Human Action Recognition Using Gaussian Mixture Model based Background Segmentation

Yiğithan Dedeoğlu
Department of Computer Engineering, Bilkent University, Ankara, Turkey

yigithan@cs.bilkent.edu.tr

Abstract Video surveillance has long been in use to monitor security sensitive areas such as banks, department stores, highways, crowded public places and borders. The increase in the number of cameras in ordinary surveillance systems overloaded both the human operators and the storage devices with high volumes of data and made it infeasible to ensure proper monitoring of sensitive areas for long times. In order to filter out redundant information generated by an array of cameras, and increase the response time to forensic events, assisting the human operators with identification of important events in video by the use of "smart" video surveillance systems has become a critical requirement. In this paper we present an instance based machine learning algorithm and system for real-time human action recognition which can help to build intelligent surveillance systems[1]. The proposed method makes use of visual information to classify actions of humans present in a scene monitored by a stationary camera. The object detection subsystem utilizes adaptive Gaussian Mixture based model for background segmentation. Template matching based supervised learning method is adopted to classify actions using object silhouettes into predefined classes.

1. Introduction

Action recognition can be described as the analysis and recognition of human motion patterns. Understanding activities of objects, especially humans, moving in a scene by the use of video is both a challenging scientific problem and a very fertile domain with many promising applications. Thus, it draws attentions of several researchers, institutions and commercial companies [6].

Our motivation in studying this problem was to create a human action recognition system that can be integrated into an ordinary visual surveillance system with real-time moving object detection, classification, tracking and activity analysis capabilities. The system is therefore supposed to

[1] This is an extension to our work presented with title "Multi-modal Action Recognition Using Audio-Visual Information". The extensions are the introduction of new actions: falling and laying; the use of Gaussian Mixture Model for background learning and the use of k-Nearest Neighbors algorithm for both pose template matching and feature reduction.

work in real time. Considering the complexity of temporal video data, efficient methods should be adopted to create a fast, reliable and robust system.

The action recognition system exploits objects' silhouettes obtained from video sequences by use of adaptive Gaussian Mixture models to recognize actions [5]. It mainly consists of two major steps: manual creation of silhouette and action templates offline and automatic recognition of actions in real-time. In classifying actions of humans into predetermined classes like walking, running, boxing, hand waiving, kicking, falling and laying; we make use of temporal signatures of different actions in terms of poses. We identify various poses of human body by using a distance metric defined on the silhouette of detected moving objects in video and labeled pose templates.

The remainder of this paper is organized as follows. Section 2 states the related work. In the next two sections we give the details of moving object segmentation and visual action recognition system. Experimental results are discussed in Section 5 and finally we conclude the paper with Section 6.

2. Related Work

There have been a number of studies related to recognition of actions using video. The systems proposed so far can be divided into three groups according to the methods they use: general signal processing techniques to match action signals, template matching and state-space approaches.

The first group treats the action recognition problem as a classification problem of the temporal activity signals of the objects according to pre-labeled reference signals representing typical human actions [6]. For instance Kanade et al. makes use of the signals generated by the change of the angle between the torso and the vertical line that passes through a human's body to distinguish walking and running patterns [2]. In another work Schuldt et al. make use of a local SVM approach to define local properties of complex motion patterns and classify the patterns using well known popular classifier Support Vector Machine [4]. General methods such as Dynamic time warping, Hidden Markov models and Neural Networks are used to process the action signals.

Second group of approaches converts image sequences into static shape patterns and in the recognition phase compares the patterns with pre-stored ones. For instance by using PCA, Chomat et al. created motion templates and a Bayes classifier was used to perform action recognition [1].

The last group considers each pose of the human body as a state and calculates a probability density function for each different action sequences. A sequence can be thought of as a tour between different states. Hence the probability density function can be calculated from different tours of the same action. The probability functions than can be used to recognize test sequences.

3. Learning Scene Background for Segmentation

One of the most important parts of the action recognition system is the segmentation of moving foreground objects from the stationary background. There are various methods in the literature for efficient segmentation of moving objects [1]. Adaptive Gaussian mixture based background learning method is relatively a more robust algorithm compared to its counterparts due to its high adaptation capability to cluttered rapidly changing background scenes. Considering the case that the most of the visual surveillance systems include monitoring of outside environments like

garages, gardens and open vast areas, Gaussian mixture model proposed by Stauffer et al. would be an ideal fit to be used for object segmentation. For this purpose, we extended our system to use adaptive Gaussian Mixture model instead of background subtraction model for moving object segmentation.

