Chapter 6

# Audio Analysis for Multimedia Retrieval

Audio analysis and classification is widely used for video summarization, indexing and retrieval for two main reasons. A large amount of information regarding the content and events of the images and video are contained in the audio signal. Audio data can be successfully used to reduce the search space by selecting cameras according to the results of processing audio data [55]. Being one-dimensional, audio is can be processed with relatively low processing power compared to image data.

The floor sensors are unable to capture data when the people are not treading on a floor area with sensors. Furthermore, they are not activated if the pressure on the sensors is not sufficiently large: for example, when a person is sitting and leaning back with the feet resting on the floor. Audio data can be used to supplement video retrieval in such situations. Audio analysis can also be conducted independently, to retrieve multimedia related to events that are characterized by sound (e.g.: conversations). The remaining sections of this chapter describe our use of audio analysis for multimedia retrieval from ubiquitous home.

## 6.1 Audio Capture and Related Issues

Cardioid, uni-directional microphones are mounted along the corridor and at the corners of the rooms. An Omni-directional microphone is installed at the center of each room where data are captured. Figure 13a shows the positioning and orientation of the microphones. The numbering of the microphones will be used to refer to them in the

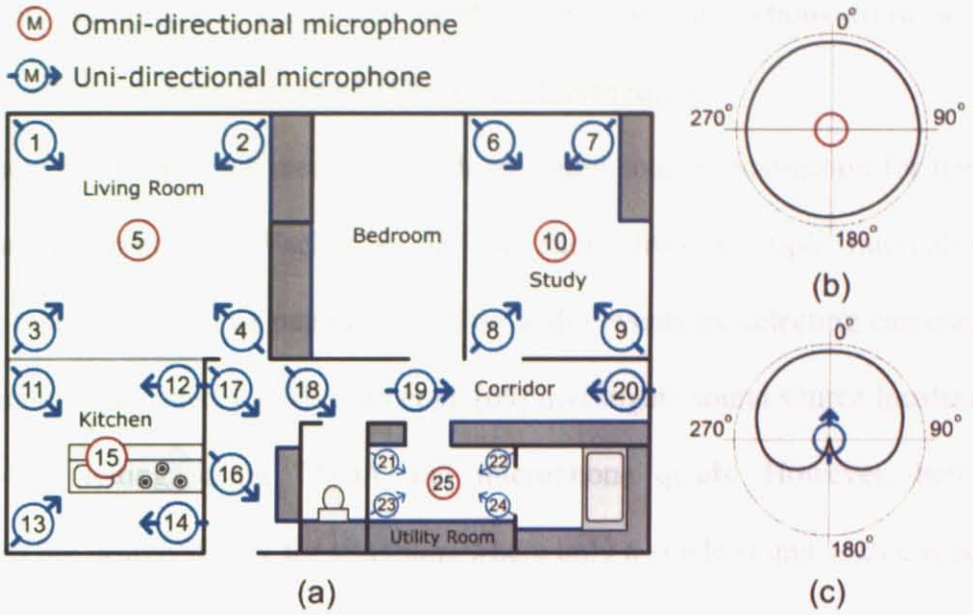



Figure 13: Microphone positioning and orientation.

coming sections of the paper. Figures 13b and 13c show the directional responses of omni directional and cardioid microphones respectively.

With 25 microphones recording continuously, the amount of audio to be processed is fairly large. The coverage of a microphone is much less restricted for a microphone than for a camera. Sounds from one source being picked up by several microphones, even those outside the room they are mounted in. Some form of sound source localization is therefore necessary to select cameras in order to show what caused a particular sound.

A brief review of the techniques used in sound source localization can be found in [56]. The most common approaches are based on time delay of arrival (TDOA), microphone arrays [57] and beam-forming techniques [58]. However, the conditions required by these approaches are not satisfied by the microphone setup in ubiquitous

home. For example, the small size of the rooms and reflections from walls can adversely affect the performance of TDOA based techniques.

There has been some recent research on sound source localization for home-like ubiquitous environments. Vacher et al. use audio from multiple microphones to facilitate tele-monitoring of patients based on audio events by selecting cameras based on maximum audio energy [59]. Bian et al. [60] investigate sound source localization in a home-like setting, using TDOA and microphone quads. However, both these techniques are designed only for situations where only a single sound source is active at a given time.

Since the microphones are located in close proximity, redundancy in captured audio data is fairly high. A trade-off has to be made between utilizing the redundancy to improve the accuracy of retrieval and minimizing processing by removing redundancy.

### 6.2 Overview of Audio Analysis

Figure 14 is an outline of the system we propose for video retrieval based on audio analysis. The audio streams are synchronized and partitioned into *segments* of one second each. Sets of audio segments that are captured using different microphones during the same 1 s interval (hereafter referred to as *segment sets*) are processed together. We start by eliminating silence from audio signals captured by individual microphones. The resulting sound segments are processed further to reduce false segments due to noise. The next step is to identify the location/s of the sound sources for each sound segment. The results are used to retrieve video directly and also classified into different classes of sound, which can again be used for video retrieval. The following sub-sections describe these steps in detail.

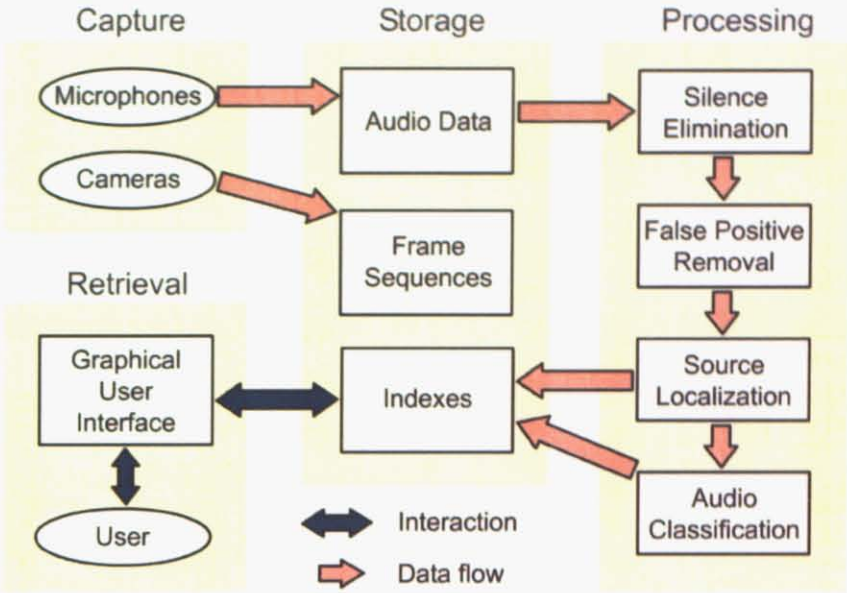



Figure 14: Overview of audio analysis.

### 6.3 Silence Elimination

A common approach for silence elimination for a single audio stream is to compare the RMS power of the audio signal against a threshold value [29]. We select this approach since it is simple, and adequate for stationary microphones in a controlled environment such as ubiquitous home. The threshold for each microphone is estimated by analyzing audio data for silence and noise for that microphone. Audio segments with a total duration of one hour were extracted from different times of day. These were partitioned into *frames* having 300 samples. Adjacent frames had a 50% overlap. The Root Mean Square (RMS) value of the samples in each frame is calculated and recorded, and the statistics obtained for each clip.

Since the probability distribution of the RMS values for different audio clips were not significantly different, the data were combined to make a single probabilistic model for silence and noise. The threshold value was selected to be at 99% level of confidence

according to this distribution. The value was selected below 100% as false negatives (sound misclassified as silence) are more costly than false positives (silence misclassified as sound). The latter can be eliminated using further analysis.

For silence elimination using the above threshold value, each audio segment is divided into overlapping frames in the same manner as previously, and the RMS value of each frame is calculated. If the calculated RMS value is larger than the threshold, the frame is considered to contain sound. Otherwise, it is considered to contain silence. Sets of contiguous frames with duration less than 0.1s are removed. Sets of contiguous frames that are less than 0.5s apart are combined together to form single segments.

The first stage of silence elimination is based on individual microphones. The audio stream is divided into overlapping frames in the same manner as previously, and the RMS value of each frame is calculated. If the calculated RMS value is larger than the threshold, the frame is considered to contain sound. Otherwise, it is considered to contain silence. The total duration of contiguous frames with RMS above the threshold are used for further processing. Sets of contiguous frames with duration less than 0.1s are removed. Sets of contiguous frames that are less than 0.5s apart are combined together to form single segments.

### 6.4 False Positive Removal

The second stage uses the data from multiple microphones in close proximity to reduce false positives resulting due to noise. For this purpose and for use in the following stages, the microphones are grouped in to regions as specified in Table 8. The bedroom has no microphones installed. However, it was identified as a separate region, for use in further analysis.

For each microphone, a binary sound segment function $B(n)$ and cumulative sound segment function $C(n)$ are defined by

$B(n) = 1$ if there is sound in the $n^{th}$ second of audio stream

$B(n) = 0$ otherwise

$C(n) = \sum B(n)$ for the set of microphones in the same room.

Noise is random and usually has a small duration. Due to its randomness, it is less likely that noise in sound segments from different microphones occur simultaneously. Due to the small duration, they can be distinguished in most situations where they do. Based on the above arguments, we use the following voting algorithm to determine the sound segment function, $S(t)$.

$S(t) = 1$ if $C(t) \otimes M(t) \geq \lceil k/2 \rceil$

$S(t) = 0$ otherwise

where

$\otimes$ denotes convolution,

Table 8: Assignment of microphones to regions.

| Region | Label | Microphones |
|---|---|---|
| Living room | LR | 1-5 |
| Study room | SR | 6-10 |
| Kitchen | KT | 11-15 |
| Entrance | EN | 16,17 |
| Corridor | CR | 18-20 |
| Utility room | UR | 21-25 |
| Bedroom | BR | - |

$M(t) = [1 \quad 1 \quad 1]$ and

$k$ = no. of microphones installed in the location

The value $S(t) = 1$ indicates that a sound was heard in the region during the $t^{th}$ second, whereas $S(t) = 0$ indicates that no sound was heard within the region during the $t^{th}$ second.

The set of sound segment functions are passed as input to the next stage, together with the audio data. Only the sound segments where $S(t) = 1$ will be processed in the following stages.

### 6.5 Sound Source Localization

After noise elimination, we have a set of sound segments for each microphone in a given region, for the situations where there was a sound *heard* in that region. We categorize the sounds contained in these segments into two types. One is *local* sounds, that is, sounds generated in the same region as the microphone belongs to. The other, *overheard* sounds, refers to the sounds that are generated in a region other than that the microphone belongs to. Each segment can contain either, or both of these types.

We intend to remove sound segments that contain only overheard sounds, from the results of noise removal. Such segments, if not removed, can mislead algorithms for video retrieval. We refer to this task as *sound source localization*, as it identifies the regions where one or more sound sources are present. However, it should be noted that we do not wish to identify at this stage whether there is a single source or multiple sources in each region. Furthermore, our source localization is restricted to region level, not to the exact location.

