6

Surveillance Using Both Video and Audio

Yigithan Dedeoglu[1], B. Ugur Toreyin[2], Ugur Gudukbay[1], and A. Enis Cetin[2]

[1] Department of Computer Engineering, Bilkent University
[2] Department of Electrical and Electronics Engineering, Bilkent University

It is now possible to install cameras monitoring sensitive areas but it may not be possible to assign a security guard to each camera or a set of cameras. In addition, security guards may get tired and watch the monitor in a blank manner without noticing important events taking place in front of their eyes. Current CCTV surveillance systems are mostly based on video and recently intelligent video analysis systems capable of detecting humans and cars were developed for surveillance applications. Such systems mostly use Hidden Markov Models (HMM) or Support Vector Machines (SVM) to reach decisions. They detect important events but they also produce false alarms. It is possible to take advantage of other low cost sensors including audio to reduce the number of false alarms. Most video recording systems have the capability of recording audio as well. Analysis of audio for intelligent information extraction is a relatively new area. Automatic detection of broken glass sounds, car crash sounds, screams, increasing sound level at the background are indicators of important events. By combining the information coming from the audio channel with the information from the video channels, reliable surveillance systems can be built. In this chapter, current state of the art is reviewed and an intelligent surveillance system analyzing both audio and video channels is described.

6.1 Multimodal Methods for Surveillance

Multimodal methods have been successfully utilized in the literature to improve the accuracy of automatic speech recognition, human activity recognition and tracking systems [355, 595]. Multimodal surveillance techniques are discussed in a recent edited book by Zhu and Huang [595]. In [597], signals from an array of microphones and a video camera installed in a room are analyzed using a Bayesian based approach for human tracking. A speech recognition system comprising of a visual as well as a audio processing unit is proposed in [147]. A recent patent proposes a coupled hidden Markov model based method for audiovisual speech recognition [361].

P. Maragos et al. (eds.), *Multimodal Processing and Interaction*,
DOI: 10.1007/978-0-387-76316-3_6,

Other abnormal human activities including falling down can also be detected using multimodal analysis. In [525], an audiovisual system is proposed for fall detection. They use wavelet based features for the analysis of audiovisual content of video. Detection of the falling people is achieved by HMM based classification. An alternative method using several other sensor types is described in [526]. Audio, Passive Infrared (PIR) and vibration sensors installed in a room are used to detect falling people in [526].

Similar to background/foreground segmentation in video applications, Cristani et al. propose an adaptive method to build background and foreground models for audio signals [115]. The method is based on the probabilistic modeling of audio channel with adaptive Gaussian mixture models. In another work, authors extend their unimodal audio based background/foreground analysis and event detection system to an audiovisual (AV) based one [116]. They propose a method to detect and segment AV events based on the computation of the so-called "audio-video concurrence matrix".

Zhang et al. propose an approach to automatic segmentation and classification of audiovisual data based only on audio content analysis [591]. They use simple audio features like the energy function, the average zero-crossing rate, the fundamental frequency, and the spectral peak tracks for real-time processing. A heuristic rule-based procedure is proposed to segment and classify audio signals and built upon morphological and statistical analysis of the time-varying functions of these audio features.

Nam et al. present a technique to characterize and index violent scenes in general TV drama and movies by identifying violent signatures and localize violent events within a movie [354]. Their method detects abrupt changes in audio and video signals by computing energy entropies over time frames.

6.2 Multimodal Method for Detecting Fight among People in Unattended Places

Detecting fighting people in unattended places is an important task to save lives and protect properties. Today most of the public places are under continuous surveillance with cameras. However, the recordings are generally saved to tapes for later use only after a forensic event. With the help of low cost digital signal processing systems surveillance video can be processed online to trigger alarms in case of violent behavior and unusual events. This will help to reduce the time it takes to respond to such events that threaten public safety and will prevent casualties. Performances of the video processing algorithms generally degrade due to the inherent noise in the video data and/or camera motion due to wind etc. Hence, it is desirable to support the decision systems with other sensors such as audio to increase the success rate.