3.1. Online Gaussian Mixture Model based Learning

The model presented by Stauffer and Grimson [5] is a novel adaptive online background mixture model that can robustly deal with lighting changes, repetitive motions, clutter, introducing or removing objects from the scene and slowly moving objects. Their motivation was that a unimodal background model could not handle image acquisition noise, light change and multiple surfaces for a particular pixel at the same time. Thus, they used a mixture of Gaussian distributions to represent each pixel in the model. Due these promising features, we implemented and integrated this model in our visual surveillance system.

In this model, the values of an individual pixel (e. g. scalars for gray values or vectors for color images) over time is considered as a "pixel process" and the recent history of each pixel, $\{X_1, \ldots, X_t\}$, is modeled by a mixture of K Gaussian distributions. The probability of observing current pixel value then becomes:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where $w_{i,t}$ is an estimate of the weight (what portion of the data is accounted for this Gaussian) of the i^{th} Gaussian $(G_{i,t})$ in the mixture at time t, $\mu_{i,t}$ is the mean value of $G_{i,t}$ and $\Sigma_{i,t}$ is the covariance matrix of $G_{i,t}$ and η is a Gaussian probability density function:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1}(X_t-\mu_t)}$$

Decision on K depends on the available memory and computational power. Also, the covariance matrix is assumed to be of the following form for computational efficiency:

$$\Sigma_{k,t} = \alpha_k^2 \mathbf{I}$$

which assumes that red, green, blue color components are independent and have the same variance. The procedure for detecting foreground pixels is as follows. At the beginning of the system, the K Gaussian distributions for a pixel are initialized with predefined mean, high variance and low prior weight. When a new pixel is observed in the image sequence, to determine its type, its RGB vector is checked against the K Gaussians, until a match is found. A match is defined as a pixel value within γ (=2.5) standard deviation of a distribution. Next, the prior weights of the K distributions at time t, $w_{k,t}$, are updated as follows:

$$w_{k,t} = (1 - \alpha) w_{k,t-1} + \alpha(M_{k,t})$$

where α is the learning rate and $M_{k,t}$ is 1 for the matching Gaussian distribution and 0 for the remaining distributions. After this step the prior weights of the distributions are normalized and the parameters of the matching Gaussian are updated with the new observation as follows:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho(X_t)$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t-\mu_t)^T(X_t-\mu_t)$$

where

$$\rho = \alpha\eta(X_t|\mu_k,\sigma_k)$$

If no match is found for the new observed pixel, the Gaussian distribution with the least probability is replace with a new distribution with the current pixel value as its mean value, an initially high variance and low prior weight. In order to detect the type (foreground or background) of the new pixel, the K Gaussian distributions are sorted by the value of w/σ. This ordered list of distributions reflect the most probable backgrounds from top to bottom since by equation of $w_{k,t}$ background pixel processes make the corresponding Gaussian distribution have larger prior weight and less variance. Then the first B distributions are chosen as the background model, where

$$B = argmin_b\left(\sum_{k=1}^{b} w_k > T\right)$$

and T is the minimum portion of the pixel data that should be accounted for by the background. If a small value is chosen for T, the background is generally unimodal. Figure 1 shows sample pixel processes and the Gaussian distributions as spheres covering these processes. The accumulated pixels define the background Gaussian distribution whereas scattered pixels are classified as foreground.