The regions of the house are partitioned in different ways. Some places, such as the study room, are separated from the rest by a door, and only a few sounds propagate to the other regions and rooms. However, the situation is different for some other regions such as the living room and kitchen, which are not so strongly partitioned. The situation is complicated further due to sounds from the bedroom being heard outside, while the bedroom itself does not have any microphones installed.

### *6.5.1. Localization Based on Maximum Energy*

A simple approach for sound source localization is to select the microphone that captures the sound with the highest volume and selecting the region associated with it [56]. Although this will miss out some sound sources when there are multiple sound sources emanating sound in the same 1 s interval, we investigate the performance of this approach for the purpose of comparison. Based on this approach, we design the following algorithm for *localization based on maximum energy*.

1. For the current segment set, calculate the mean square value (which is proportional to short-term energy) of the samples in each segment.
2. Calculate the average energy for each region by averaging mean square values for the segments from the microphones in that region.
3. Select the region that has the maximum value, as the region where the sound source is located.

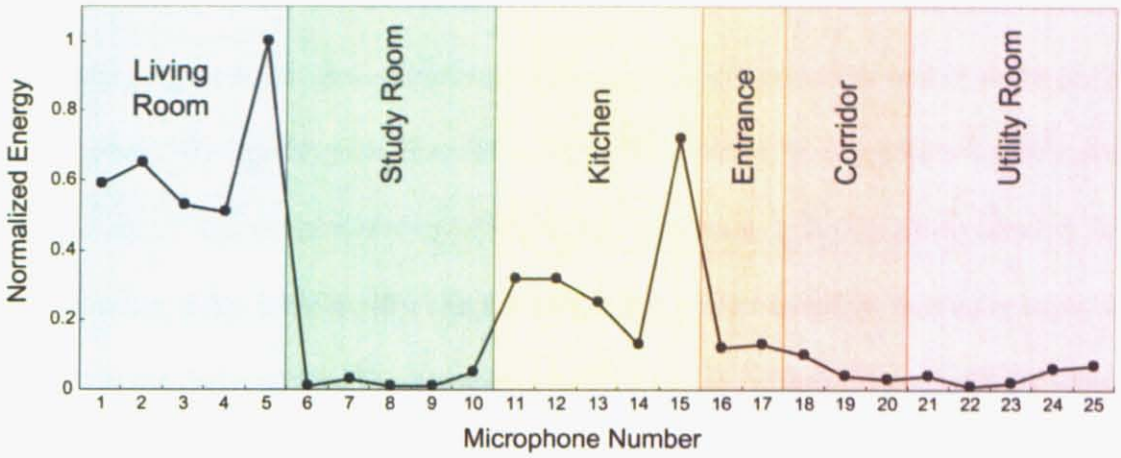



Figure 15: Energy distribution template for the living room.

### *6.5.2. Energy Distribution Templates*

A sound generated in one region of ubiquitous home can be heard in other regions, with varying levels of intensity. Based on this fact, we attempt to model local sounds in each region with the variation of energy received by each microphone. For each region $r$, a number of segment sets are selected for instances when sounds were generated only in that particular region. The energy distribution $E(n)$ for the segment set is determined by calculating the energy for each segment in the set. The *Energy Distribution Template*, $T_r$ of the region $r$ is estimated by averaging all the energy distributions. Each template is normalized to be in the range [0 1]. Figure 15 shows the template for sounds originating in the living room. Normalized energy is quite high for the microphones in the living room compared to those in other regions, with the highest energy level recorded by the omni directional microphone. However, relatively high levels of energy are registered by the microphones in the kitchen, due to the absence of a wall between the kitchen and the living room.

### *6.5.3. Scaled Template Matching*

Each audio segment set as a mixture of audio signals generated in one or more regions of the house. We hypothesize that the energy distribution of a segment set is a linear combination of one or more energy distribution templates, and attempt to identify them (see *Appendix B* for details). We use the following *scaled template matching algorithm* for finding these templates. The main idea behind the algorithm is to repeatedly identify the loudest sound source available, and removing its contribution. This process repeats until it is evident that there is no significant sound energy left to assume the presence of a sound source. The procedure for scaled template matching is described below:

1. Calculate the energy distribution, $E(n)$ for the current segment set
2. For each region $r$, determine average energy $E_r$ by averaging energy values of the microphones in that region
3. Find the region $r$ in the distribution with the maximum value of $E_r$ for the current energy distribution $E(n)$. Identify this region as containing a local sound segment.
4. Scale $E(n)$ by dividing by the max value in that region, $A_m$.
5. Subtract the template $T_r$ corresponding to this region, $r$, and obtain the residual $R(n)$
6. If the average value of $R(n) \leq 0.2$ then stop. Otherwise, multiply $R(n)$ by $A_m$ again
7. Repeat steps 2-5 on $A_m R(n)$, for $k$ times where $k$ is the number of regions where sound segments are detected.

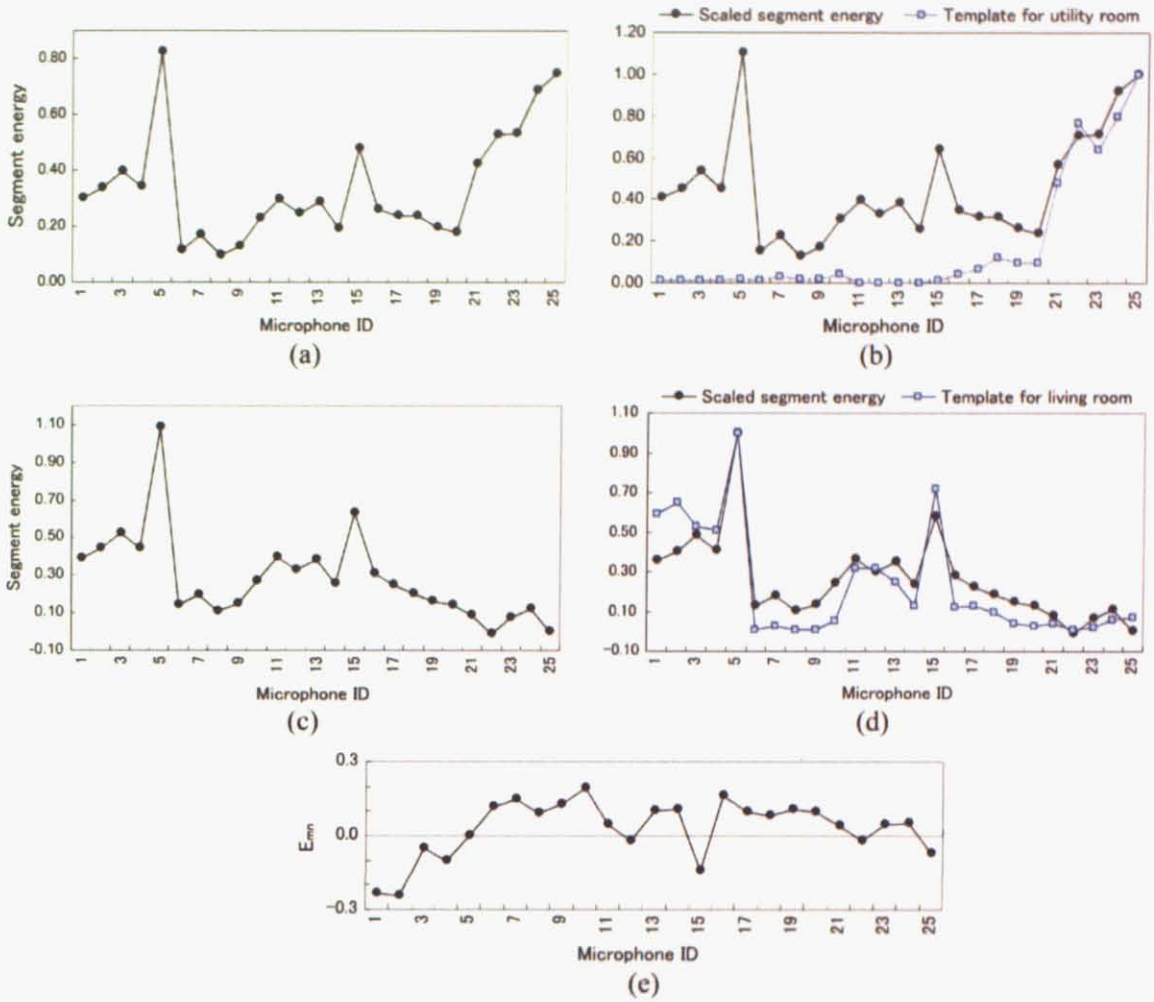



Figure 16: Scaled template matching for source localization. (a) original segment set (b) matching with template for the utility room (c) energy distribution after template subtraction (d) matching with template for the living room (e) residual segments

Figure 16 visualizes the application of this algorithm to a set of audio segments. In this case, there have been local sound segments in the utility room and the living room. Figure 16a shows the energy distribution for the current segment set. Since the highest average energy is present in the *utility room* region (microphones 21-25), the utility room is identified as having local sounds. The energy distribution is scaled by diving by the energy for microphone 25 (the maximum value for this region), and the template for utility room is subtracted from the result (Figure 16b). The result after subtraction, $R(n)$, is shown in Figure 16c. The process is repeated for the template for

the living room (Figure 16d), which records the highest average in $R(n)$. The algorithm stops at this point as the average value becomes lower than the set threshold (Figure 16e).

### 6.6 Audio Classification

The results of source localization can be used to retrieve video for basic *audio events*, that is, instances where something generated a sound in a given region. However, this retrieval is at a very low semantic level and will results in a large amount of video. Retrieval by different classes of audio, such as voices and music, will greatly enhance the precision of retrieval. We conducted a pilot study on audio classification. An audio database was constructed by studying the sound segments extracted from experiments in ubiquitous home. These were classified into the categories shown in Table 9. The duration of each audio clip in the database was between 1 and 15 seconds.

The classes were selected by observing data from the real-life experiment, and

Table 9: Description of audio database.

| Label | Class | No. of audio clips |
|---|---|---|
| 1 | Footsteps | 40 |
| 2 | Noise | 40 |
| 3 | Voices of people inside the house | 112 |
| 4 | Voice of a household robot | 32 |
| 5 | Voices from television | 50 |
| 6 | Other sounds from television | 60 |
| 7 | Vacuum cleaner | 60 |
| 8 | environmental sounds | 86 |
| | Total | 480 |

aiming at detecting higher level events such as conversations, and watching TV. We attempted audio classification based on frame-based and clip-based time domain features. These features are relatively easy to calculate given the large amount of data, and facilitate reasonably accurate classification according to results reported in similar work [61]. The frame size for calculation of features was the same as for silence elimination. The selected features were:

- Mean of RMS values of the set of frames
- Standard deviation of RMS values of the set of frames
- Mean of Zero crossing ratios of the set of frames
- Standard deviation of Zero crossing ratios of the set of frames
- Silence ratio of the audio clip

All of the features were calculated according to the definitions in [61]. Each feature was normalized by subtracting the mean and dividing by the standard deviation for the feature in the entire database.