In this chapter, we present a system using both video and audio providing information about violent behavior in a scene monitored by a camera and a microphone. Both of the sensor channels are processed in real-time and

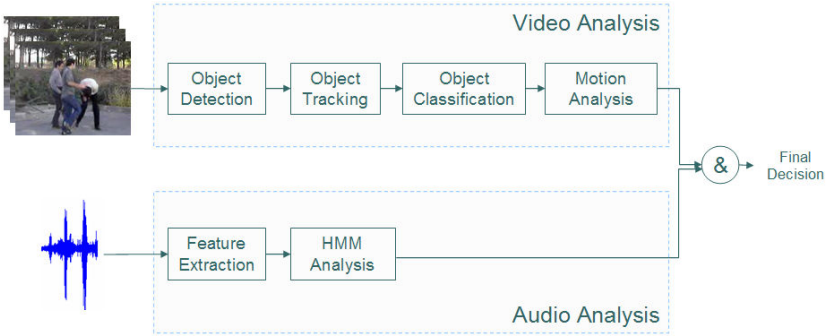

Fig. 6.1. Overview of multimodal fight detection system.

their output are fused together to give the final decision. The overview of our method is shown in Fig. 6.1. In this multimedia system video and audio is processed independently and their decision results of individual processing systems are fused to reach a final decision. In the next subsections we discuss the video processing part of the proposed system.

6.2.1 Video Analysis

Video processing unit of the system analyzes the motion characteristics of humans present in the monitored area in real-time to detect a fight event. In order to accomplish this we need fast and reliable algorithms to detect humans and analyze their actions. In our method, first, moving objects are segmented from the scene background by using an adaptive background subtraction algorithm and then segmented objects are classified into groups like vehicle, human and human group using a silhouette based feature and SVM classification method. After distinguishing humans from other objects, we analyze the motion of the human groups. In case of a fight or a violence, the limbs of the people involved in the violent action generate high frequency motion characteristic. We decide on a fight if the motion characteristic of a human group matches that of a fight action.

Learning Scene Background for Segmentation

There are various methods for segmenting moving objects in video. Background subtraction, statistical methods, temporal differencing and optical flow techniques are commonly used ones. For a discussion on the details of these methods, the reader is referred to [126]. One of the statistical methods extensively used due to its ability to robustly deal with lighting changes, repetitive motions, clutter, introducing or removing objects from the scene and slowly moving objects is presented by Stauffer et al. [505]. It uses a mixture of Gaussian models to represent each pixel on the video stream. Although

this method gives good results, it is computationally more demanding than an adaptive background subtraction method. Hence, we use a combination of a background model and low-level image post-processing methods to create a foreground pixel map and extract object features at every video frame. Our implementation of adaptive background subtraction algorithm is partially inspired by the study presented in [103] and works on grayscale video imagery from a static camera. Background subtraction method initializes a reference background with the first few frames of video input. Then it subtracts the intensity value of each pixel in the current image from the corresponding value in the reference background image. The difference is filtered with an adaptive threshold per pixel to account for frequently changing noisy pixels. The reference background image and the threshold values are updated with an Infinite Impulse Response (IIR) filter to adapt to dynamic scene changes. Let $I_n(x)$ represent the gray-level intensity value at pixel position (x) and at time instance n of video image sequence I which is in the range [0, 255]. Let $B_n(x)$ be the corresponding background intensity value for pixel position (x) estimated over time from video images I_0 through I_{n-1}. As the generic background subtraction scheme suggests, a pixel at position (x) in the current video image belongs to foreground if it satisfies:

$$|I_n(x) - B_n(x)| > T_n(x) \tag{6.1}$$

where $T_n(x)$ is an adaptive threshold value estimated using the image sequence I_0 through I_{n-1}. The above equation is used to generate the foreground pixel map which represents the foreground regions as a binary array where a 1 corresponds to a foreground pixel and a 0 stands for a background pixel. The reference background $B_n(x)$ is initialized with the first video image I_0, $B_0 = I_0$, and the threshold image is initialized with some predetermined value (e.g., 15).

