIA, Université d'Avignon [1]. The "Bayesian Information Criterion" (BIC) is a popular method that is able to perform speaker segmentation in real time.

Let the set $\Theta = \{\theta_j \in \Re^d \mid j \in 1 \ldots N\}$, a sequence of N parameter vectors, the BIC is defined as:

$$BIC_\Theta = L - \alpha P \tag{19}$$

where P is the penalty and α a free design parameter. L is the logarithm of the probability performed by the set of observations Θ over the model λ_i,

$$L = P(\Theta|\lambda_i) = \sum_{k=1}^{N} log(\theta_k|\lambda_i) \tag{20}$$

Let an instant $\theta_j \in \Theta$, it can be defined two partitions of Θ: $\Theta_1 = \{\theta_1 \ldots \theta_j\}$ and $\Theta_2 = \{\theta_{j+1} \ldots \theta_N\}$ with length N_1 and N_2. In order to take a decision about a change in the speakers, two different hypotheses can be considered:

- H_0: An unique class λ is better at modeling the data from the whole set of observations Θ.
- H_1: Two independent classes λ_1 and λ_2 are better in jointly modeling the before and after of the postulated instance of speaker change θ_j.

The best hypothesis is chosen by evaluating the following expression:

$$\Delta BIC = BIC_{H_0} - BIC_{H_1} = BIC_\Theta - (BIC_{\Theta_1} + BIC_{\Theta_2}) \tag{21}$$

Therefore, and looking at Eqs. 19 and 20 the criterion of change speaker turn at time i can be defined by evaluating the following expression:

$$\Delta BIC(i) = P(\Theta|\lambda) - P(\Theta_1|\lambda_1) - P(\Theta_2|\lambda_2) - \alpha P \tag{22}$$

where the penalty term depends on the number of parameters (complexity) employed to estimate the whole model λ and the two sub-models λ_1,λ_2. This constant is set to a low value, since it only affects the decision threshold around 0.

In overall, a speaker change turn is decided at the time instant i for which $\Delta BIC(i) > 0$, which means that the dual model is now better adapted to the observation data in comparison to the single model. Taking into account the definition of the Rabiner distance, the distance among two Hidden Markov Models is

$$D_{rab} = \frac{D(\lambda_a, \lambda_b) + D(\lambda_a, \lambda_b)}{2} \tag{23}$$

with D as,

$$D(\lambda_a, \lambda_b) = \frac{1}{N_b}\left(\sum_{k=1}^{N_b} log\, p(\theta_k|\lambda_a) - \sum_{k=1}^{N_a} log\, p(\theta_k|\lambda_b)\right) \tag{24}$$

Assuming that the two data segments have the same duration, and sorting the terms of the Eq. 23, we obtain a probabilistic definition of the Rabiner distance,

$$D'_{rab} = \big(P(\Theta_1|\lambda_2) + P(\Theta_2|\lambda_1)\big) - \big(P(\Theta_1|\lambda_1) + P(\Theta_2|\lambda_2)\big) \tag{25}$$

where $P(\Theta_i|\lambda_j)$ is defined in Eq. 20. Eq. 25 is similar to the one presented in the BIC procedure 19, but using a second shared term: $P(\Theta_1|\lambda_1) + P(\Theta_2|\lambda_2)$, which changes as the models adapt to the training set.

The first term of both equations, Eq. 19 and Eq. 25, measure how well the whole audio segment is adapted to a single model or to two models, respectively. In both cases a high probability is obtained if the segment belongs to the same speaker.

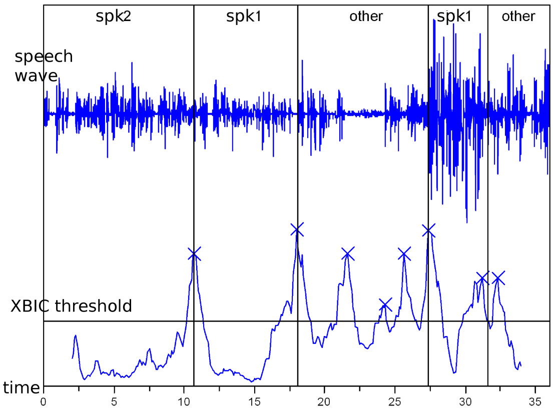


Figure 10: *Segmentation example.*

In the case of the Eq. 25, when two acoustically similar segments are evaluated, the measure oscillates around the 0 value. When a speaker change occurs, the XBIC distance peaks, mostly due to the cross probability term. By this reason the XBIC measure is defined as:

$$XBIC(i) = P(\Theta_1|\lambda_2) + P(\Theta_2|\lambda_1) \tag{26}$$

If a suitable threshold is chosen, the speaker change can be determined at the time instant i as $XBIC(i) <$ threshold$_{XBIC}$

3.5.3. Speaker Identification

The Person Identification (PID) problem consists of recognizing a particular speaker from a segment of speech spoken by a single speaker. Matched training and testing conditions and far-field data acquisition are assumed, as well as a limited amount of training data and no a priori knowledge about the room environment. The algorithm implemented in this work was tested in the CLEAR'07 PID evaluation campaign, obtaining the best position in mean in all test/train conditions [17].

The algorithm commences by processing each data window by subtracting the mean amplitude, supposing the DC offset is constant throughout the waveform. A Hamming window was applied to each frame and a FFT is computed. The FFT amplitudes are then averaged in 30 overlapped triangular filters, with central frequencies and bandwidths defined according to the Mel scale. Their output represents the spectral magnitude in that filter-bank channel. Instead of the using Discrete Cosine Transform, such as in the Mel-Frequency Cepstral Coefficients (MFCC) procedure [5], the samples are parametrised using the Frequency Filtering (FF):

$$H(z) = z - z^{-1} \tag{27}$$

over the log of the filter-bank energies. Finally, we obtain 30 FF coefficients. A vector of parameters is calculated from each speech segment. The choice of this setting was justified through experimental validation, showing that FF coefficients result in higher accuracies than MFCC coefficients for both speech and speaker recognition tasks, as well as being more robust against noise and computationally more efficient [19]. Additionally, the FF coefficients are uncorrelated, and they have an intuitive interpretation in the frequency domain.