A number of classifiers, including *Multi-layer Perceptron (MLP)*, *k-Nearest Neighbor* and *Random Forest* were trained and tested on the database. The results were evaluated using 10-fold cross validation. *AdaBoost with MLP* classifier yielded the highest overall accuracy, and therefore was selected as the classifier to be implemented in the proposed system.

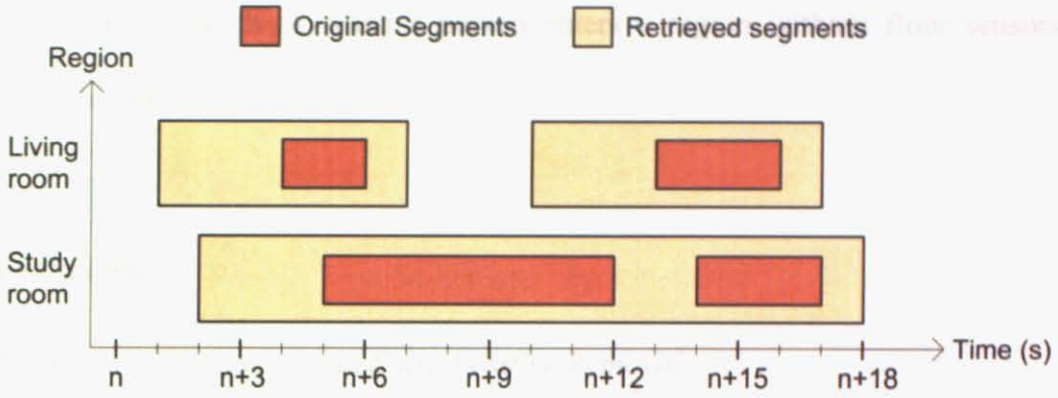



Figure 17. Microphone positioning and orientation.

### 6.7 Video Retrieval

The method of audio-based retrieval is fairly straightforward once audio segments are localized. First, consecutive sound segments that are less than 4 seconds apart are joined together, to prevent retrieving a large number of fragmented videos for the same event. After joining, the start time for retrieving video is set to be 3 seconds earlier than that of the sound segment, to allow the user to prepare to receive the audio event. The end of the video clip is set to be 1 second later than that of the sound segment. Figure 17 is a visual representation of this process. Video clips are retrieved from all cameras in the region for the time interval determined as above. The same approach is applicable for retrieving video segments for different classes of audio.

The video clips retrieved for footstep sequences are extended using sound segments as follows. If there is only one footstep sequence overlapping partially with a sound segment, it is combined with the footstep sequence. The video created by video handover is extended to include the time during which sounds were present before the start of the footstep sequence, or after the sequence ended. This improves the video in

certain situations, such as when a person enters a region without floor sensors but continues to talk.

## 6.8 Evaluation

### *6.8.1 Silence Elimination and False Positive Removal*

The performance of silence elimination and false positive removal was evaluated using 90 hours of audio data, extracted from the data captured during a single day in ubiquitous home. Silence elimination resulted in 0% false negatives and 2.2% false positives on this data set. The algorithm for false positive removal was able to remove 83% of the false positives that remained after silence elimination in individual audio streams.

### *6.8.2 Sound Source Localization*

We used 200 minutes of audio data captured during the real-life experiment, for evaluation of sound source localization. The data were captured from 7:45 a.m. to 11:05 a.m. on the 12$^{th}$ April 2005, from all microphones. This time interval was selected to ensure that all regions of the house were used for a considerable duration for ordinary household activities. The data were transcribed manually to find out the ground truth with regard to the sounds generated in each region (Table 10).

We studied the degree of overhearing between different regions of ubiquitous

Table 10: Ground truth for audio data.

| Region | LR | SR | KT | EN | CR | UR | BR |
|---|---|---|---|---|---|---|---|
| No. of segments | 5238 | 639 | 1667 | 134 | 192 | 814 | > 105 |

home, after applying silence elimination and noise removal on these audio data. To present the situation with overheard sounds in a meaningful way, we calculate the ratio of overhearing, $H_{ij}$ as

$$H_{ij} = N_{ij}/T_j$$

Where

$N_{ij}$ =Sound coming from room $j$ and heard in room $i$

$T_j$ = Total number of sound segments generated in room $j$

and $i, j \in \{LR, SR, KT, EN, CR, UR, BR\}$

Table 11 shows the $7 \times 7$ matrix $H = [H_{ij}]$. To take an example on how to interpret the values, 73% (0.73) of the sounds coming from the living room can be heard in the kitchen. It is evident that the results are not exactly symmetric. For example, only 28% of the sounds coming from the kitchen can be heard in the living room, compared to the 73%for the other way. The reason is that most sounds generated in the living room are loud sounds, for example group conversations and sounds from the TV. The sounds generated in the kitchen, on the other hand, are softer (e.g. preparing food).

For maximum energy-based localization, it is not logical to calculate matrix $H$ as the algorithm can detect only one region with local sounds for a given segment set. Table 12 shows the situation with overheard sounds after source localization using

Table 11: Overheard sounds before source localization.

| Region | LR | SR | KT | EN | CR | UR | BR |
|---|---|---|---|---|---|---|---|
| LR | 1.00 | 0.00 | 0.28 | 0.03 | 0.13 | 0.00 | 0.00 |
| SR | 0.00 | 1.00 | 0.00 | 0.01 | 0.39 | 0.07 | > 0 |
| KT | 0.73 | 0.00 | 1.00 | 0.12 | 0.01 | 0.00 | 0.00 |
| EN | 0.06 | 0.00 | 0.10 | 1.00 | 0.36 | 0.16 | > 0 |
| CR | 0.00 | 0.02 | 0.02 | 0.27 | 1.00 | 0.18 | > 0 |
| UR | 0.00 | 0.00 | 0.00 | 0.02 | 0.27 | 1.00 | 0.00 |
| BR | ? | ? | ? | ? | ? | ? | ? |

Table 12: Overheard sounds after source localization.

| Region | LR | SR | KT | EN | CR | UR | BR |
|---|---|---|---|---|---|---|---|
| LR | 1.00 | 0.00 | 0.03 | 0.02 | 0.03 | 0.00 | 0.00 |
| SR | 0.00 | 1.00 | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 |
| KT | 0.05 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| EN | 0.01 | 0.00 | 0.00 | 1.00 | 0.06 | 0.05 | 0.00 |
| CR | 0.00 | 0.04 | 0.00 | 0.13 | 1.00 | 0.05 | > 0 |
| UR | 0.00 | 0.00 | 0.00 | 0.02 | 0.06 | 1.00 | 0.00 |
| BR | ? | ? | ? | ? | ? | ? | ? |

scaled template matching.

We define the Precision $P$, Recall $R$, and Balanced F-measure $F$ for audio segmentation for each region of the house, as:

$$P = N_c / N_t$$

$$R = N_c / N_a$$

$$F = 2PR/(P+R)$$

where

$N_c$ = no. of local audio clips retrieved correctly

$N_t$ = total no. of clips retrieved

$N_a$ = actual no. of local audio clips

Table 13 summarizes the results of the evaluation. The precision, recall and balanced F-measure are shown for original data after false positive removal, for the results using maximum energy based selection and for the results obtained by scaled template matching. The values cannot be determined for the bedroom since ground truth is not known due to the absence of microphones.

It is evident that the results for rooms other than the kitchen have improved to near 100% with the scaled template matching algorithm. For the kitchen, there has been a significant improvement even though the results are not as good. The high recall rates demonstrate the ability of the algorithm to localize multiple sound sources with a high accuracy.

The high accuracy recorded with the proposed scaled template matching algorithm is mainly due to the fact that it utilizes the high degree of partitioning present

Table 13: Accuracy of sound source localization.

| | Before localization | | | Max. energy | | | Proposed method | | |
|---|---|---|---|---|---|---|---|---|---|
| Region | P | R | F | P | R | F | P | R | F |
| LR | 0.56 | 1.00 | 0.72 | 0.97 | 0.93 | 0.95 | 0.95 | 0.99 | 0.97 |
| SR | 0.78 | 1.00 | 0.88 | 1.00 | 0.45 | 0.62 | 0.96 | 0.80 | 0.88 |
| KT | 0.72 | 1.00 | 0.84 | 0.96 | 0.78 | 0.86 | 0.97 | 0.79 | 0.87 |
| EN | 0.69 | 1.00 | 0.82 | 1.00 | 0.60 | 0.75 | 0.86 | 0.89 | 0.88 |
| CR | 0.46 | 1.00 | 0.63 | 1.00 | 0.80 | 0.89 | 0.84 | 0.97 | 0.90 |
| UR | 0.71 | 1.00 | 0.83 | 1.00 | 0.82 | 0.90 | 0.91 | 0.89 | 0.90 |

Table 14: Results of audio classification.

| Class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Precision | 0.68 | 0.91 | 0.79 | 0.91 | 0.73 | 0.82 | 1.00 | 0.78 |
| Recall | 0.80 | 1.00 | 0.82 | 0.97 | 0.76 | 0.88 | 1.00 | 0.67 |
| F-measure | 0.74 | 0.95 | 0.81 | 0.94 | 0.74 | 0.85 | 1.00 | 0.72 |

in a home-like environment. Different results can be expected in environments that are larger and have less partitioning. The evaluation of 200 minutes of audio from each of the 25 microphones was quite tedious. While this was done with care to ensure as less error as possible, there may still be some human error (of about 1-2%) included in the results.

### *6.8.3 Audio Classification*

The results of audio classification are presented in Table 14. The accuracy of classification using only time domain features suggest that more accuracy can be obtained by adding frequency domain or MFCC domain features. The classes in the sound database have to be refined further, after a study of requirements for retrieval based on audio classification.

Chapter 7

# Event and Action Detection Using Multiple Modalities

The algorithms described in the previous chapters facilitate efficient multimedia retrieval by automated selection of sources, video summarization and audio classification. These results can be interpreted as corresponding to basic events. The video clips and key frames retrieved using footstep sequences correspond to "human presences". However, this event is quite coarse in terms of granularity, as the person can be walking, standing, or performing several other tasks during his/her presence. The video retrieved for different audio classes correspond to "audio events" of the same class, in the corresponding location; for example, "sounds from the television in the living room", "vacuum cleaning the study room". As for the second example, it can also be interpreted as an action.

The rest of this chapter describes the techniques we use to detect additional events using image data (which were hitherto unprocessed) and to segment the video clips retrieved for footstep sequences based on the activities, or *actions*, performed by the person.

## 7.1 Issues

Both action recognition and event recognition are hard, owing to two main problems. The first problem is related to definition of an action or an event. There has been an ongoing discussion on the question "what are actions and events in the context of multimedia" over the last decade [62][63]. Some actions can be interpreted as events,

and vice versa, further complicating the issue. Since it is hard to define what events and actions are, identifying a complete set of events or activities is a difficult task. The usual approach to solve this problem is to recognize a desired set of actions or events against "others" [64]. An alternative, which is particularly useful in surveillance, is to detect "unusual actions and events" [65].