Since this system will be used in outdoor environments as well as indoor environments, the background model needs to adapt itself to the dynamic changes such as global illumination change (day night transition) and long term background update (parking a car in front of a building). Therefore the reference background and threshold images are dynamically updated with incoming images. The update scheme is different for pixel positions which are detected as belonging to foreground ($x \in FG$) and which are detected as part of the background ($x \in BG$):

$$B_{n+1}(x) = \begin{cases} \alpha B_n(x) + (1-\alpha) I_n(x), & x \in BG \\ \beta B_n(x) + (1-\beta) I_n(x), & x \in FG \end{cases} \tag{6.2}$$

$$T_{n+1}(x) = \begin{cases} \alpha T_n(x) + (1-\alpha)(\gamma \times |I_n(x) - B_n(x)|), & x \in BG \\ T_n(x), & x \in FG \end{cases} \tag{6.3}$$

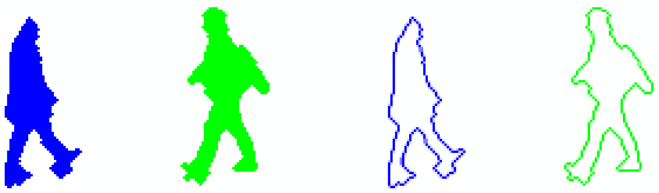

Fig. 6.2. Detected regions and silhouettes. *Left*: Detected and labeled object regions. *Right*: Extracted object silhouettes.

where α, β ($\in$ [0.0, 1.0]) are learning constants which specify how much information from the incoming image is put to the background and threshold images.

The output of foreground region detection algorithm generally contains noise and therefore is not appropriate for further processing without special post-processing. Morphological operations, erosion and dilation [203], are applied to the foreground pixel map in order to remove noise that is caused by the first three of the items listed above. Our aim in applying these operations is to remove noisy foreground pixels that do not correspond to actual foreground regions and to remove the noisy background pixels near and inside object regions that are actually foreground pixels.

Calculating Object Features

After detecting foreground regions and applying post-processing operations to remove noise and shadow regions, the filtered foreground pixels are grouped into connected regions (blobs) and labeled by using a two-level connected component labeling algorithm presented in [203]. After finding individual blobs that correspond to objects, spatial features like bounding box, size, center of mass and silhouettes of these regions are calculated. In order to calculate the center of mass point, $c_m = (x_{c_m}, y_{c_m})$, of an object O, we use the following formula [463]:

$$x_{c_m} = \frac{\sum_i^n x_i}{n}, \quad y_{c_m} = \frac{\sum_i^n y_i}{n} \tag{6.4}$$

where n is the number of pixels in O. Both in offline and online steps of the classification algorithm, the silhouettes of the detected object regions are extracted from the foreground pixel map by using a contour tracing algorithm presented in [203]. Fig. 6.2 shows sample detected foreground object regions and the extracted silhouettes. Another feature extracted from the object is

the silhouette distance signal. Let $S = \{p_1, p_2, \ldots, p_n\}$ be the silhouette of an object O consisting of n points ordered from top center point of the detected region in clockwise direction and c_m be the center of mass point of O. The distance signal $DS = \{d_1, d_2, \ldots, d_n\}$ is generated by calculating the distance between c_m and each p_i starting from 1 through n as follows:

$$d_i = Dist(c_m, p_i), \quad \forall\ i \in [1 \ldots n] \tag{6.5}$$

where the $Dist$ function is the Euclidean distance between two points a and b.