In order to capture the temporal evolution of the parameters, the first and second time derivatives of the features are appended to the basic static feature vector. The so-called Δ and Δ-Δ coefficients are also used in this work. Note that the first coefficient of the FF output, which is proportional to the signal energy, is also employed to compute the model estimation, as well as its velocity and acceleration parameters. Next, for each speaker that the system has to recognize, a model of the probability density function of the parameter vectors is estimated. Gaussian Mixture Models with diagonal covariance matrices are employed, and the number of components is set to 64 for each mixture. The large amount of components assure that the statistical learning is robust, and is further justified by the availability of a very large training set.

The parameters of the models are estimated from speech samples of the speakers using the well-known Baum-Welch algorithm [24]. Given a collection of training vectors, the maximum likelihood model parameters are estimated using the iterative Expectation-Maximisation (EM) algorithm. The sensitivity of EM to cases with few training data is well-known, yet under the present training conditions (with ample data), 10 iterations are demonstrably enough for parameter convergence. This parameter is retained for both training conditions and for all the client models.

In the testing phase of the speaker identification system, a set of parameters $\mathbf{O} = \{\mathbf{o}_i\}$ is computed from the speech signal. Next, the likelihood of obtaining the signal under each client model $\mathbf{O}$ is calculated and the speaker model with the largest posterior probability is chosen,

$$s = \arg\max_j \left\{ L\big(\mathbf{O}|\lambda_j\big) \right\} \tag{28}$$

where s is the recognised speaker, and L is the likelihood function from a linear combination of M unimodal Gaussians of dimension D [3]. Therefore, $L\big(\mathbf{O}|\lambda_j\big)$ is the likelihood that the vector $\mathbf{O}$ has generated by the speaker model λ_j.

3.6. Sound Based Localization Module

Conventional acoustic person localization and tracking systems usually consist of three basic stages. In the first stage, estimation of such information as Time Difference of Arrival or Direction of Arrival is usually computed by the combination of data coming from different microphones. The second stage involves a set of relative delays or directions of arrival estimations to derive the source position that agrees most with the data streams and with the given geometry. In the third (and optional) stage, possible movements of the sources can be tracked according to a motion model. These techniques need several synchronized high-quality microphones. In the BU Smart Room setting we have only access to low-quality, uncalibrated and unsynchronized microphones. As explained in the room setup description, only pairs of microphones within an audio capture device are synchronized, there is no synchronization across different capture devices. Moreover, the channels within a single capture device have a highly correlated noise of the same power level as the recorded voice, thus estimating the position of speakers in this environment is a very difficult and challenging task.

The acoustic localization system used in this project is based on the SRP-PHAT localization method, which is known to perform robustly in most scenarios [6]. The SRP-PHAT algorithm (also known as Global Coherence Field) tackles the task of acoustic localization in a robust and efficient way. In general, the basic operation of localization techniques based on SRP is to search the room space for a maximum in the power of the received sound source signal using a delay-and-sum or a filter-and-sum beamformer. In the simplest case, the output of the delay-and-sum beamformer is the sum of the signals of each microphone with the adequate steering delays for the position that is explored. Concretely, the SRP-PHAT algorithms consists in exploring the 3D space, searching for the maximum of the contribution of the PHAT-weighted cross-correlations between all the microphone pairs. The SRP-PHAT algorithm performs very robustly due the the PHAT weighting, keeping the simplicity of the steered beamformer approach.

Consider a smart-room provided with a set of N microphones from which we choose M microphone pairs. Let $\mathbf{x}$ denote a $\mathbf{R}^3$ position in space. Then the time delay of arrival $TDOA_{i,j}$ of an hypothetic acoustic source located at $\mathbf{x}$ between two microphones i, j with position $\mathbf{m}_i$ and $\mathbf{m}_j$ is:

$$TDOA_{i,j} = \frac{\parallel \mathbf{x} - \mathbf{m}_i \parallel - \parallel \mathbf{x} - \mathbf{m}_j \parallel}{s}, \tag{29}$$

where s is the speed of sound.

The 3D room space is then quantized into a set of positions with typical separations of 5-10cm. The theoretical TDOA $\tau_{\mathbf{x},i,j}$ from each exploration position to each microphone pair are precalculated and stored.

PHAT-weighted cross-correlations of each microphone pair are estimated for each analysis frame [21]. They can be expressed in terms of the inverse Fourier transform of the estimated cross-power spectral density ($G_{m_1 m_2}(f)$) as follows:

$$R_{m_i m_j}(\tau) = \int_{-\infty}^{\infty} \frac{G_{m_i m_j}(f)}{|G_{m_i m_j}(f)|} e^{j2\pi f \tau} df, \tag{30}$$

The estimated acoustic source location is the position of the quantized space that maximizes the contribution of the cross-correlation of all microphone pairs:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{i,j \,\in\, \mathbb{S}} R_{m_i m_j}(\tau_{\mathbf{x},i,j}), \tag{31}$$

where $\mathbb{S}$ is the set of microphone pairs. The sum of the contributions of each microphone pair cross-correlation gives a value of confidence of the estimated position, which can be used in conjunction with a threshold to detect acoustic activity and to filter out noise. In our work, we use a threshold of 0.5 per cluster of 4 microphones. It is important to note that in the case of concurrent speakers or acoustic events, this technique will only provide an estimation for the dominant acoustic source at each iteration.

3.7. Gesture Tracking Module

Hand gestures used in human–computer interaction (HCI) systems are either communicative or manipulative. Manipulative hand gestures are used to act on virtual objects or parameters, whereas communicative gestures are combination of dynamic acts and static symbols that have communication purposes. HCI applications often focus on a subset of these classes of gestures and consider only these classes as meaningful. The hand gesture recognition module we have designed for the smart room application is aimed to use both communicative and manipulative gestures for HCI and to automatically distinguish these in real time. This module is based on the work presented in [11, 12, 13].

The gesture recognition module is the main channel of interaction with between the users and the computers in BU Smart Room, since it allows the users to give commands to the underlying smart system. The gesture recognition system our module is based on, handles the event recognition problem using two primary methods:

- by manipulative gestures, i.e. by allowing direct translation of the hand trajectory and shape into continuous events, such as manipulation of virtual objects, or
- by communicative gestures, i.e. by recognizing certain patterns performed by the user and firing high level discrete events.

In order to make use of both of these methodologies, we have designed a simple extendible gesture language that consists of actions and objects. The structure of each command consists of a combination of one action and one object, the former corresponding to a certain pattern and the latter to a simple hand movement, specifically in this order to prevent false positives. The reasoning behind this is based on the fact that communicative gestures are chosen in a way that they are unlikely to be performed unintentionally, wheras the manipulative ones may correspond to very common hand movements. Since we restrict the objects to follow actions, the intentional manipulative gestures corresponding to the objects can unambiguously be recognized.