The second problem is the recognition of desired actions and events with the available sensor data. The common approach used to solve this problem is to build a classifier based on supervised learning using a set of training data. However, most of the time, the sensor data are not sufficient to represent the desired action or event, unless certain preconditions are satisfied. For example, most of the researches on sports video retrieval rely on the common camera positioning for video capture, and context data representing the play field [66]. Furthermore, the features extracted from the sensor data might not contain information that facilitates easy and accurate classification, unless the features are selected carefully. Therefore, accurate recognition of actions and events with high semantic levels is a difficult task.

We use a *data-driven* approach in action and event detection for the ubiquitous home. The actions and events are selected by observing the size and dimensionality of data, and also the content. We choose simple actions and events that are easy to detect, yet helpful to the user to interpret the results. While the algorithms described in the previous sections were based on analyzing a single sensor modality, we combine data from multiple sensory modalities where they compliment each other, for more accurate action recognition. The following sections describe the algorithms used for action and event detection in ubiquitous home.

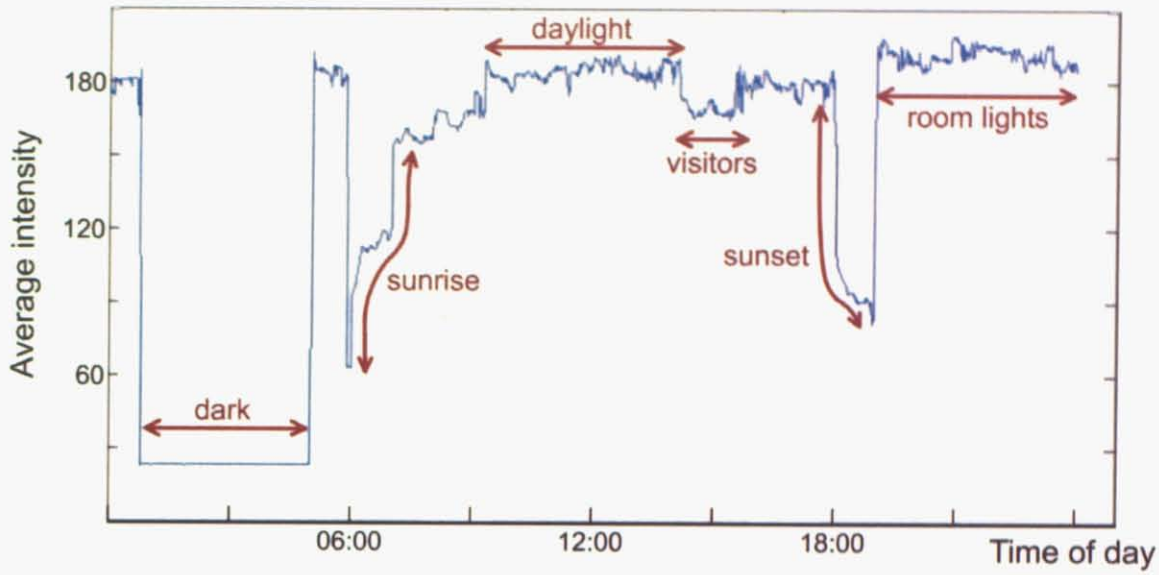



Figure 18: Lighting changes in the living room and the corresponding events.

### 7.2 Event detection based on lighting changes

We intend to use change of light level as a cue for video retrieval from all the locations in the house. Lighting changes in a room can take place due to very simple events, such as turning a light on/off or opening a window curtain. However, if combined with the scene context, they can be used to identify significant events that take place in a house. For instance, lighting changes in the rooms at night will provide a rough idea as to when the residents went to sleep, and whether some of them woke up at night etc. Being very quick events, they can be used to create very short summaries of a day's events. Figure 18 shows the variation of light level inside the living room of ubiquitous home, during a full day. The text labels indicate the actual events that caused the corresponding light levels and changes.

We presume that the lighting level in a selected region of ubiquitous home at a given time can be represented by the average intensity of all pixels in all the images captured in that region at that time. Lighting changes are relatively easy to detect, as

they are represented by sharp changes in average intensity calculated as above. However, the problem is to find a threshold level for this change that is suitable for separating significant events (such as entering a room in the morning when it is partially lit from outside) from insignificant events such as a curtain being blown away by the wind. For rooms or regions with windows, the amount of external light changes with the seasons, weather, curtains, and time of day. For ubiquitous home, this tasks is made further complicated by automatic gain control in the cameras. Setting a single threshold level to match all these conditions is impossible.

Our approach to solve this problem is to assign a rank of significance to each lighting change, based on the sharpness of change. The user selects the rank and browses the events through an interactive interface, thereby reducing the search space intuitively.

We consider a day as the unit of video to be processed simultaneously, as it includes a full cycle of lighting variations both inside and outside a house. For each camera, the frame intensity function $I(t)$, where $I(t)$ = the average intensity of the frame $t$, is calculated. This is low pass filtered by averaging over a window of 5 frames (corresponds to a 1 s interval).

The intensity gradient function, $g(t)$ is calculated as

$$g(t) = |I(t+1) - I(t)|$$

The small intensity variations, which are mainly due to moving objects and persons, are removed by thresholding $g(t)$. For this, the threshold for each camera was

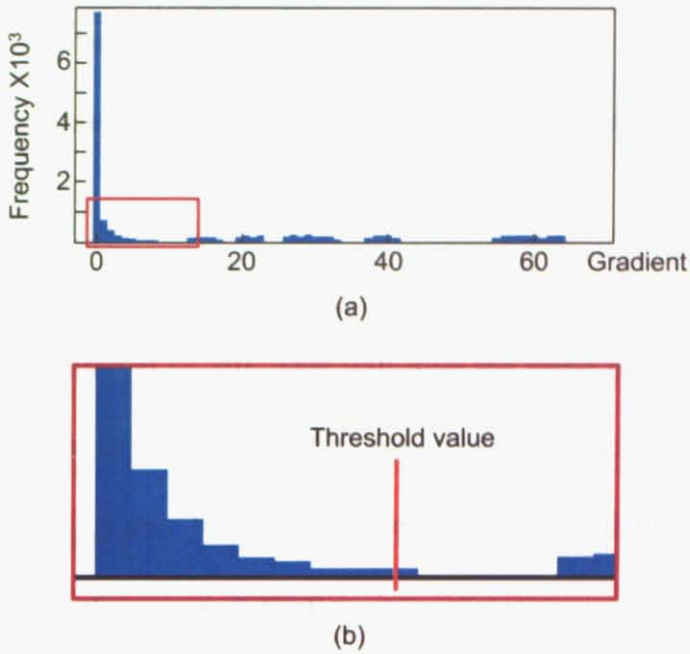



Figure 19: Threshold estimation using gradient histograms.

determined using 24 hours of training data. The intensity gradient function is calculated for the training data set and its histogram with a bin size of 1.0 is constructed (Figure 19a). The half-Gaussian shaped cluster of bins at the beginning of the histogram corresponds to insignificant lighting changes. The threshold value $H$ is selected at the last bin value in this cluster. Figure 19b shows an enlarged section of the histogram in Figure 19a with the threshold value detected using the above method.

The thresholded gradient function, $G(t)$ is defined as

$$G(t) = 0 \text{ if } g(t) < H$$

$$G(t) = g(t) \text{ otherwise}$$

The scaled gradient function, $N(t)$ is determined by

$$N(t) = 0 \text{ if } G(t) = 0$$

$$N(t) = 1 + \frac{G(t) - G_{min}}{G_{max} - G_{min}} \times 9 \quad \text{if } G(t) > 0$$

Where $G_{min}$ and $G_{max}$ are the minimum and maximum gradients recorded during the selected date.

The objective of scaling is to set the range of $N(t)$ to [0 10], with all lighting changes scaled to the interval [1 10]. This will give the same priority to events taking place in different regions in different lighting conditions. In case real-time calculation is necessary, $G_{min}$ and $G_{max}$ can be estimated using previous data, although this will cause a minor change in the range of $N(t)$.

The rank function $R(t)$ for each region is determined by taking the summation of $N(t)$ for all the cameras in that region. The partitioning of regions is the same as shown in Section 4.4. It is evident that $R(t)$ is higher for sharper lighting changes, and for lighting changes seen by all cameras in the region. The user, at the time of retrieval, can interactively specify $R(t)$ and retrieve events. This is described in detail in Chapter 8.

## 7.3 Action Classification for Retrieval

Step segmentation and video handover results in video and key frame sequences. However, these can be lengthy if the persons tracked stayed a long time in the house. Furthermore, it is desirable to partition these results further according to the actions they performed.

### *7.3.1. Clustering of Footstep Sequences*

We observed the results of clustering different combinations of variables in sensor

activation data, using Kohonen Self Organizing Maps (SOM). The activation durations showed a grouping that is independent from other variables. Durations between 0.10 and 0.96 seconds formed a distinct cluster consisting of 90% of the data. To examine if this grouping leads to any meaningful summarization, the video data was retrieved using the following approach. Sensor activation data was segmented using [0.10, 0.96] seconds as the threshold interval. The activations that occur with less than 1 second time gap in between were clustered to obtain activation sequences, corresponding to time intervals. Video clips for these time intervals were retrieved from the relevant cameras and examined.

It was evident that video clips corresponding to the segment with durations > 0.96 s corresponded to video containing activities with irregular or infrequent foot movement, such as sitting, waiting, and preparing food. The rest corresponded to walking and vacuum cleaning. Therefore, clustering using this approach enables retrieval of short video clips pertaining to two basic categories of actions.

### *7.3.2. Detailed Action Classification*

Clustering of sensor activations, as described in the previous section, results in only a basic classification of activities. It is necessary to have more specific activity classification, to be able to retrieve video for queries related to daily life. An action database (Table 15) was constructed by extracting portions of footstep sequences created in footstep segmentation of the data from the real-life experiment. The selected actions seem somewhat unbalanced; for instance, walking and standing are basic body gestures and cooking is an activity with higher detail. However, the selected actions are those appearing most frequently in footstep sequences retrieved from ubiquitous home.

Therefore, it is more practical to train a classifier for these actions. The duration of the sequences was between 30 seconds to 5 minutes. Each sequence contained a minimum of 20 sensor activations.

Sensor activations taken individually do not reflect the dependence of the footsteps on previous footsteps and the relationship within a group of footsteps. For example, a standing person will keep the feet at nearly the same place, with occasional changing of weight on one foot to the other whereas a walking person has only one foot on the floor most of the time and keeps a somewhat regular distance between consecutive footsteps. We define and calculate the following features for activity classification for an activation sequence $S = \{A_1, A_2, ..A_n\}$ where $A_i$ is the $i^{th}$ sensor activation of the sequence.

