Human activity recognition: Interaction between persons

João Pedro Fernandes

Instituto Superior Técnico
1049-001 Lisboa,
Portugal

***Abstract*—This article presents a system that is capable of automatically identify interactions between two persons using a video camera, and compares two models of the human body (silhouette model and anatomic model) when used with that purpose.**

The implemented system segments and tracks active regions in a color image sequence in order to estimate the silhouette model. This model is then refined to obtain the anatomic model. This is made at the expense of addicional processing steps, where Gaussian mixture models are used to classify individual pixel, based on their color. Next, the pixels are merged into blobs that share spacial and color features. Finally it is used a coarse human body model to classify the blobs into body parts, thus obtaining the anatomic model.

After estimating both of the human body models, the system extracts descriptive features from each one, that are used in conjunption with a k-nearest neighbors classifier to proceed to the interactions identification.

The system was tested to identify four types of interactions: hug, shake-hands, cross and fight. The experiments show that the implemented system can effectively identify interactions, presenting a high recognition rate for both of the human body models tested.

***Keywords*—Video Vigilance Systems, Human Interactions, Silhouette Model, Anatomic Model.**

I. Introduction

THE demand for intelligent video surveillance systems has grown over the last few years. Altough currently there are systems that are capable or performing some automatic tasks, like people tracking and face recognition, in nowadays the challenge is no longer knowing who people are, and where they are, but knowing what they´re doing.

Most computer vision studies on human action have concentraded on a single person in isolation, [1], [2] or in the detection of anomalous activities [3], [4]. Recently many authors began to study interactions between people [5], [6], [7]. Such systems typically consist of a low or mid-level computer vision system to detect and segment a moving object and a higher level interpretation module that classifies the motion into atomic behaviors. One author that has been particulary active in the interaction recognition problem is Aggarwal, the work presented in this paper follows the work done by Aggarwal et al. in [8], where they present a method for segmentation and tracking of multiple body parts in a bottom-up fashion. The contribution presented in this paper is then a classifier that uses features extracted from a human body model similar to the one presented by Aggarwal to classify interactions between two people.

This paper presents a system that is capable of tracking and segmenting humans, in order to recognize four interaction patterns: hug, shaking hands, cross and fight. Given that the interaction recognition is a complex problem, some assumptions are made in order to simplify it. So, it is assumed that the background is static, that the illumination is stable and that there are at most two people in scene. The persons that are performing the interactions, are actors that move perpendicularly to the camera view, interact near the camera but with all of their body inside the camera view and wear different color garments in the upper and lower body. Finally, the camera position is static, records color images and has perspective projection.

In the proposed system, background subtraction is performed in every frame, to segment the persons. Then a tracking method is used to follow each person along the sequence and deal with occlusions between persons. Next a human anatomic model is built by clustering individual pixels into homogeneous blobs. After each blob person membership has been decided, each blob is further classified into body parts. After this process two types of features are extrated, features from the silhouettes of the persons and features from the anatomic model, this two types of features are then used separately with a K-nearest neighbor classifier to recognize the interactions performed. Fig. I shows the implemented system overall diagram. In the end, a comparison between the two types of features used for classification of the interactions is made, and conclusions about the use of each one in the context of interaction recognition are made.

The rest of the paper is organized as follows. Section 2 describes the segmentation of the persons, Section 3 describes the tracking method used, Section 4 presents a method for estimating an anatomic model for human body representation, Sections 5 and 6 present the features extracted and the classifier used in the interaction recognition. Experiments and conclusions follow in Sections 7 and 8.