Gestures are performed by the flexible hands in 3D space. The motion and change in the shape of the hand have different characteristics. Thus, they are often modeled separately and considered to be different modalities. Both appearance based and 3D model based approaches have been pursued for computer vision based hand shape modeling. 3D models are complex geometrical models using a priori knowledge about hand geometry, while appearance based models use features extracted from the projected intensity image of the hand. In general, most of the computer vision techniques extract the silhouettes of the hands first.

To simplify the hand segmentation problem, either simple backgrounds are assumed, or colored markers are used. Since hands are homogeneously colored, hand tracking can be accomplished using color–based features. However, other skin colored regions such as the face make this task difficult and time consuming. Due to real time execution considerations, markers are used in our system to detect the hand in each image. The choice of color for the marker is not predefined, but is restricted to colors that are significantly different than the skin color.

In the case of smart rooms, simple backgrounds can not be assumed. By employing colored markers, it is possible to robustly detect and track the hands by using simple color based detection algorithms. We have developed two different techniques for marker detection in BU Smart Room, which basically make use of different color systems, namely RGB and HSV. While the RGB–based method uses either fixed or floating ranges to connect neighboring colors for segmentation, the HSV–based method calculates the Hue–Saturation histogram of the marker and uses it as a signature. These methods will be explained in more detail in the following section.

The choice of the hand features extracted from the hand region primarily depends on the target application. Real time HCI applications often prefer to employ simpler models, while establishing a certain amount of variability and descriptive power for specific hand postures. The features used may range from the easily detectable motion of the hand and openness or closeness of the fingers, to the angles of each finger articulation, which might be needed for an advanced sign language recognition system. Since the setup or the quality of the cameras in different smart room environments are likely to be subject to considerable change, the system is assumed to consist of the most basic cameras. Therefore, simple appearance–based methods are preferred for our application.

Appearance–based approaches are not based on prior and explicit knowledge of the hand model. Therefore, model parameters are not directly derived from the 3D spatial description of the hand. Such methods learn to relate the appearance of a given hand shape image to its actual posture.

The most common appearance–based approach is using low level features derived from the images to model hand shapes. These parameters include image moments, image eigenvectors, i.e. “eigenhands”, contours and edges, orientation histograms and Hu or Zernike moments. Currently, the training and recognition algorithms rely on Hu moments and the angle of the hand image. The Smartflow environment allows to plug in different feature extractors in a simple manner. Therefore the feature extraction module can easily be replaced with other methods for testing. Yet, the training phase should be repeated with the new feature extraction module.

Dynamic hand gestures are sequential data that involve both temporal and spatial context. Hidden Markov Models (HMMs), which can deal with the temporal invariability of signals are often employed for gesture recognition [15, 25, 36, 37]. In the recognition process, a given trajectory in the model parameter space of the gestures is tested over the set of trained HMMs in the system. The model that gives the highest likelihood is taken to be the gesture class the trajectory belongs to.

HMM is a powerful tool for sequential data modeling and recognition. Yet, it is not directly extendible to multiple synchronous or asynchronous data sequences. HMM variants such as coupled HMMs have been proposed to overcome this problem. Input–Output HMMs, which are proposed by Bengio *et al.* in [2] are neural network–HMM hybrids, which attack the shortcomings of HMMs by representing the local models, i.e. the hidden states, with possibly nonlinear and flexible complex structures such as Multi Layer Perceptrons (MLPs). IOHMMs condition the stochastic architecture of HMMs on an input sequence that has no independence assumptions. The architecture can be conditioned on any function of observable or system parameters. Therefore, the stochastic network of an IOHMM is not homogeneous. Due to their inhomogeneity, IOHMMs are expected to learn long time dependencies better than HMMs.

The gesture recognition module in BU Smart Room is designed to make use of discrete, continuous and input–output HMMs.

In the following section, the marker registration tool and the marker color signatures used in the BU Smart Room system will be explained in detail.

3.7.1. Marker Registration and Detection

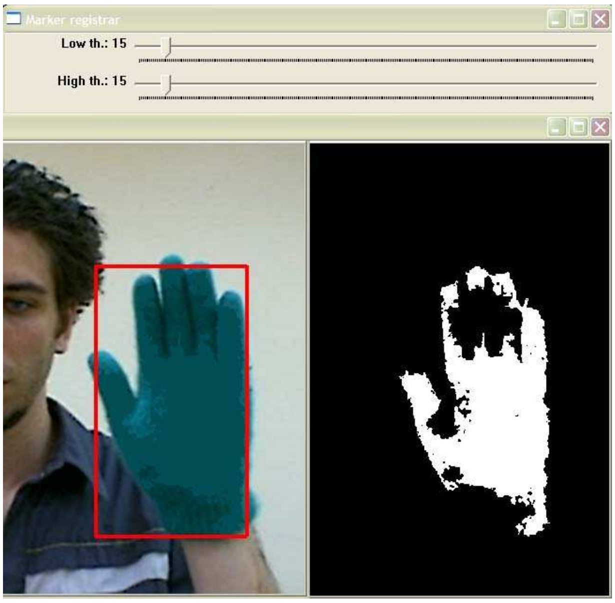
The marker registration client developed for Smartflow, namely "ipl marker registrar", aims to learn the parameters of the chosen marker. This module employs two different methodologies:

- RGB–based manual detection method
- HSV–based automatic detection method

The capabilities of the client are accessible in run–time by pressing "h", which displays a help screen. This screen shows how to switch modes, methods and settings. The manual and automatic detection modes make use of different parameters and therefore, they have different user interfaces.

In the manual registration mode, the user needs to click on the marker with the left button to run a connected components algorithm, starting with the pixel clicked. The connection criteria is euclidean distance:

- between the first pixel and the pixel being considered in the fixed range mode (activated by pressing "f"), or
- between the last pixel connected and its neighbors in the floating range mode (activated by pressing "g").