- Mean and standard deviation of sensor activation durations
- Standard deviation of X coordinates of the sequence
- Standard deviation of Y coordinates of the sequence
- Overlap of footsteps $O_s$, defined as

Table 15: Composition of the activity database

| Action | No. of sequences |
|---|---|
| Walking | 40 |
| Standing | 56 |
| Sitting on a chair | 30 |
| Sitting on the floor | 10 |
| Cooking or washing dishes | 22 |
| Vacuum cleaning | 10 |

$$O_s = D / T$$

Where

$D$ = sum of durations in all activations $\{A_1, A_2, ..A_n\}$

and

$T$ = Duration of $A_n$ + start time of $A_n$ - start time of $A_1$

- Activation rate $R$, defined as

$$R = n / T$$

The sequences in the activity database were classified and tested using WEKA Machine learning tools [68]. Multi-layer Perceptron (MLP), *k*-Nearest Neighbor and Random Forest classifiers with different parameters were trained and tested on the database. The results were evaluated using 10-fold cross validation. An MLP classifier with 3 layers yielded the highest overall accuracy, and therefore was selected to be implemented in the proposed system.

### *7.3.3. Combining other modalities to improve accuracy*

The data from other sensors were used to improve the accuracy of activity classification, based on the following heuristic rules:

- For vacuum cleaning, audio classification for the same region should detect vacuum cleaning sounds for the corresponding time interval. Otherwise, the action is re-classified as walking.

- A cooking event is rejected if at least 60% the X and Y coordinates of the sequence of footsteps are outside the area near the kitchen pantry. In this case, the action is re-classified as standing
- For sequences classified as "sitting on chair", the X and Y coordinates should lie within predefined regions near the chairs. This has a problem in the long term as people tend to rearrange furniture at times, but is logical for heavy chairs like sofas for a small house and duration in the order of months.

## 7.4 Evaluation

### *7.4.1 Event detection based on lighting changes*

The number of lighting change events detected per hour was in the range of 0 to 8. It was possible to detect fine changes such as opening a door and entering a room that is already lit and sharp events such as turning lights of a room on/off at night. Due to the positioning of cameras and automatic gain control characteristics, very few false positives caused by moving people close to the camera were detected. Figure 20 shows the lighting changes that took place in the living room between 5:00 a.m. and 6:00 a.m. on the 12$^{th}$ of April 2005. For each event, images captured before and after the lighting change from one camera are displayed. By looking at only these two events, it is possible to understand that the family members in the photo entered the living room at 05:01a.m., and one of them left the room at 05:54 a.m.

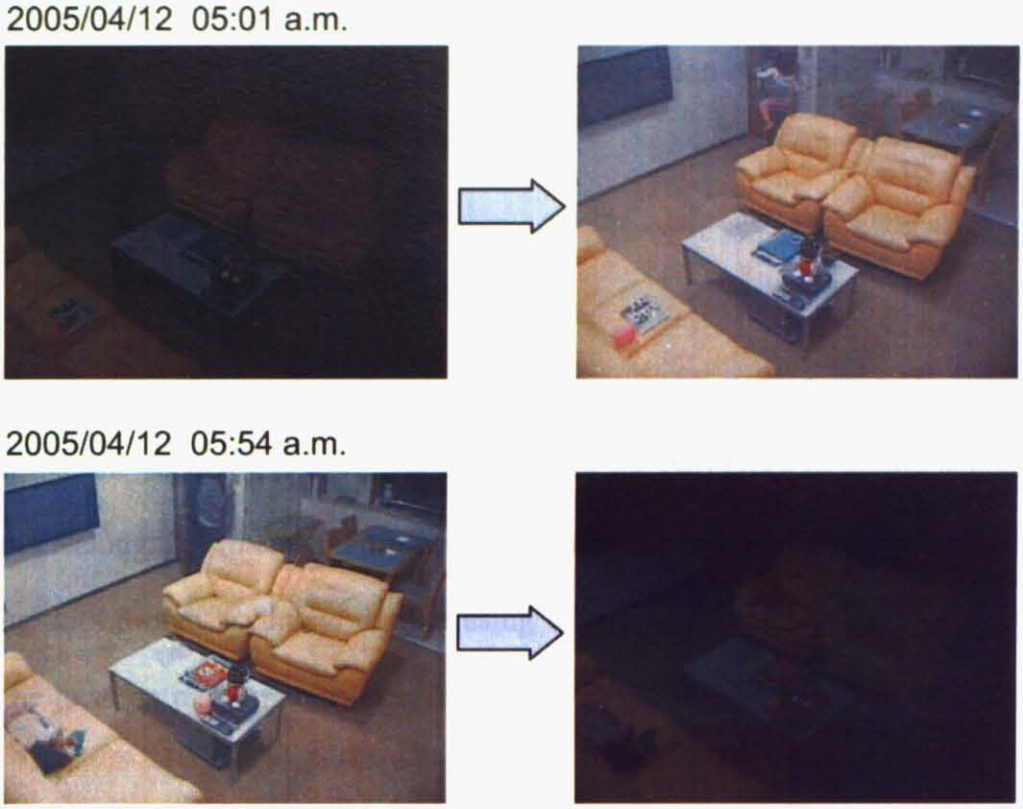



Figure 20: Retrieved events for lighting changes.

### *7.4.2 Basic Activity Classification*

We calculate precision P, recall R, and balanced F-measure F for evaluation of retrieval of video with regular foot movement.

$$P = N_c / (N_c + N_m)$$

$$R = N_c / (N_c + N_o)$$

$$F = 2PR / (P + R)$$

Here $N_c$ is the number of correctly retrieved video clips, $N_m$ is the number of clips that were not retrieved, and $N_o$ is the number of mistakenly retrieved clips. Step

sequences with accurate step segmentation were clustered to retrieve video clips and the clips observed to evaluate the performance. The precision of retrieval was 93.7% and the recall 96.7 %. The F-measure was 95.2%.

### *7.4.3 Detailed Action Classification*

Table 16 presents the results of action classification for the selected classifier. It is evident that the accuracy is lower for recognizing the two types of sitting actions, and vacuum cleaning. An interesting observation is that cooking can be distinguished from standing with a high accuracy, despite using only floor sensor data and even without using the location of the person as a feature. Further examination of data revealed that

Table 16: Accuracy of action recognition before using multiple modalities.

| Label | Action | Precision | Recall | F-measure |
|---|---|---|---|---|
| 1 | Walking | 0.848 | 0.780 | 0.813 |
| 2 | Standing | 0.811 | 0.750 | 0.779 |
| 3 | Sitting on a chair | 0.455 | 0.625 | 0.526 |
| 4 | Sitting on the floor | 0.571 | 0.500 | 0.533 |
| 5 | Cooking or washing dishes | 0.902 | 0.920 | 0.911 |
| 6 | Vacuum cleaning | 0.600 | 0.714 | 0.652 |

Table 17: Confusion matrix before using multiple modalities.

| Correct action | Classified as | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 39 | 1 | 1 | 0 | 0 | 9 |
| 2 | 1 | 30 | 2 | 1 | 5 | 1 |
| 3 | 0 | 1 | 5 | 2 | 0 | 0 |
| 4 | 0 | 1 | 3 | 4 | 0 | 0 |
| 5 | 0 | 4 | 0 | 0 | 46 | 0 |
| 6 | 6 | 0 | 0 | 0 | 0 | 15 |

there is higher overlap of footsteps and activation ratio for cooking compared to standing. While cooking, a person tends to make a number of small foot movements and puts weight on both feet due to the need to balance the body while handling the kitchen appliances. A person who is just standing, on the other hand, usually rests his/her weight on one foot and switches it regularly, rather than moving feet. These differences were represented well in the selected features, allowing classification with high accuracy.

The confusion matrix (Table 17) shows that most of the confusions occur between the two types of sitting actions. Walking and vacuum cleaning is another pair with a large amount of confusion. The heuristic rules involving other modalities were selected considering the patterns of confusion. Tables 18 and 19 show the accuracy and the confusion matrix after applying these rules. It is evident that there is a 1% to 18% improvement in the overall accuracy represented by F-measure.

### 7.5 Discussion

Given the large amount of information contained in image data, it should be possible to detect more actions and events using image analysis. However, we restricted our work on image analysis for the ubiquitous home, owing to the following reasons. Due to the constraints with disk space, they have a low frame rate (5 frames/second), low resolution (320x240), and a high degree of lossy (JPEG) compression with poor quality (15 kB/image). Automatic gain control is used so that there is maximum possible visibility in the captured images. Such images are sufficient for human observation upon retrieval, but not adequate for obtaining high accuracy using existing image

Table 18: Accuracy of action recognition after using multiple modalities.

| Label | Action | Precision | Recall | F-measure |
|---|---|---|---|---|
| 1 | Walking | 0.873 | 0.960 | 0.914 |
| 2 | Standing | 0.821 | 0.821 | 0.821 |
| 3 | Sitting on a chair | 0.500 | 0.625 | 0.556 |
| 4 | Sitting on the floor | 0.625 | 0.625 | 0.625 |
| 5 | Cooking or washing dishes | 0.939 | 0.920 | 0.929 |
| 6 | Vacuum cleaning | 1.000 | 0.714 | 0.833 |

Table 19: Confusion matrix after using multiple modalities.

| Correct action | Classified as | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 48 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 32 | 2 | 1 | 3 | 0 |
| 3 | 0 | 1 | 5 | 2 | 0 | 0 |
| 4 | 0 | 1 | 2 | 5 | 0 | 0 |
| 5 | 0 | 4 | 0 | 0 | 46 | 0 |
| 6 | 6 | 0 | 0 | 0 | 0 | 15 |

analysis algorithms. For example, face recognition is impossible given the resolution and compression. A pilot study on face detection using Viola-Jones feature detector [69] yielded a maximum accuracy of 86.5% [70]. However, this required scaling and smoothing of images, which need resources in terms of both processing power and disk space.

The actions and events that are detected using the above algorithms are fairly basic, and there is room for further work in action and event recognition. However, we stop at the current state, demonstrating the applicability of data driven selection of actions and features, supervised learning and sensor fusion for accurate retrieval.

Chapter 8

# User Interaction

## 8.1 Issues

The user retrieves video, audio and key frames through a graphical user interface. The main purpose of a user interface is to enable the users to accept queries from the users and present the results to them in a comprehensible manner. However, it is evident that there is a considerable difference between the semantic levels of the results obtained in Section 4 and the user queries listed in Section 1. Commonly known as "semantic gap" [71], this causes problems in most multimedia retrieval systems. If the gap is closed by making the users submit lower level queries, the usability of the system goes down. On the other hand, trying to fill the gap using heuristics or simple assumptions will result in lower accuracy. Our approach to solve this problem is twofold. From the side of the system, better visualization of events is provided using *hierarchical media segmentation.* At the same time, user intelligence is incorporated to the query process by means of interactive queries. The following subsections describe these two main concepts and the design of the user interaction in detail.