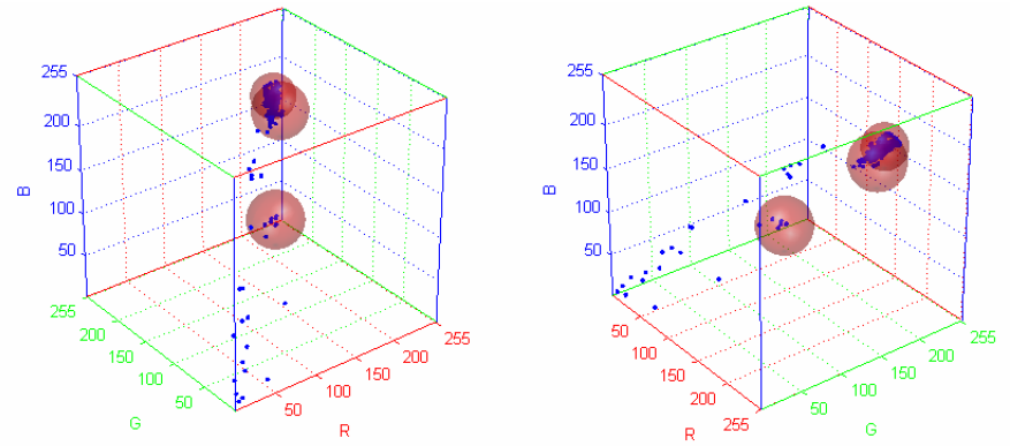

Figure 1 Two different views of a sample pixel processes (in blue) and corresponding Gaussian Distributions shown as alpha blended red spheres.

4. Vision-based Action Recognition

The proposed vision-based system operates on both color and gray scale video imagery from a stationary camera. It can handle object detection in indoor and outdoor environments and under changing illumination conditions. Since the moving objects in a scene can be of any type (humans, vehicles, animals or clutter) it would be beneficial to distinguish humans from other objects in order to increase the success rate in upper steps. The classification algorithm makes use of the shape of the detected objects and temporal tracking results to successfully categorize objects into pre-defined classes like human, human group and vehicle. The proposed algorithm also successfully tracks video objects even in short-lived full occlusion cases. In addition to these, some important needs of a robust smart visual motion detection system such as removing shadows, detecting sudden illumination changes and distinguishing left/removed objects are met [3].

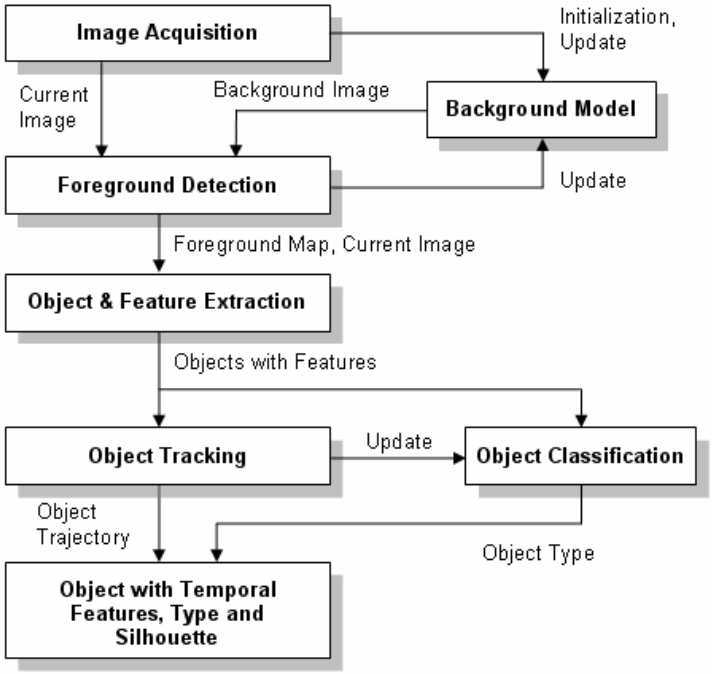

Figure 2 Overview of object extraction

Our visual action recognition system consists of two steps: an offline training step and an online recognition step. The common procedure in these to steps is extraction of objects and their features from image sequences. The overview of this common step is shown in Figure 2. The reader is referred to [3] for the details of object extraction process.

The offline step of the recognition algorithm is summarized in Figure 3 and consists of Object extraction, silhouette-based template pose database creation and action template database creation.