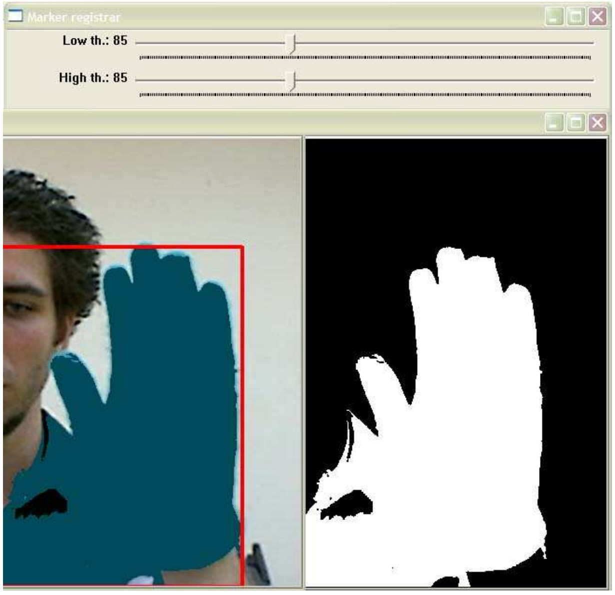
Low th.: 15

High th.: 15

Figure 11: *Effect of low settings for the fixed range mode*

Typically, fixed range mode needs much higher thresholds to be set than the floating mode. Examples of too low, too high and good settings for both of the modes are given in Figures 11, 12, 13, 14, 15 and 16. The settings in the user interface are simply the asymmetric range thresholds for values higher and lower than the original pixel. The best settings can be selected and visually confirmed by the user, which are then saved for further utilization by the hand tracking module.

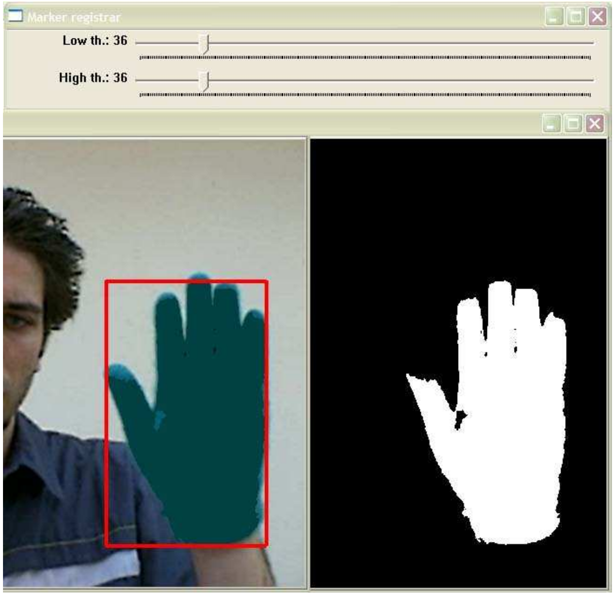
There are two problems with this approach. First of all, this mode of execution requires human intervention for marker registration, which might cause a problem if the computer is not inside the smart room. Also, RGB color space based tracking



Low th.: 85

High th.: 85

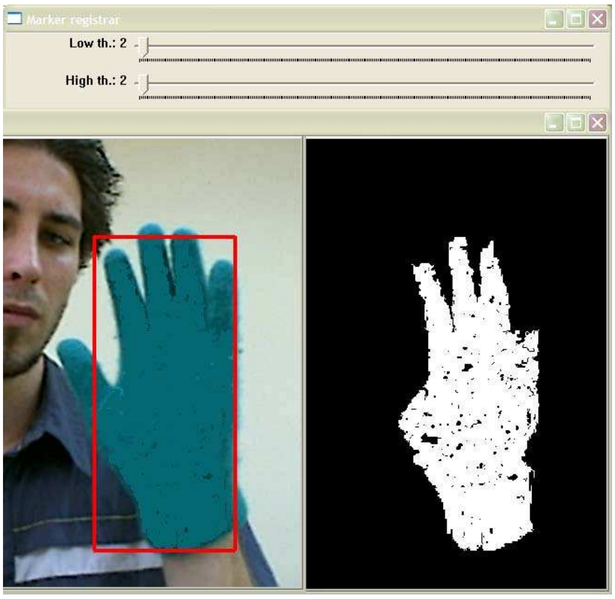
Figure 12: *Effect of high settings for the fixed range mode*



Low th.: 36

High th.: 36

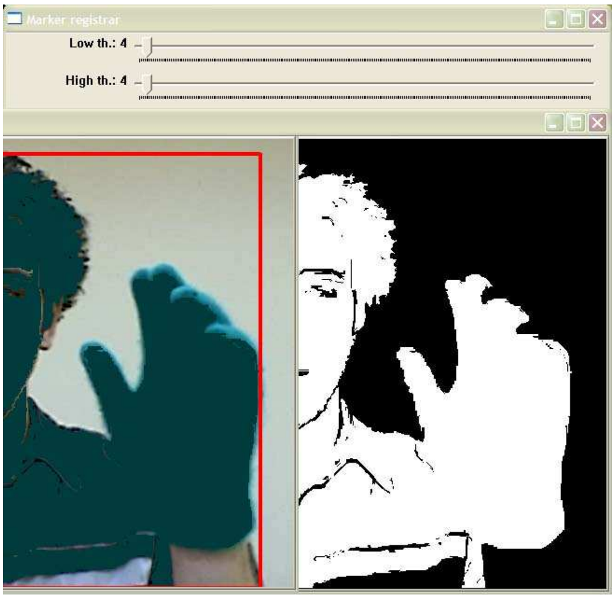
Figure 13: *Good settings for the fixed range mode*


Marker registrar

Low th.: 2

High th.: 2

Figure 14: *Effect of low settings for the floating range mode*


Marker registrar

Low th.: 4

High th.: 4

Figure 15: *Effect of high settings for the floating range mode*

is not robust to illumination changes, which would make the registration settings obsolete. Therefore, a second method is implemented, which is an automatic HSV based approach. This method can be used to detect markers without the supervision of a user, and the corresponding tracking method is more robust to illumination changes.

The main difference of the automatic method from its manual counterpart is that a supervisor does not need to click on the marker. Instead, by assuming that the marker has the fastest motion in an image sequence, this module attempts to distinguish the marker from image differences, i.e. motion. It first captures two images separated by a few frames, smoothes them with a Gaussian filter and finds their difference. The difference image is thresholded according to a user setting, and the pixels that are too dark or too bright are also eliminated using two more thresholds, also selectable via user settings. The final image is converted into HSV color space and the 2D Hue–Saturation histogram is calculated. The main consideration in this method is that the pixels corresponding to the marker fall to a single bin. In order to ensure this the number of hue and saturation bins should not be selected too high. Selecting the numbers too low on the other hand, causes the final bin to not only correspond to the marker, but possibly to some other colors, and accordingly, to other objects.