## 8.2 Approach

The user interaction was designed in two main stages while the research was in progress. First, a simple graphical user interface was designed to access video by selecting a date, time interval and a particular camera. More functionality added to this interface with the implementation of the algorithms in the previous chapters, using evolutionary

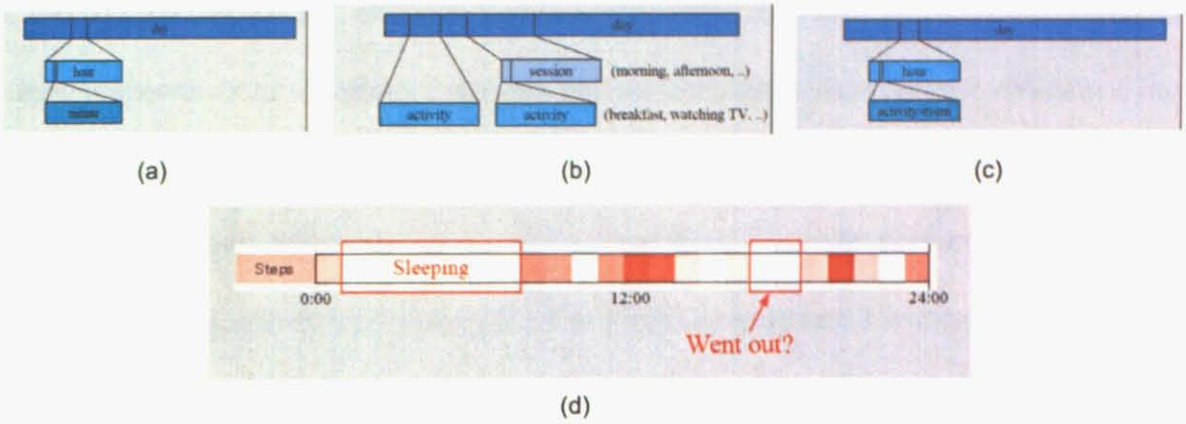



Figure 21: Hierarchical media segmentation.

prototyping technique [72]. After developing a prototype version that was used for a user study with residences (Chapter 9), the user interaction strategy with system was redesigned according to user feedback and based on the concepts described in the following sections.

## 8.3 Hierarchical Media Segmentation

The system captures multimedia data continuously, from which segments corresponding to events are retrieved as results. The original data from ubiquitous home are indexed by source (camera ID, microphone ID, floor sensors coordinates) and timestamp. The results obtained using the algorithms in Section 4 are indexed by the timestamp, location and event/action. Figure 21a presents the segmentation of data after capture and analysis.

However, these indices are still unable to facilitate efficient multimedia retrieval, as humans tend to remember the events and would like query for events that are segments in a different way. For example, "Retrieve video showing the regions house where people were in at 9:47 a.m." will be a valid query for the system, but is very unlikely for a human user to enter. A more likely query is "What was I doing after

breakfast last Saturday?". Humans tend to segment the time and experiences at home by days, sessions (e.g.: morning, late afternoon), locations, and events (Figure 21b). However, these are difficult to model using a computer-based system, due to high semantic level and fuzzy boundaries of the segments. We propose a trade-off between these two levels, shown by Figure 21c. The media is segmented hierarchically by date, hour, location and event. Visualization of a daily summary allows the users to identify sessions. For example, the user can identify the sessions of the day using the summary of floor sensor activity, sound and lighting level, as shown in Figure 21d. This method is an extension of the concept of *hierarchical timeline segmentation* [73].

## 8.4 Interactive Retrieval

The system, at its current state and with the available sensor data, is unable to perform some useful tasks (e.g. person recognition). Furthermore, the accuracy for the algorithms that are implemented is less than 100%. However, the performance can be improved greatly if it is possible to incorporate user's intelligence to the system. We propose interactive retrieval to achieve this. A query is broken down to a number of steps, and each step returns intermediate results to the user so that the user can provide further input navigating towards the desired results. For example, if the user wants to retrieve video showing what person *A* did during a given time interval, the system provides a key frame from each video sequence created by segmenting footsteps. The user can take a look at the key frames and identify those showing person *A*, resulting in accurate retrieval in the expense of one additional step.

## 8.5 User Interface Design

The user interface has been designed with the following objectives:

1. Ability to use with only a pointing device (either a tablet monitor or a touch monitor).

2. Require a minimum technical knowledge for understanding the inputs and results: all user-adjustable parameters will be interpreted to the user in a way they understand. For example, a slider control input labeled as "Sampling rate gradient for key frames" with range 0.0 to 1.0, has little meaning in the user's perspective. This can be modified by labeling the input as "Desired amount of key frames", with *few* and *many* labeling the ends of the slider.

3. Facilitate easy navigation within data without starting over: humans tend to search for *relative queries* such as "what happened next?" and "what happened in the other room during this time?". We attempt to facilitate such queries by dividing the results into a set of tabs and update the results in tabs according to the parameters submitted for the last query. While browsing the results, a user can navigate along the timeline outside the query boundaries, using button-based inputs such as *previous* and *next*.

The user interface is designed to have more intuitive inputs. For example, the users can click on an image showing the home layout, to select a room/region to retrieve events from. Camera selection is facilitated in the same manner, and the view from a camera is immediately available to the user so that the right camera can be selected

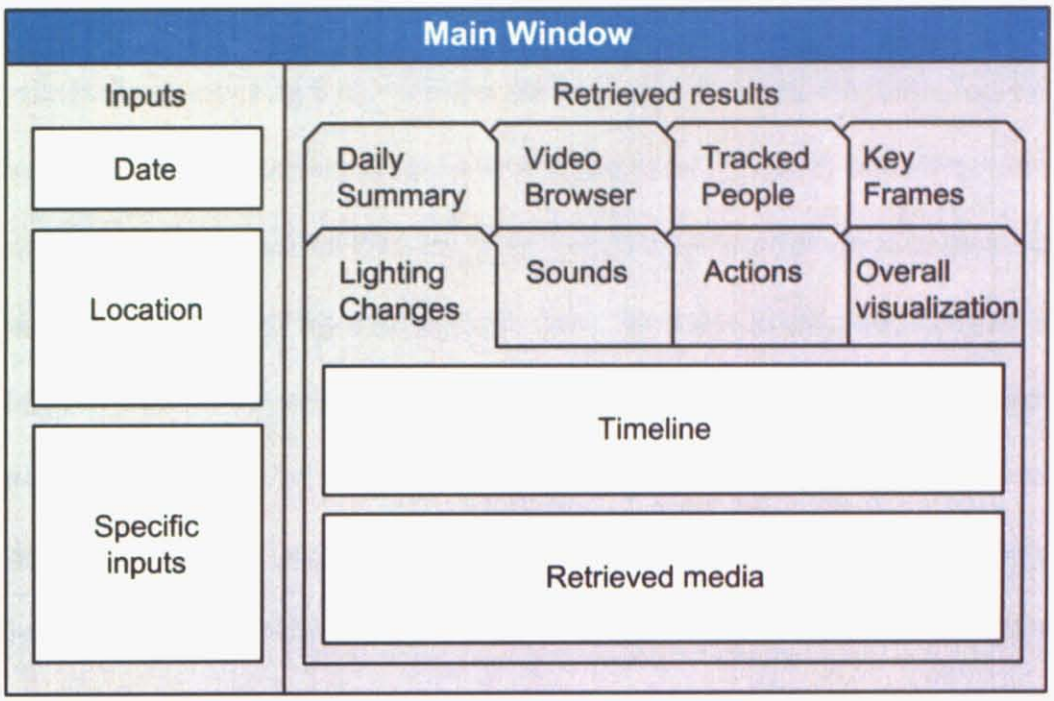



Figure 22: Organization of the user interface.

interactively. Instead of line or bar graphs, color levels are used to indicate the level of each activity as this is faster and easier to interpret.

## 8.6 Presentation and Visualization of results

Figure 22 shows the basic organization of the interface. The main window is divided into two main sections, as inputs and results. The input section includes the user inputs for date, location, and other specific options such as time intervals. The results for queries based on inputs on the left are grouped into tabs, as shown enlarged in Figure 22. All the tabs are updated at the same time when a user changes inputs, so it is possible to see different groups of results by merely clicking on the appropriate tab. The following subsections describe the presentation of each of these groups.

### *8.6.1 Daily summary*

The user starts by entering a day using a calendar interface, upon which a summary of the day's activity is displayed along the time line (Figure 23). Stripes of different colors, segmented in to one hour intervals, are used to represent different types of data captured and results retrieved during the selected date. On each graph, the strength of the corresponding color, indicates the amount of data/results present. White corresponds to no data/results, whereas the strongest color saturation indicates the maximum amount of data present during that day. Numbers are deliberately excluded to avoid information overloading. The colors used in all visualizations are scaled so that the results are shown to the user with maximum possible contrast. An example situation where this is useful is when all the lighting changes are less sharp during mid-day. Due to scaling of colors for visualization, the user is still able to see them clearly. The user can select the location of the house to retrieve the summary from, if necessary, using the inputs on the left.