Different objects have different shapes in video and therefore have silhouettes of varying sizes. Even the same object has altering contour size from frame to frame. In order to compare signals corresponding to different sized objects accurately and to make the comparison metric scale-invariant we fix the size of the distance signal. Let N be the size of a distance signal DS and let C be the constant for fixed signal length. The fix-sized distance signal $\widehat{DS}$ is then calculated by sub-sampling or super-sampling the original signal DS as follows:

$$\widehat{DS}[i] = DS[i * \frac{N}{C}], \quad \forall\ i \in [1 \ldots C] \tag{6.6}$$

In the next step, the scaled distance signal $\widehat{DS}$ is normalized to have integral unit area. The normalized distance signal $\overline{DS}$ is calculated with the following equation:

$$\overline{DS}[i] = \frac{\widehat{DS}[i]}{\sum_1^n \widehat{DS}[i]} \tag{6.7}$$

Fig. 6.3 shows a sample silhouette and its original and scaled distance signals. Before using Support Vector Machine algorithm, we transformed contour signals to frequency domain in order to both reduce the amount of data for representation of objects and gaining robustness against rotation. The characteristic features of most objects are hidden in the lower frequency bands of contour signals. We used three different transformations, Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT) and block wavelet. From FFT, DCT and wavelet transformations, we take the first 3-15 coefficients except the first coefficient and use these coefficients as the feature vector while training and testing an SVM.

Classifying Objects

In order to detect fight among people, we need to identify humans in a scene and especially we need to detect the formation of human groups. Categorizing the type of a detected video object is a crucial step in achieving this goal. The process of object classification method consists of two steps:

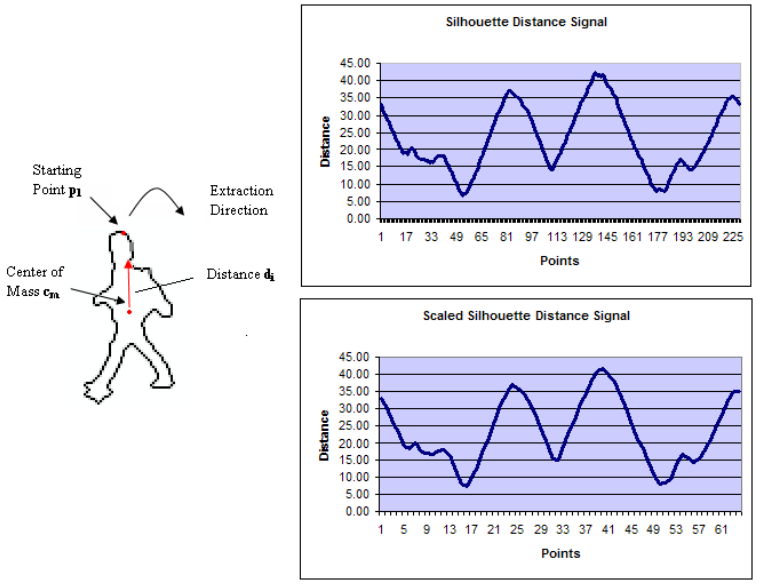

Fig. 6.3. Feature extraction from object silhouette. *Left*: Silhouette extraction. *Upper-Right*: Calculated distance signal, DS. *Lower-Right*: Scaled distance signal, $\widetilde{DS}$.

- Offline step: A template database of sample object silhouettes is created by manually labeling object types (one from human, human group and vehicle) and an SVM model is created using the features obtained from the sample objects as explained in previous section.
- Online step: The silhouette of each detected object in each frame is extracted and its type is recognized by using the SVM trained using the sample objects in offline step.

Detecting Fight

During a fight and especially when a person is hitting another person, whole-body displacement is relatively small whereas motion of the limbs of the people is high. Hence, we analyze the motion track of a human group and the motion inside the moving region of the human group. Sample silhouettes of people during a fight are shown in Fig. 6.4.