The settings in the automatic mode are as follows:

- RGB threshold - Used to determine the pixels to consider in the difference image
- Dark threshold - Leaves out the pixels that are too dark
- Bright thresold - Leaves out the pixels that are too bright
- # Hue bins - Number of hue bins in the histogram
- # Saturation bins - Number of saturation bins in the histogram

The effects of choosing wrong numbers for the RGB, dark and bright thresholds are straightforward. To clarify the effect of selection of number of bins, two cases are reproduced that correspond to very high and good selection of number of bins for hue and saturation in Figures 17 and 18.

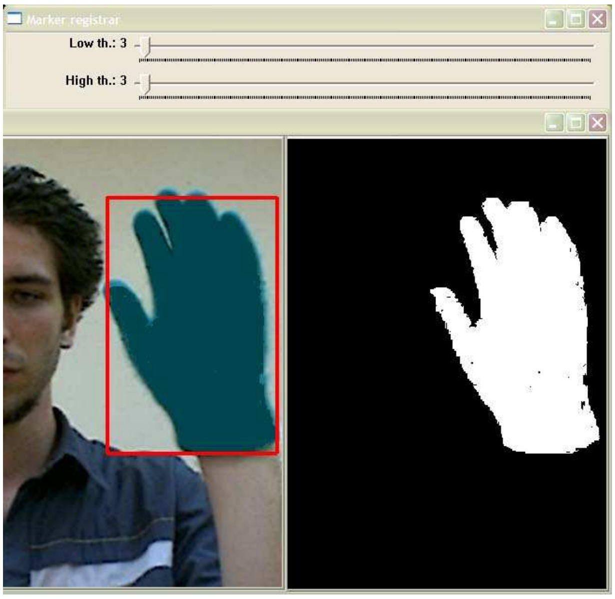
In the case of automatic registration, the marker signature is the histogram bin with the maximum number of pixels. This bin is back–projected in the tracking phase to detect the marker in the incoming frames. The tracking and feature extraction modules are explained in the following section.

3.7.2. *Hand Tracking and Gesture Recognition*

The hand detection module attempts to track the hand using the algorithm corresponding to the method used during the final registration phase. For the manual case, as explained in the previous section, the low and high thresholds and a reference color are needed to detect the hand. In order to enhance the method, double thresholding is applied. The algorithm first looks for pixels that are close enough to the refence pixel, using a threshold that is $\frac{1}{5}$ of the saved user settings. Starting from such pixels, the algorithm connects the neighboring pixels according to the criteria supplied by the registration module.

In the automatic registration case, the detection module back–projects the hue–saturation histogram calculated by the registration module and employs a region growing algorithm, where each connected pixel falls into the selected bin.

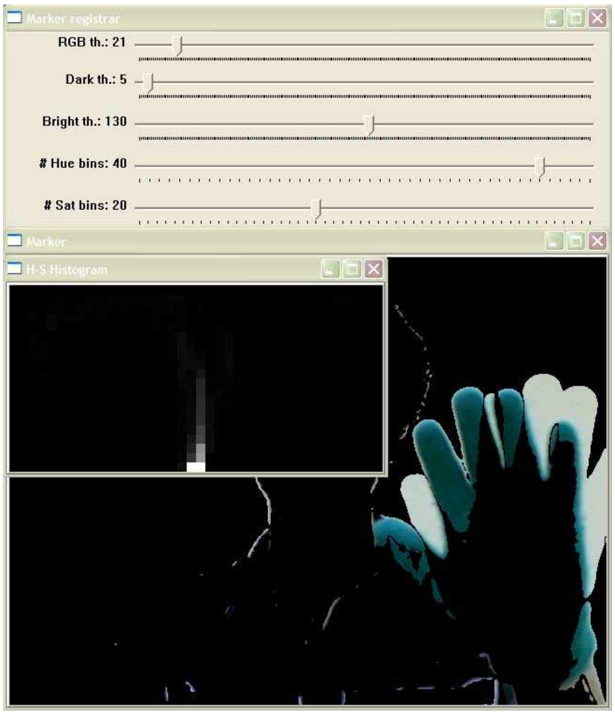
Even though the hue–based algorithm is more robust to illumination changes, both methods may generate noisy results in more challenging conditions. Therefore filtering is a necessity, especially if recognition is to be performed on the data. Due to high framerate, the marker can be assumed move linearly between the frames, and the state change can be described by a



Low th.: 3

High th.: 3

Figure 16: *Good settings for the floating range mode*



RGB th.: 21

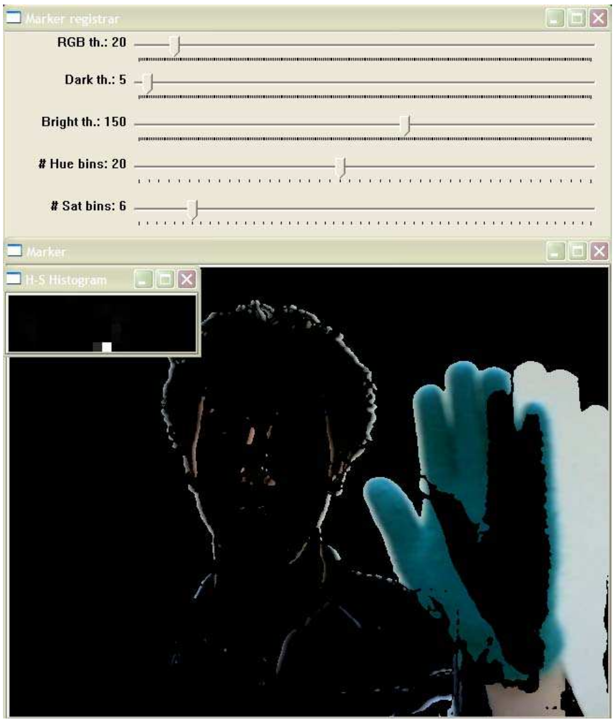
Dark th.: 5

Bright th.: 130

Hue bins: 40

Sat bins: 20

Figure 17: *Effect of too many bins. The majority of pixels fall into two bins.*



RGB th.: 20

Dark th.: 5

Bright th.: 150

Hue bins: 20

Sat bins: 6

Figure 18: *Example of a good setting for number of bins.*

linear dynamic system. Therefore, a Kalman filter is directly applicable. This filtering method is used in the gesture recognition system recognition module is shown to significantly enhance the accuracy of the results [11, 12, 13].