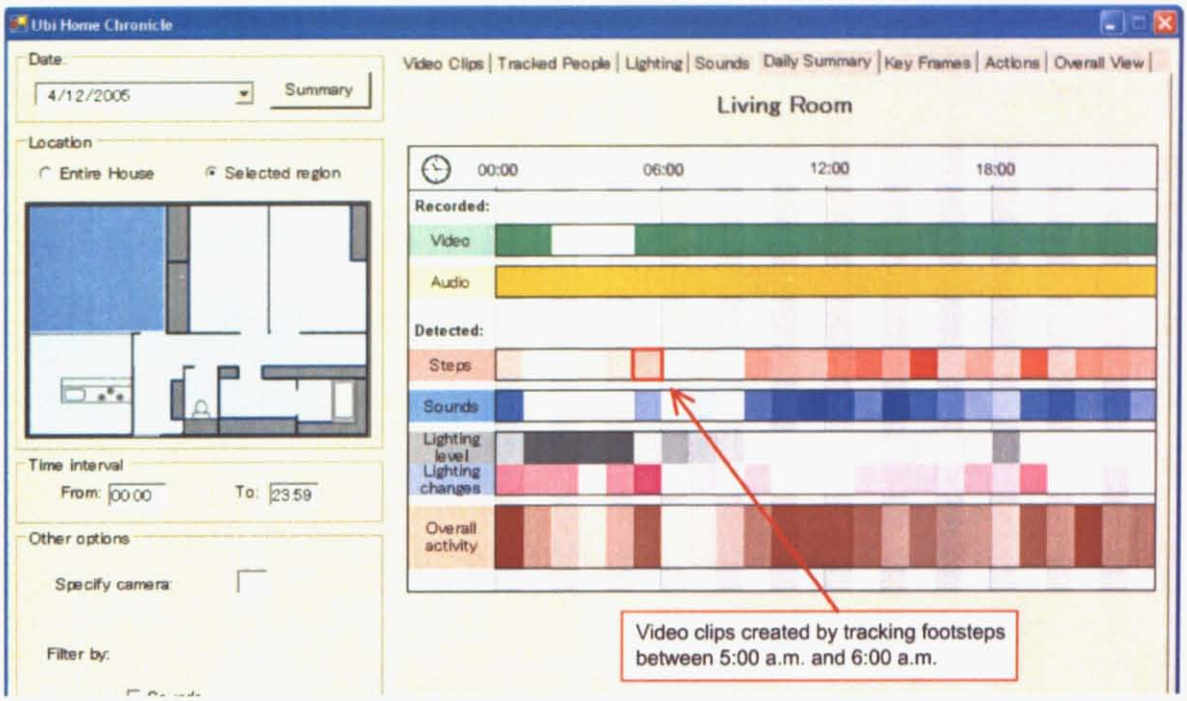
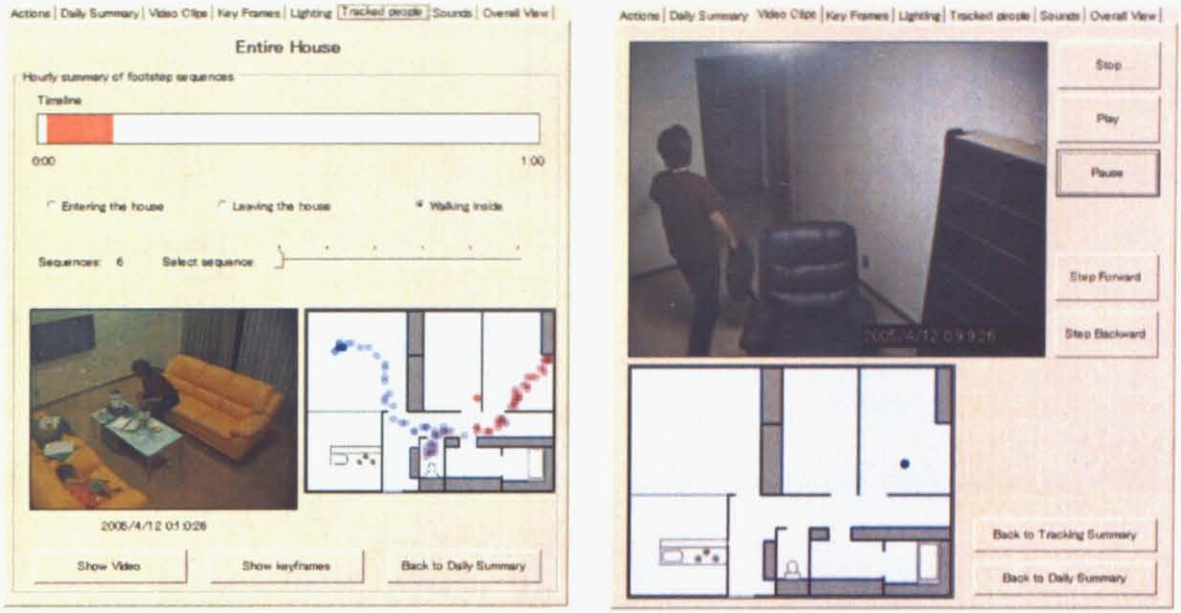



Figure 23: Visualization of the daily summary.

Figure 23 is a screenshot of the daily summary for the living room, on the12$^{th}$ of April 2005. It is possible to interpret a fair amount of activity by simply observing the daily summary. It is evident that video has not been captured between 2:00a.m. and 5:00 a.m. on this date. Looking at the graphs labeled "Steps" and "Sounds", it can be deduced that the residents have slept after midnight and before 1:00 a.m. It can also be seen that a large number of footsteps in the living room have been detected between 3:00 p.m. and 4:00 p.m. This can be due to having visitors (in this case the actual reason), children playing, or a special event such as a party.

The user can click on any one-hour block on the daily summary, to see the selected type of results for that one hour. For example, clicking on the block marked with a red border in Figure 23 will show a summary of media for "tracked people" in the house between 5:00 a.m. and 6:00 a.m., which can be browsed further to see the




(a) (b)

Figure 24: Viewing video for tracked people.

results from personalized video retrieval. The following subsections describe this and similar options that are available to the user.

### *8.6.2 Tracked people*

Figure 24a shows the retrieval of footstep sequences for a selected duration of one hour. The timeline now shows only this duration. The User can select footstep sequences corresponding to instances of persons entering the house, leaving the house or walking inside. The resulting sequences can be previewed one at a time, using the slider control. The timeline shows the duration of the selected sequence. This can be modified further to indicate the type of action the person is performing, using different colors. The preview image shows the first frame of the video clip created for this sequence. The dots on the house floor plan show the path of the person's footsteps. The color of the dots changes from blue to red with time, indicating the direction of the person. Thus the

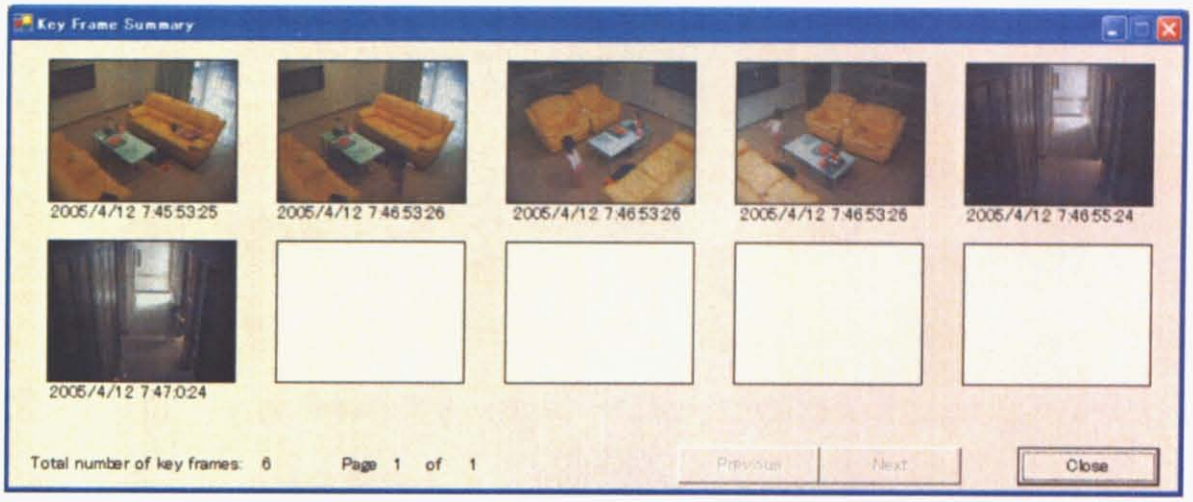



Figure 25: Displaying key frame sets.

user can interpret the video sequence to a certain extent even before fully viewing it, and thereby find the desired results faster. After selecting the desired video clip, the user can play it at normal speed, or browse it using the video clip viewer in Figure 24b. A moving dot on the house plan will now show the location of the person.

### *8.6.3 Key frames*

The user can choose to view a sequence of key frames after selecting a footstep sequence from the preview window describes in Section 7.6.2. The key frames extracted using the adaptive spatio temporal sampling algorithm are shown to the user (Figure 25).

### *8.6.4 Sounds*

The main elements of the preview window for video retrieved for sounds are the same as for the footstep sequences. The duration of each sound segment or class is shown along the timeline when the user selects it (Figure 26). After selecting the desired sound clip, the user can retrieve video from the cameras in the region where the sound was

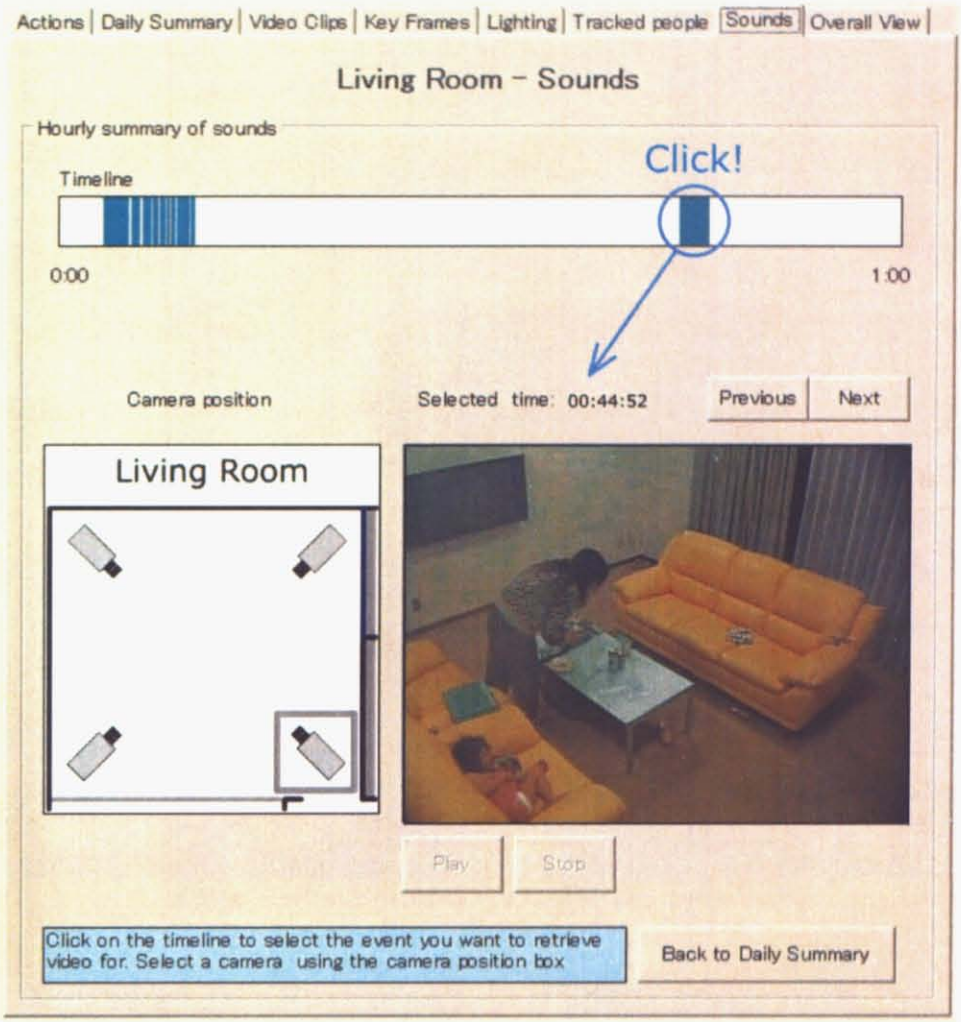
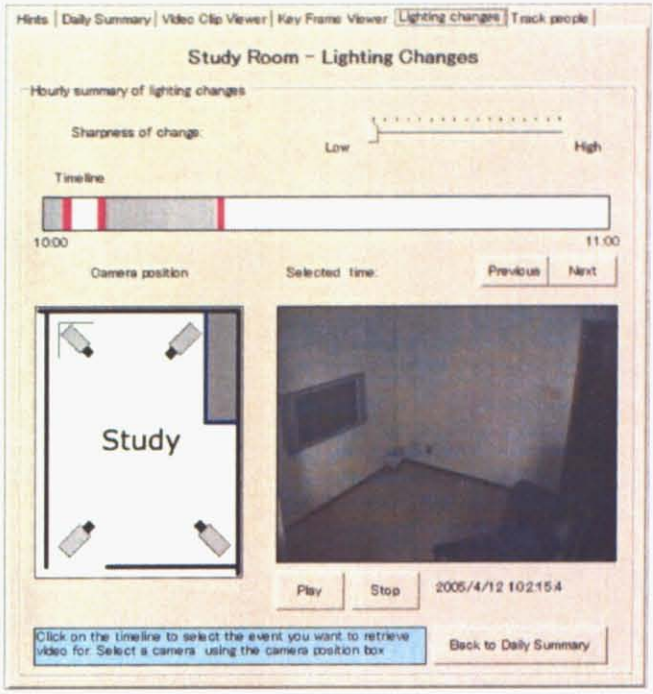
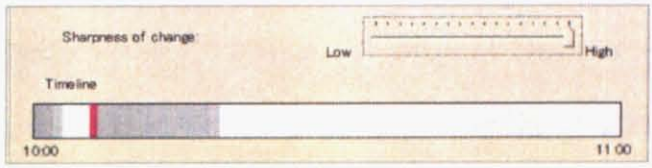



Figure 26: Retrieving video for sound segments.

generated. The desired camera can be selected interactively while watching the video, by clicking on the appropriate camera drawn on the floor plan of the region.