For each object region R we calculate the number of moving pixels $\in R$ by using frame differencing method to approximate the motion inside the region. The pixels which satisfy the following condition are considered as moving:

$$|I_n(x) - I_{n-1}(x)| > T_n(x) \tag{6.8}$$

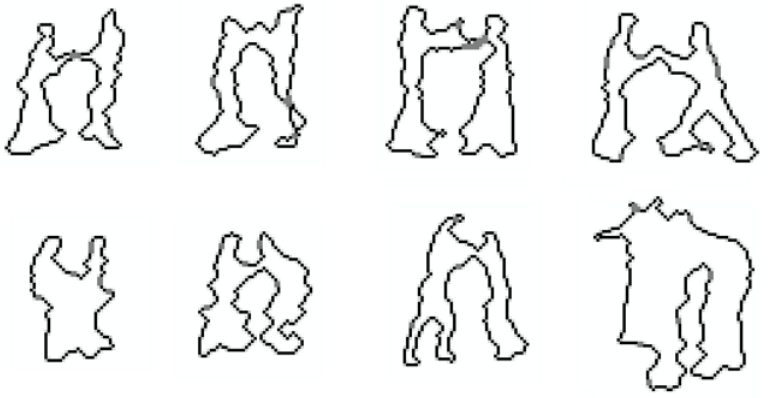

Fig. 6.4. Silhouettes of people during a fight.

where I_n and T_n correspond to image frame and adaptive threshold at time n respectively as explained in Section 6.2.1. Let α_R be the ratio of the number of moving pixels to the total number of pixels inside the region R. Then for object regions where $\alpha_R \geq \gamma$ a violent action is possible, where γ is a threshold constant obtained by tests.

6.2.2 Audio Analysis

In a typical surveillance environment, microphones can be placed near the cameras. Audio signals captured by sound sensors can be used to detect screams in audio stream as a possible indication of violent actions. Shouting has a high amplitude, non-stationary characteristic sound, whereas talking has relatively lower amplitude peakiness. Typical shouting and talking audio recording samples are shown in Fig. 6.5. In this case, the two sound waveforms are clearly different from each other. However, these waveforms may "look" similar as the distance from the sensor increases. For some other cases such as when there is background noise it may become even harder to distinguish different sound activities. In addition, the difference between these two types of signals becomes obvious after wavelet domain signal processing.

Typically audio recordings are due to regular chatting between people and background noise. When there is shouting or broken glass sounds etc this indicates an unusual event. Significantly loud voice or sound activity is detected using the Teager Energy operator based speech features originally developed by Jabloun and Cetin [229]. The sound data is divided into frames as in any speech processing method and the Teager energy based cepstral (TEOCEP) [229] feature parameters are obtained using wavelet domain signal analysis. The sound signal is divided into 21 non uniformly divided subbands

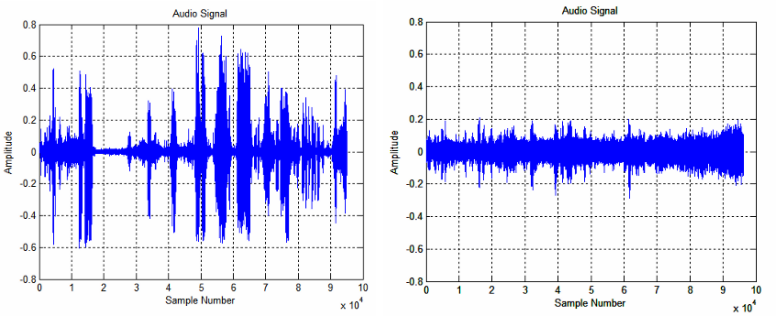

Fig. 6.5. Sample audio signals. *Left*: Shouting signal. *Right*: Talking signal.

similar to the Bark scale (or mel-scale) giving more emphasis to low-frequency regions of the sound.