3.7.3. Feature Extraction and Gesture Recognition

The current design of the gesture recognition module is using Hu moments and the hand image angle as features. Shape features are only necessary when the defined gestures cannot be distinguished otherwise. If each gesture has a unique trajectory, discrete HMMs can be used for the recognition phase. The gestures chosen for the BU Smart Room require shape information to be present. Therefore, discrete HMMs cannot be used unless hand shapes are quantized, which is undesirable. Therefore, our current design produces two data streams; one corresponding to the motion vector of the hand for each frame, and one consisting of the Hu moments and the angles.

In our design, the produced data streams are to be used in a continuous HMM or IOHMM based recognition framework. Both frameworks are implemented and tested for different databases, but are not ported into Smartflow system yet. This is left as a future work. Once implemented, the module will be able to recognize the gestures in Table 1 automatically from continuous streams. The first column in Table 1 corresponds to the possible actions, and the second column corresponds to the objects that can be embedded in commands. Any combination of actions and objects is allowed, as long as actions precede the objects. Using this system, the users will be able to turn on–off or dim the lights, open–close the door gradually or entirely, command a program running on the computer or answer the phone remotely by performing hand gestures.

3.8. User Interface of the BU Smart Room System

To monitor the activity and interactions in the smart room, we implemented a graphical user interface module using OpenGL

Action	Object
Turn On	Door
Turn Off	Light
Turn Up	Phone
Turn Down	Computer

Table 1: *Defined gestures for the BU Smart Room setting.*

[22] and GLUT libraries [10], and integrated it into the SmartFlow system. For visualization purposes, we were inspired by the Marauder’s Map in the Harry Potter book series. The basic idea is that the map is covered with tiny ink dots accompanied by minuscule names, indicating every person’s location in Hogwarts, the magic academy. In a similar manner, we have created an abstract 2D map of the room and represented people inside the room with color coded feet.

After identifying a person using recognition modules, his state is set “online” in the flow providing the UI module with the necessary data. On the members list of the interface, people who are present in the room are represented with higher opacity and absent ones are represented with decreased opacity. Currently, the members list is fixed. As an improvement to the module, we want to connect it to a database and create an editable room member list. Thus, a flexible UI will be provided by preserving the object-oriented structure.

Sound based localization module feeds the UI module with coordinates relative to the microphone arrays. The data are converted to screen coordinates and an approximate location of the person inside the room is marked on the map grid. As shown in Fig. 19, the map grid represents the tiles in the smart room in a smaller scale. The map elements include the server tables, cameras and simply the entrance of the room, which constitute the obstacles of the room. As a further improvement, we want to visualize specific activities and interactions between multiple people by integrating the audio domain with visual domain. Another visualization method can be implemented on an augmented reality interface. Such an innovative interface can be implemented by annotating the people with virtual text labels containing their name on the real-time video capture [7]. Moreover, interactions can be represented by registering computer graphics objects to the video capture. Thus, the cameras in the room will provide more information than the appearance to an ordinary viewer of the smart room.

4. CONCLUSIONS AND FUTURE WORK

The aim of this project is to monitor a room for the purposes of analyzing the interactions and identities of a small set of individuals. We worked with different modalities from multiple uncalibrated sensors to observe a single environment and generate multimodal data streams. These streams are processed with the help of a generic client-server middleware called SmartFlow and signal processing modules.

Modules for visual motion detection, visual face tracking, visual face identification, visual opportunistic sensing, gesture recognition, audio-based localization, and audio-based identification were implemented and integrated under SmartFlow to work in coordination across two different platforms (Linux and Windows).

A graphical interface was implemented to visualize the functionalities which was inspired by the Marauder’s Map in the Harry Potter series. In its current state, it shows an identified person and his or her position in the room, based on audio-visual data.

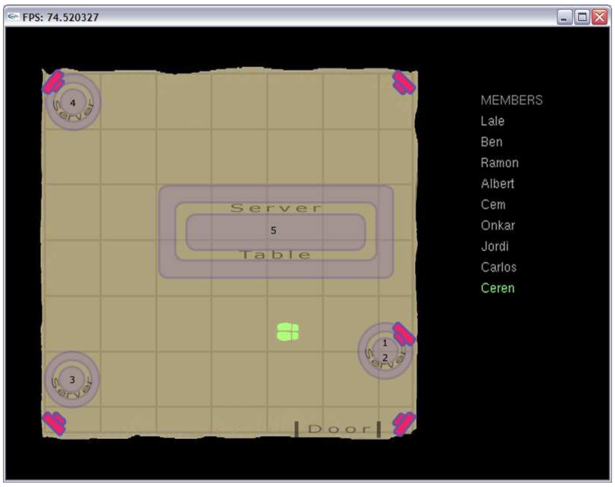


Figure 19: *First version of the GUI inspired by the Marauder’s Map.*

Although SmartFlow is a powerful tool to manage different datastreams, it has its problems, primarily due overload of hardware and limitations of network bandwidth. However in the philosophy of our project, we have attempted to solve these problems not by increasing the available computational resources, or by employing better sensors, but by intelligent allocation of the available resources to different datastreams. This was mainly achieved by allowing different framerates for video. A second objective was to tune this framerate according to various quality measures that indicate the relevance of a given datastream with respect to the identification task. This part was implemented for audio, where the quality is based on the energy of the signal.

The first stage of the project concerned setting up the architecture, collecting the datasets, implementing different modules, and experimenting with them. Since we were working with sensors of much lower cost and quality than related approaches in the literature (see for instance the results of the International Evaluation Workshop on Classification of Events, Activities, and Relationships (CLEAR 2006) [30]), we have identified two subgoals at the start of the project: opportunistic sensing and multimodal fusion. The former aims at identifying persons in the absence of robust (face) recognition. Our opportunistic sensing was implemented to build a statistical model of the pixels representing a person entering the room on the fly, for each of the cameras. Our second aim (fusion of audio-visual information) will be the starting point for future research between the participants after the project. In particular, we have identified the following future research questions:

- In our project we have managed to localize persons into the smart room using audio as well as video streams. In UPC, research has been done on event recognition using audio signals. Different modules have been written for the recognition of speech-non speech identification, hand clapping, footstep recognition, keyboard typing etc. Some of these modules were tested in the BU room. We like to work on visual recognition algorithms that can be used to make this recognition process more robust.
- Another possible research direction is implementing a biologically motivated focus of attention in order to limit the amount of (video) data processing. By allowing quality measures in the recognition process we can ignore some datastreams and/or call upon another modality or

sensor to support the recognition process. In the current version of our system, audio and video streams are used continuously to process the data. In many cases this means processing irrelevant data and overloading the system.