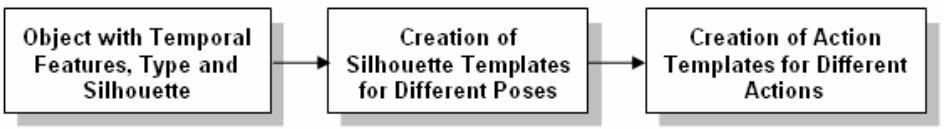

Figure 3 Offline action recognition steps

4.1. Silhouette-based pose template database creation

A typical human action such as walking, running, kicking, etc. involves repetitive motion. Although throughout a video sequence several hundreds of silhouettes can be extracted for a subject, the shapes of the silhouettes will exhibit a periodic similarity. Hence, the basic set of shapes for a full period can represent an action. Furthermore, the key frames in the basic motion set show differences from action to action. For instance a walking sequence from side view can be represented with three key frames: one for the case the legs are full open, one for the case the legs are fully closed and one for in the case in between. Considering a hand waving action, again this can be represented with two or three key frames: one for the case arms are fully open on sides, one for the case hands are vertical and one for the case in between if necessary. Some of the possible poses that can be seen during walking action are depicted in Figure 4 with an ID number beneath.

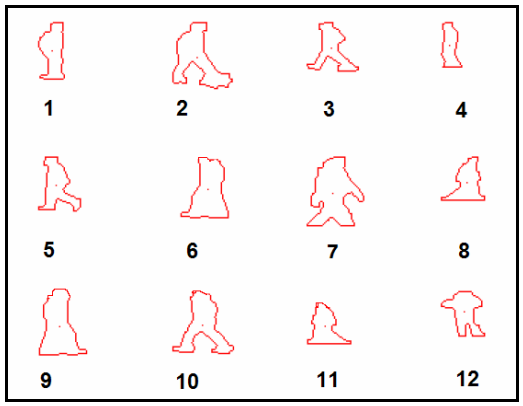

Figure 4 Silhouette-based pose template database

The template pose database is manually created with extracted object silhouettes as shown in Figure 4. The key point in creating the pose database is to include as much key frames as possible for a specific action and at the same time to pay attention to make the distance inter key frames of different actions as much as possible. For this purpose while creating the pose database before adding a pose to the database we search for whether there are similar poses in the database or not. A threshold T for the distance between two poses is used to decide whether they are similar of not. In other words if $dist(p1, p2) < T$ (where p1 and p2 are different pose silhouettes). We used the k-Nearest Neighbor approach here and k-neighbors of a pose are searched in the database, which are similar according to the above criteria. Then instead of adding the new silhouette distance signal to the database the average silhouette distance signal of the k-neighbors and the new one is added and the k-neighbors are deleted from the database. The distance and silhouette distance signal will be defined shortly.

4.2. Creating action template database

After creating the pose database the next step is to create action templates. Actions can be represented with a histogram of key frames it matches. In other words, if we create a histogram of the size of the total number of key frames in the silhouette template database, and match generated silhouettes at each frame of the training action sequence, to a key frame in the template pose database and increase the value of the corresponding bin (key frame's ID) in the histogram, we can create a histogram for the action. This histogram will be different for different actions. Hence, the normalized histogram can be stored in a template action database and later used for

action sequence matching, that is, action recognition. Figure 5 shows two sample histograms for walking and running actions.

While creating the action database, pose silhouette of the object is searched in the template silhouette database. During the search to silhouettes are matched according to a distance function defined on the silhouette distance signal. Here we adopted the k-Nearest Neighbor method to find the matching silhouette. Instead of increasing the value in a single bin, we find the k-nearest neighbors of the silhouette signal in the database and update their corresponding bin value. This results in better and smooth histograms and eliminates the errors that may have causes by marginal changes in the silhouette contour.