To calculate the TEOCEP feature parameters, a two-channel wavelet filter bank is used in a tree structure to divide the audio signal $s(n)$ according to the mel-scale as shown in Fig. 6.6, and 21 wavelet domain sub-signals $s1(n)$, $l = 1, \ldots, L = 21$, are obtained [152]. The filter bank of a biorthogonal wavelet transform is used in the analysis [259]. The lowpass filter has the transfer function

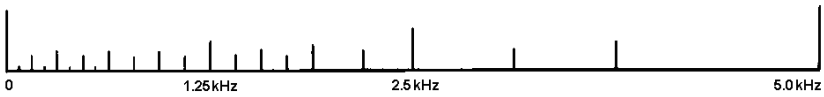

Fig. 6.6. The subband frequency decomposition of the sound signal.

$$H_l(z) = \frac{1}{2} + \frac{9}{32}(z^{-1} + z^1) - \frac{1}{32}(z^{-3} + z^3) \tag{6.9}$$

and the corresponding high-pass filter has the transfer function

$$H_h(z) = \frac{1}{2} - \frac{9}{32}(z^{-1} + z^1) + \frac{1}{32}(z^{-3} + z^3) \tag{6.10}$$

For every subsignal, the average Teager energy e_l is estimated as follows:

$$e_l = \frac{1}{N_l} \sum_{n=1}^{N_l} |\Psi[s_l(n)]| \,; \qquad l = 1, \ldots, L \tag{6.11}$$

where N_l is the number of samples in the l^{th} band, and the Teager energy operator (TEO) is defined as follows:

$$\Psi[s(n) = s^2(n) - s(n+1)s(n-1) \tag{6.12}$$

The TEO-based cepstrum coefficients are obtained after log-compression and inverse DCT computation as follows:

$$TC(k) = \sum_{l=1}^{L} \log e_l \cos\left[\frac{k(l-0.5)\pi}{L}\right]; \qquad k = 1, \ldots, N \tag{6.13}$$

The first 12 $TC(k)$ coefficients are used in the feature vector. The TEO-CEP parameters are fed to the sound activity detector algorithm described in [525] to detect significant sound activity in the environment.

When there is significant sound activity in the room, another feature parameter based on variance of wavelet coefficients and zero crossings is computed in each window. The wavelet signal corresponding to the [2.5 kHz, 5.0 kHz] frequency band is obtained after a single stage wavelet filterbank. The variance, σ_i^2 of the wavelet signal and the number of zero crossings, Z_i, in each window i is computed.

Broken glass and similar sounds are not quasi-periodic in nature. As talking is mostly quasi-periodic because of voiced sounds the zero crossing value, Z_i, is small compared to noise like sounds. When a person shouts the variance of the wavelet signal σ_i^2 increases compared to the background noise and regular chatting. So we define a feature parameter κ_i in each window i as $\kappa_i = \frac{\sigma_i^2}{Z_i}$, where the index i indicates the window number. The parameter κ_i takes non-negative values.

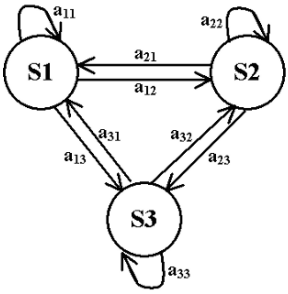

Fig. 6.7. Three state Markov model. Three Markov models are used to represent speech, walking, and fall sounds.

Activity classification based on sound information is carried out using HMMs. Three three-state Markov models are used to represent shout and talking sounds as shown in Fig. 6.7. In Markov models, S1 corresponds to the background noise or no activity. If sound activity detector (SAD) indicates

that there is no significant activity, S1 is selected. If SAD detects sound activity in a sound frame, then either S2 or S3 is chosen as the current state according to the value of κ.

A non-negative threshold value, T, that is small enough to reflect the periodicity in step sounds is introduced in the κ-domain. If $|\kappa| < T$, S2; otherwise, S3 is attained as the current state. The classification performance of HMMs is based on the number of state transitions, rather than specific κ values. Hence, choice of T does not affect the values of the transition probabilities in different models.

In order to train HMMs, the state transition probabilities are estimated from 20 consecutive κ_i values corresponding to 20 consecutive 500-sample-long wavelet windows covering 125 msec of audio data.