(a)




(b)

Figure 27: Interactive retrieval of lighting change events.

### *8.6.5 Lighting change events*

Figure 27 illustrates the use of interactive retrieval to retrieve video for events that cause lighting changes. The light level in the region, scaled over the given date, is shown on the time line so that the lighting changes can be interpreted easily. The slider labeled as "sharpness of change" is coupled to a threshold within the range of the rank function. When this set to minimum, all events are displayed on the time-line (Figure 27a). When set to maximum, only the event that has the maximum rank during the selected one hour time interval is displayed (Figure 27b). Selection of the camera for

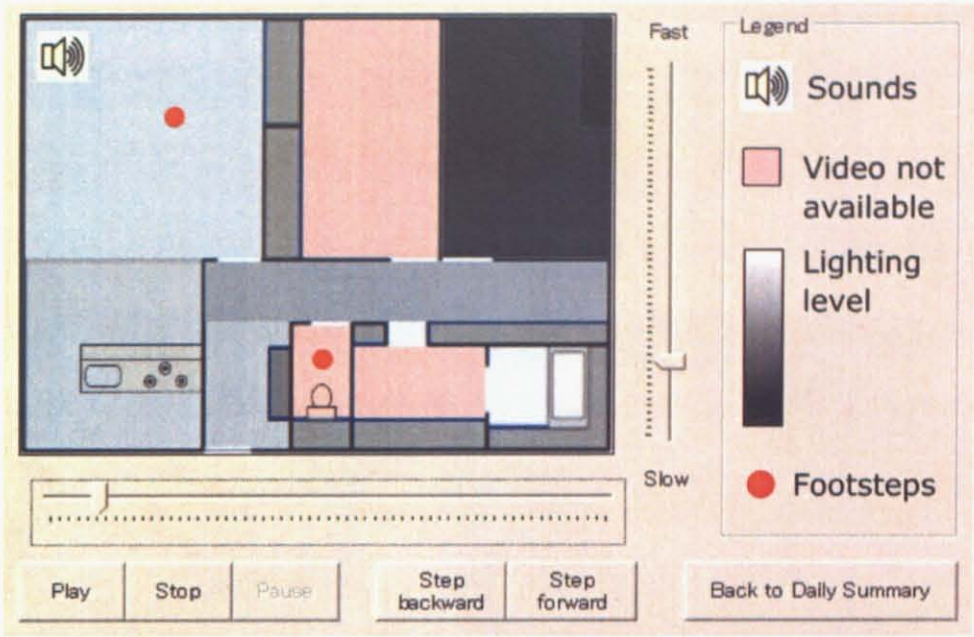



Figure 28. Animated preview of overall activity.

viewing is done interactively by clicking on the image of the camera on the room layout, thereby providing an intuitive method to select the viewpoint (Figure 27a).

### *8.6.6 Overall activity visualization*

The interface components describe above display results of one category, most of the time from one location. However, it is desirable to have a simultaneous overview of what is happening in the entire house, if possible. The main problem in achieving this is finding a method to visualize the various types of data from a large number of sources, without overloading the user with the data. For instance, there are 17 cameras in ubiquitous home, recording data continuously. If the videos from these cameras are displayed concurrently, it will be extremely difficult for the user to focus on everything and get a general idea of what happened. The same applies for audio data, too.

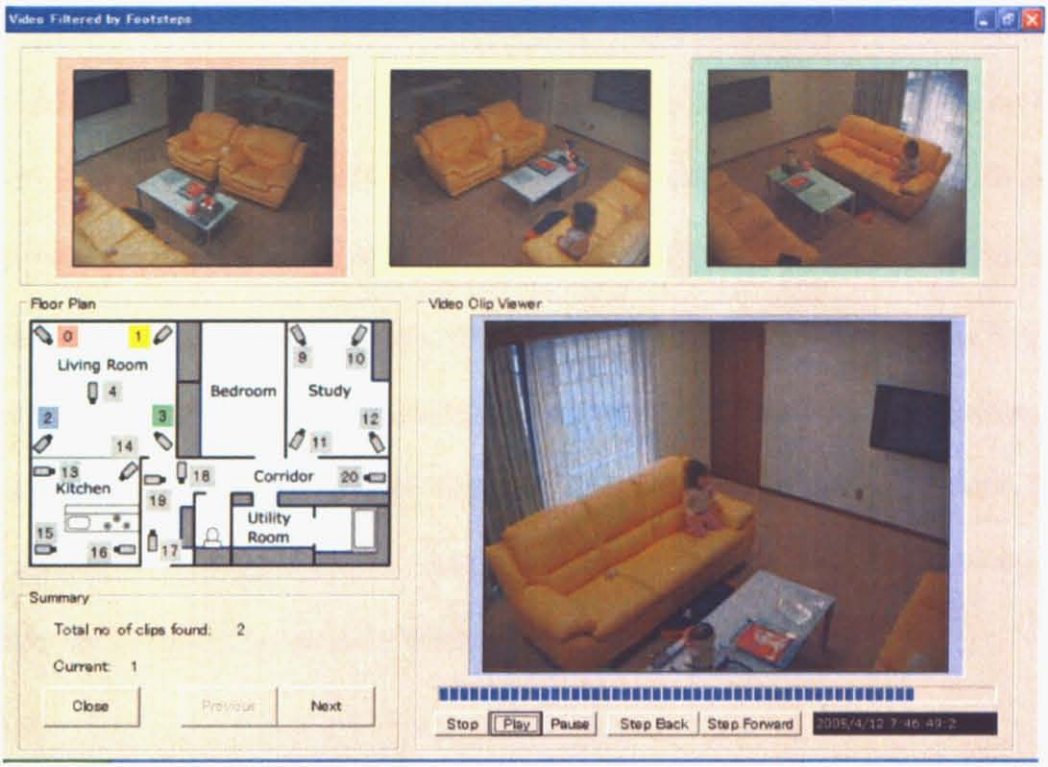



Figure 29: Video browser for multi-camera preview.

We propose a simultaneous visualization of the desired results, to facilitate an overall visualization of activity inside ubiquitous home. Figure 28 shows a screenshot of this visualization. The proposed visualization is an animation with a timestamp, where symbols are displayed plan of the house to visualize different types of activity. The brightness of each region in the plan corresponds to the actual light level in that region of the house. Dots are displayed to represent floor sensor activations. A speaker is displayed in each region when there is a local sound segment from that region. The speed of the animation can be changed so that it can be browsed faster to see what happened.

### *8.6.7 Video browser*

The video browser is an extension of the simple form of a multi-camera surveillance system. The user can select a location and a time interval to retrieve video from. It is also possible to select a *main camera* for retrieving images. The video from this camera will be shown larger whereas video from other cameras in the same region will be shown in smaller size. Unlike in a normal video surveillance system, the resulting video can be filtered to retrieve video for situations only with sound and/or footsteps. Figure 29 is a screenshot of the video browser. The colored border around each picture frame indicates the camera it is retrieved from.

## 8.7 Example Scenario of Retrieval

Suppose Mrs. Taro wants to see what her son, Takeshi, did after coming back from school on a particular date and before leaving for sports practices in the evening. Also, suppose she remembers that Takeshi came back home house some time after 2:00 p.m. and left home around 3:00 p.m. For a conventional retrieval system based on date, time and camera, it is necessary to watch the video from the camera showing the entrance to the house from 2:00 p.m., until the frames showing him entering the house are detected. Thereafter, it is necessary to switch between several cameras to track him as he moves within the house. This becomes increasingly tedious with movement and the duration of Takeshi's stay.

The proposed system can be used for this query, in a much simpler manner. First, Mrs. Taro enters the particular date and retrieves the daily summary. On the "Footsteps" graph of the summary, she can click on the one-hour block corresponding to the duration between 2:00 p.m. and 3:00 p.m. After selecting "Entering the house", she can

see the set of preview frames showing people who entered the house during this time interval. By browsing these preview frames showing the persons entering the house, Mrs. Sato can find the key frame showing Takeshi. She can now see either the complete video clip showing what Takeshi did inside the house. Since the cameras and microphones are selected automatically, no further user input is required. If necessary, Mrs. Sato can view a set of key frames instead of the complete video, achieving even faster retrieval.

## 8.8 Discussion

Designing user interaction and developing a graphical user interface is a challenging and iterative task, as the user feedback often includes requests for increased functionality. While a badly designed interface requires changes, a good and usable interface stimulates imagination of the users and results in requests for added functionality. Therefore, it is essential to carry out evaluations regularly while adding new retrieval algorithms to the system. However, for the ubiquitous home, it is quite difficult to use residents as subjects for user studies regularly. Therefore we conducted a single, extensive user study with residents and used the results for improving the system.

The results from some of the stages of data analysis are yet to be integrated to the current version of the user interface. This is mainly due to the parallel development of both the user interface and the algorithms for data analysis and multimedia retrieval. However, this could be completed by merely adding database queries and program modules to display the results, without any research effort. Furthermore, the design of user interaction is sufficiently modular so that new types of queries and visualizations can be added to the interface without changes to the current system.

The system could benefit from using different input devices and output visualizations. In a 5-day demonstration session with a large number of users, we found that users found the system more natural to browse using a graphic pen tablet monitor, instead of the conventional keyboard and mouse. While the current two-dimensional visualizations are easy to use and informative, three-dimensional visualizations with improved navigation capabilities are worth investigating, due to the large content and three-dimensional nature of the actual environment.