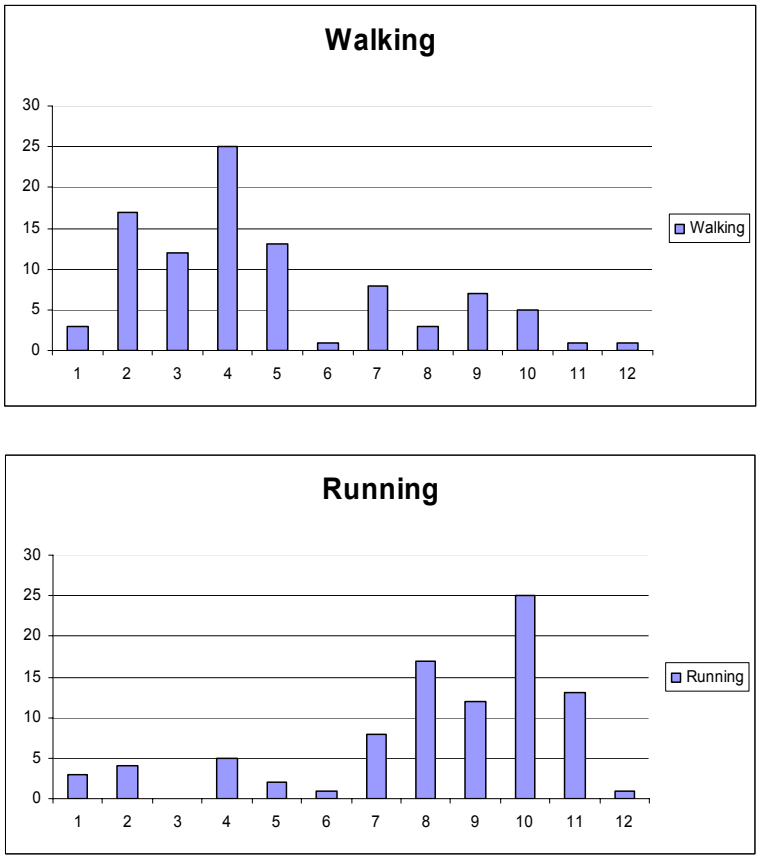

Figure 5 Sample action histograms for walking and running

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 cm 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 cm and each p_i starting from 1 through n as follows:

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

where the *Dist* function is the Euclidian 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 *D^S* is then calculated by sub-sampling or super-sampling the original signal *DS* as follows:

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

Then the distance signal is normalized to have unit area. Considering two distance signals DS_A and DS_B the distance between these signals $Dist_{AB}$ is calculated by using the L_1 distance as follows:

$$Dist_{AB} = \sum_{i=1}^{n} \left| DS_A[i] - DS_B[i] \right|$$

4.3. Recognizing actions in real-time

After creating action template database with histograms for distinct actions, test actions can be recognized at real-time.

In order to recognize an action, we keep a circular history of the IDs of the matching silhouettes in the template pose database for the subject's silhouette. At each frame we also create a normalized histogram of the IDs present in the history buffer. In the next step, we find the distance between the test subject's action histogram and each action histogram in the action template database by simple Manhattan distance. The type of the action in the database with minimum distance is assigned to the test action so the test action is recognized.

5. Experimental Results

We performed our experiments with two subjects. One subject is used to create the template pose and template action databases. The other subject is used for recognition tests.

We created action templates for the following actions: walking (W), running (R), boxing (B), kicking (K), hand waiving (HW), falling (F) and laying (L). The confusion matrix generated as the results of our tests is shown below (X stands for unstable recognition):

	W	R	B	K	HW	F	L	X	Success
W	6								100%
R	1	3							75%
B			4						100%
K				2		1		1	50%
HW					2				100%
F						3			100%
L				1			4		80%

6. Discussion

In this paper, we proposed a novel system for reliable action recognition using Gaussian Mixture model for moving object segmentation. The test results show that the presented method is promising and can be improved with some further work, such as applying maximum likely hood to classification results.

We may need to perform more tests, since the test data is a small set. We can find and try action recognition database from the Internet.

A weakness of visual recognition is that it is view dependent. We may work on automating the template database creation parts as well such as by applying automatic feature reduction on template silhouettes. This will possibly lead to a self calibrating action recognition system.

References

[1] O. Chomat, J.L. Crowley, Recognizing motion using local appearance, International Symposium on Intelligent Robotic Systems, University of Edinburgh, 1998.

[2] R. T. Collins et al. A system for video surveillance and monitoring: VSAM final report. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.

[3] Y. Dedeoglu, Moving object detection, tracking and classification for smart video surveillance, Master's Thesis, Bilkent University, Ankara, 2004.

[4] C. Schuldt, I. Laptev and B. Caputo, Recognizing Human Actions: A Local SVM Approach, in proc. of ICPR'04, Cambridge, UK.

[5] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, page 246252, 1999.

[6] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. Pattern Recognition, 36(3):585-601, March 2003.