During the classification phase a state history signal consisting of 20 κ_i values are estimated from the sound signal acquired from the audio sensor. This state sequence is fed to Markov models corresponding to shouting and talking cases in running windows. The model yielding the highest probability is determined as the result of the analysis of the sound sensor signal.

Feature parameter κ takes high values for a regular speech sound. Consequently, the value of a_{33} is higher than any other transition probabilities in the talking model. For the shout case, a relatively long noise period is followed by a sudden increase and then a sudden decrease in κ values. This results in higher a_{11} value than any other transition probabilities. In addition to that, the number of transitions within, to and from S2 are notably fewer than those of S1 and S3. The state S2 in the Markov models provides hysteresis and it prevents sudden transitions from S1 to S3 or vice versa, which is especially the case for talking.

6.2.3 Deciding on a Fight or Violent Behavior

Audio and video analysis results are combined at each frame of the video sequence in the proposed system. In order to accomplish this audio sample frames are matched to video frames by combining the video frame rate and audio signal sampling rate.

For each time frame, video processing results and audio processing results are combined with the simple AND operator and the final decision is given on the result of this operator. In other words, we require both of the processing channels to decide on fight action to raise an alarm.

6.2.4 Experimental Results

Experimental results on sample video sequences containing violent action scenarios are presented in this section. All of the tests are performed by using a video player application, vPlayer, on Microsoft Windows XP Professional operating system on a computer with an Intel Pentium dual core 3.2GHz CPU and 1 GB of RAM.

	Human	Human Group	Vehicle	Success Rate
Human	20	0	0	100 %
Human Group	1	18	1	90 %
Vehicle	0	2	18	90 %

Table 6.1. Confusion matrix for object classification.

We tested our object classification algorithm on sample video clips. After collecting sample silhouettes and applying transformations to create feature vectors, we obtained an SVM model. In support vector machine training algorithm, we used Radial Basis Function (RBF) kernel with gamma=1, and tested the performance of SVM algorithm with different types transformations (FFT, DCT and wavelet) and different coefficient numbers. In our tests, feature vectors obtained by wavelet transform slightly outperformed other transformation methods. We used three object classes in our tests: human, human group and vehicle. We used 248 random objects for training the SVM model (93 human, 102 human group and 53 vehicle pictures) and 60 different objects (20 from each group) for testing the algorithm. The confusion matrix is shown in Table 6.1.

We also tested our multi-model fight detection algorithm on sample video clips with audio. The results are shown in Tables 6.2, 6.3 and 6.4 as confusion matrices. In both audio-only and video-only processing, the success rate for detecting fight is high. However, the false alarm rate which is calling a normal action as fight is high in both cases. When we fuse the results of these two channels together we get a lower false alarm rate and thus a higher average success rate.

	Fight	Normal	Success Rate
Fight	13	2	86.7 %
Normal	8	13	61.9 %
Average			74.3 %

Table 6.2. Confusion matrix for multimodal fight detection using both Audio & Video.

6.3 Conclusion

In this chapter, we proposed a novel multimodal system for real-time violence detection. Video data and audio signals are analyzed independently with the proposed algorithms and the analysis results from these two signals are fused together to reach a final decision. The test results show that the presented

	Fight	**Normal**	**Success Rate**
Fight	14	1	93.3 %
Normal	10	11	52.3 %
Average			72.9 %

Table 6.3. Confusion matrix for multimodal fight detection using only Audio.

	Fight	**Normal**	**Success Rate**
Fight	13	2	86.7 %
Normal	9	12	57.1 %
Average			71.2 %

Table 6.4. Confusion matrix for multimodal fight detection using only Video.

method is promising and can be improved with some further work to reduce false alarms. The use of audio signals in parallel with video analysis helps us to detect violent behavior with less false alarms.

A weakness of the proposed video analysis algorithm is that it is view dependent. If the camera setup is different in training and testing, the success rate will be lower. Automating video object classification method with online learning would help to create an adaptive algorithm.