- For better identification and tracking results we will experiment with the fusion of the different modalities, using different methodologies (score level, decision level etc.), to improve robustness. Also, modalities can be used to support other modalities; e.g. the head pose module can improve the results for the face recognition. We will experiment with simple, limited-context discriminative patterns. A typical example is the hair colour to identify persons in the cases that the face is seen from the back, and the face recognition module can not be used.
- Finally, we want to improve our visualization modules. Determining the orientation and focus of attention is of interest in several settings (e.g. museums). If the camera projection matrices could be retrieved accurately, localization data could be represented in a 3D virtual environment in real-time. Diverse types of interaction modules can be integrated to the system. Orientation of people can be tracked by letting people wear objects with inertial orientation sensors or digital compasses, or by tracking the walk trajectory and predicting the next move.

5. ACKNOWLEDGEMENTS

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'07 web site: www.cmpe.boun.edu.tr/enterface07.

6. REFERENCES

[1] ALIZE Toolkit, Université d'Avignon: http://www.lia.univ-avignon.fr/heberges/ALIZE/ 77

[2] Bengio, Y., P. Frasconi, "Input/Output HMMs for sequence processing", *IEEE Trans. Neural Networks* vol. 7(5), pp. 1231–1249, 1996. 79

[3] Bimbot, F., J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaz, D.A. Reynolds, "A Tutorial of Text-Independent Speaker Verification," *EURASIP Journal on Applied Signal Processing*, vol.4, pp.430-451, 2004. 78

[4] Cook, D.J., S.K. Das, (eds.) *Smart Environments: Technologies, Protocols, and Applications*, John Wiley & Sons, Inc., 2005. 71

[5] Davis, S. B. and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Trans. ASSP*, No. 28, pp. 357-366, 1980. 78

[6] DiBiase, J., H. Silverman, M. Brandstein, "Robust localization in reverberant rooms," in M. Brandstein and D. Ward (eds.) *Microphone Arrays*, Springer Verlag, 2001. 78

[7] Feiner, S., MacIntyre, B., Höllerer, T., and Webster, A., "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment," in *Proc. ISWC'97*. Cambridge, MA, USA, pp.74-81, 1997. 83

[8] Fung, G., and O. Mangasarian, "Proximal Support Vector Machine Classifiers, "*Proc. KDDM*, pp. 77-86, 2001. 76

[9] Ghahramani, Z., and G.E. Hinton, "The EM algorithm for mixtures of factor analyzers," Technical Report CRG-TR-96-1, University of Toronto, (revised), 1997. 75

[10] GLUT: http://www.opengl.org/resources/libraries/glut/spec3/spec3.html 83

[11] Keskin, C., A.N. Erkan and L. Akarun, "Real time hand tracking and 3D gesture recognition for interactive interfaces using HMM", *Proc. ICANN/ICONIP*, İstanbul, Turkey, 2003. 79, 82

[12] Keskin, C., O. Aran and L. Akarun, "Real time gestural interface for generic applications", *European Signal Processing Conference, EUSIPCO Demonstration Session*, Antalya, Turkey, 2005. 79, 82

[13] Keskin, C., K. Balci, O. Aran, B. Sankur and L. Akarun, "A multimodal 3D healthcare communication system", *3DTV Conference*, Kos, Greece, 2007. 79, 82

[14] Kirby, M., L.Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. PAMI*, vol 12, no. 1, pp. 103-108, Jan. 1990. 75

[15] Lee, H. and J.H Kim, "An HMM–Based threshold model approach for gesture recognition", *IEEE Trans. PAMI*, vol. 21, no. 10, pp. 961-973, 1999. 79

[16] Luque, J., R. Morros, A. Garde, J. Anguita, M. Farrus, D. Macho, F. Marqués, C. Martínez, V. Vilaplana, J. Hernando, "Audio, Video and Multimodal Person Identification in a Smart Room," CLEAR 2006, LNCS, 2007. 74

[17] Luque, J. and J. Hernando, " Robust Speaker Identification for Meetings: UPC CLEAR-07 Meeting Room Evaluation System," to appear in *CLEAR 2007, LNCS*, 2007. 78

[18] Messer, K., J. Matas, J.V. Kittler, J. Luettin and G. Maitre, "XM2VTSDB: The extended M2VTS Database," *Proc. AVBPA*, 1999. 75

[19] Nadeu C., D. Macho, and J. Hernando, "Time and Frequency Filtering of Filter-Bank Energies for Robust Speech Recognition", *Speech Communication*, 34, pp. 93-114, 2001. 78

[20] NIST SmartFlow system: http://www.nist.gov/smartspace/nsfs.html 73

[21] Omologo, M., P. Svaizer, "Use of the crosspower-spectrum phase in acoustic event location," *IEEE Trans. on Speech and Audio Processing*, 1997. 78

[22] OpenGL: http://www.opengl.org/. 83

[23] *Proceedings of the 1998 DARPA/NIST Smart Spaces Workshop*, National Institute of Standards and Technology, pp.3-1 to 3-14, July 1998. 73

[24] Rabiner, L.R. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. IEEE*, vol.17, no.2, Feb. 1989. 78

[25] Rabiner, L.R. and B. Juang, "An introduction to hidden Markov models", *IEEE ASSP Magazine*, pp. 4–16, Jan. 1996. 79

[26] Salah, A.A., E. Alpaydın, "Incremental Mixtures of Factor Analyzers," in *Proc. ICPR*, vol.1, pp. 276-279, 2004. 75

[27] Schölkopf, B., A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002. 76

[28] Stanford, V., J. Garofolo, O. Galibert, M. Michel, and C. Laprun, "The NIST Smart Space and Meeting Room Projects: Signals, Acquisition, Annotation, and Metrics," *Proc. ICCASP*, vol.4, pp.736-739, 2003. 73

[29] Stauffer, C., and Grimson,W., "Adaptive background mixture models for real-time tracking", in *Proc. IEEE CVPR*, 1999. 73

[30] Stiefelhagen, R., and Garofolo, J., (eds.) *Multimodal Technologies for Perception of Humans*, LNCS 4122, Springer Verlag, 2007. 83

[31] Tangelder, J.W.H., Ben A.M. Schouten, Stefan Bonchev, "A Multi-Sensor Architecture for Human-Centered Smart Environments," *Proceedings CAID&CD Conference*, 2005. 71

[32] Tangelder, J.W.H., Ben A.M. Schouten, "Sparse face representations for face recognition in smart environments," *Proc. ICPR*, 2006. 75

[33] Temko, A., D. Macho, C. Nadeu, "Enhanced SVM Training for Robust Speech Activity Detection," *Proc. ICCASP*, 2007. 73, 76

[34] Vilaplana, V., Martínez, C., Cruz, J., Marques, F., "Face recognition using groups of images in smart room scenarios," *Proc. ICIP*, 2006. 74, 75

[35] Viola, P., and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proc. IEEE CVPR*, vol.1, pp.511-518, 2001. 74

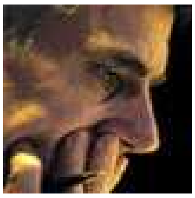
[36] Vogler, C. and D. Metaxas, "Adapting hidden Markov models for ASL recognition by using 3D computer vision methods", *In Conference on Systems, Man and Cybernetics*, pp.156–161, 1997. 79

[37] Vogler, C. and D. Metaxas, "ASL recognition based on a coupling between HMMs and 3D motion analysis", *Proc. ICCV*, 1998. 79

7. BIOGRAPHIES

Ramon Morros is an Associate Professor at the Technical University of Catalonia (UPC), Barcelona, Spain, where he teaches in the areas of digital communications, signal processing, image processing and audiovisual networks. He received a degree in Physics from the University of Barcelona (UB) in 1989 and the Ph.D. degree from the UPC in 2004. His research interests are on the fileds of image sequence segmentation and coding, sequence analysis, and visual person identification. He has participated in many public and private research projects in these areas and currently is involved in the CHIL project in the fields of video and multimodal person identification. He is author of conference and journal papers and holds one patent. He has also worked on several companies, either as a staff member or as a free-lance consultant.
Email: morros@gps.tsc.upc.edu

Albert Ali Salah worked as a research assistant in the Perceptual Intelligence Laboratory of Boğaziçi University, where he was part of the team working on machine learning, face recognition and human-computer interaction. After receiving his PhD in Computer Engineering under the supervision of Prof. Lale Akarun in 2007, he joined the Signals and Images research group at Centrum voor Wiskunde en Informatica (CWI) in Amsterdam. With his work on facial feature localization, he received the inaugural EBF European Biometrics Research Award in 2006. His current research interests include sensor networks, ambient intelligence, and spatio-temporal pattern recognition.
Email: a.a.salah@cwi.nl



Ben A.M. Schouten was born in 1953 and graduated from the Rietveld Art Academy in 1983. For many years he worked as a computer artist and his work has been exhibited in many places in the Netherlands and abroad. He received his master's degree in mathematics, specializing in chaos theory, in August 1995, founded Desk.nl in 1996, and he received his PhD in 2001, on content based image retrieval schemes and interfaces that express in an adaptive and intuitive way image similarities according to human perception. His thesis was awarded a bronze medal for Information Design at the New York Arts Festival. Currently he is a researcher at the Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. His main research interests are human computer interfacing, biometrics, smart environments, image and video understanding and data visualization. Ben Schouten teaches interface & design at the School of Arts and Technology (USAT) in the Netherlands and is the director of the European Biometrics Forum.
Email: bens@cwi.nl

Carlos Segura Perales received the BS and MS degrees in Telecommunication Engineering at the Technical University of Catalonia, and the MS degree at the Technical University of Berlin in 2003. He is currently a research fellow at the Technical University of Catalonia, working on his PhD dissertation entitled "Speaker Localization in Multimodal Smart Environments".
Email: csegura@gps.tsc.upc.edu

Jordi Luque Serrano is a PhD student at Technical University of Catalonia, Spain. He received his Engineering of Telecommunication degree from that university in 2005, and he is finishing his PhD Thesis. His research interests are related to the field of speech processing. Specifically, he has worked on the speaker identification and verification problems, diarization of meetings and automatic speech recognition.
Email: luque@gps.tsc.upc.edu

Onkar Ambekar received his BS in Instrumentation from B.A.M. University, India, in 2000, and his MS on Sensor System Technology from the University of Applied Sciences in Karlsruhe, Germany in 2004. He has worked in Germany and Austria on object detection and tracking for vehicle guidance, and holds a patent in this area. He is currently a PhD student at the CWI institute in Amsterdam.
Email: ambekar@cwi.nl

Ceren Kayalar received her BS degree in computer engineering from Dokuz Eylül University in 2004, and her MS degree from Sabancı University with a thesis entitled "Natural Interaction Framework for Pedestrian Navigation Systems on Mobile Devices". Her research interests are mobile devices and applications, virtual reality, computer graphics, augmented reality, user interfaces, context-aware computing, and human-computer interaction. Her PhD work on an outdoor augmented reality system for cultural heritage at the Sabancı University is sponsored by TUBITAK.
Email: ckayalar@su.sabanciuniv.edu

Cem Keskin was born in 1979. He received a double-major BS degree in physics and computer engineering from Boğaziçi University, in 2003. He received the MS degree in 2006, with a thesis entitled "Vision Based Real-Time Continuous 3D Hand Gesture Recognition Interface For Generic Applications Based On Input-Output Hidden Markov Models", under the supervision of Prof. Lale Akarun. Currently he pursues a PhD degree at the same institution, where he also works as a research assistant.
Email: keskinc@boun.edu.tr

Lale Akarun received the BS and MS degrees in Electrical Engineering from Boğaziçi University, İstanbul, in 1984 and 1986, respectively. She obtained her PhD from Polytechnic University, New York in 1992. Since 1993, she has been working as a faculty member at Boğaziçi University. She became a professor of Computer Engineering in 2001. Her research areas are face recognition, modeling and animation of human activity and gesture analysis. She has worked on the organization committees of IEEE NSIP99, EUSIPCO 2005, and eNTERFACE 2007. She is a senior member of the IEEE.
Email: akarun@boun.edu.tr