II. Segmentation

Detection of foreground regions and their segmentation is an essencial task in the solution of the proposed problem. With that end, background subtraction is performed in each

frame. The color distribution of each pixel $\mathbf{v}(x, y)$ at image coordinate (x, y) is modeled as a Gaussian

$$\mathbf{v}(x, y) = [v_H(x, y), v_S(x, y), v_V(x, y)]^\top. \quad (1)$$

The mean $\mu_Z(x, y)$ and standart deviation $\sigma_Z(x, y)$ of pixel intensity at every location (x, y) of the background model is calculated for each color channel $Z \in \{H, S, V\}$ using 20 training frames that are captured when no person appears in the camera view. It were used 20 training frames, since from a statistical viewpoint, 20 is regarded as the minimum number of samples for reliabe computation of mean and covariance. Foreground segregation is performed for every pixel (x, y), by using a simple background model, as follows: at each image pixel (x, y) of a given input frame, the change in pixel intensity is evaluated by computing the Mahalanobis distance from the Gaussian background model $\sigma_Z(x, y)$ for each color channel Z,

$$\delta_Z(x, y) = \frac{|v_Z(x, y) - \mu_Z(x, y)|}{\sigma_Z(x, y)}. \quad (2)$$

The foreground image $F(x, y)$ is defined by the maximum of the three distance measures, δ_H, δ_S, and δ_V for the H, S and V channels.

$$F(x, y) = max[\delta_H(x, y), \delta_S(x, y), \delta_V(x, y)]. \quad (3)$$

F is then thresholded to make a binary mask image. The background subtraction method used is similar to the one in [9]. In general, low threshold values produce larger foreground regions and more background noise, while high threshold values produce smaller foreground regions with possible holes and less background noise. The threshold value was trained through experimental trials, where the foreground areas obtained with different thresholds were compared.

After the background subtraction, morphological operations are performed as a post-processing step to remove small regions of noise pixel. Fig. 2 shows an example of an input image and its foreground-segmented image.

III. Tracking

This module includes three main steps: (A) correspondence between foreground region and track, (B) merging and splitting detection, and (C) occlusion handling. The tracking method used is based on a method proposed in [10].

(a) (b)

Fig. 2. Examples of an input image frame (a) and its foreground image (b).

A. Active Regions Correspondence

As in most of the tracking approaches, the correspondence process attempts to associate the foreground regions with one of the existing tracks. Let $T^t = \{T_1^t, T_2^t, ..., T_m^t\}$ denote the existing tracks and $M(t) = \{M_1^t, M_2^t, ..., M_n^t\}$ denote the foreground region measures at time t. This process starts with the construction of a distance matrix $\Delta^{t,t-1}$ between the active track T^t and each of the foreground region measure M^t. The distance matrix $\Delta^{t,t-1}$ (rows correspond to existing tracks and columns to foreground regions in the current frame) is based on the Euclidean distance,

$$\Delta_{ij}^{t,t-1} = \sqrt{(T_i^t x - M_j^t x)^2 + (T_i^t y - M_j^t y)^2} \quad (4)$$

where $T_i^t x$, $T_i^t y$, $M_j^t x$ and $M_j^t y$ represent the centroid coordinates of T_i^t and M_j^t, $i = 1, ..., m$ $j = 1, ..., n$.

Considering the similarity between the tracker and measure, if their distance is larger than a threshold, they will not be associated and the relative element in matrix $\Delta^{t,t-1}$ will be set to infinitude. Based on analyzing the matrix $\Delta^{t,t-1}$, a correspondence matrix C^t at time t is constructed to assign the foreground region measure to the track. The following is the details of its construction.

1) Firstly, all the elements of matrix C^t are set to zero.
2) Find the position of the minimal elements in every row $\alpha = \{\alpha_1, ..., \alpha_m\}$ and column $\beta = \{\beta_1, ..., \beta_m\}$ of the matrix $\Delta^{t,t-1}$.

$$\Delta_{i\alpha_i}^{t,t-1} = min(\Delta_{ij}^{t,t-1}), \quad j = 1, ..., n \quad (5)$$

$$\Delta_{\beta_j j}^{t,t-1} = min(\Delta_{ij}^{t,t-1}), \quad i = 1, ..., m \quad (6)$$

3) Finally add one to the correspondent element in matrix C^t.

$$C_{i\alpha_i}^t = C_{i\alpha_i}^t + 1, \quad i = 1, ..., m \quad (7)$$

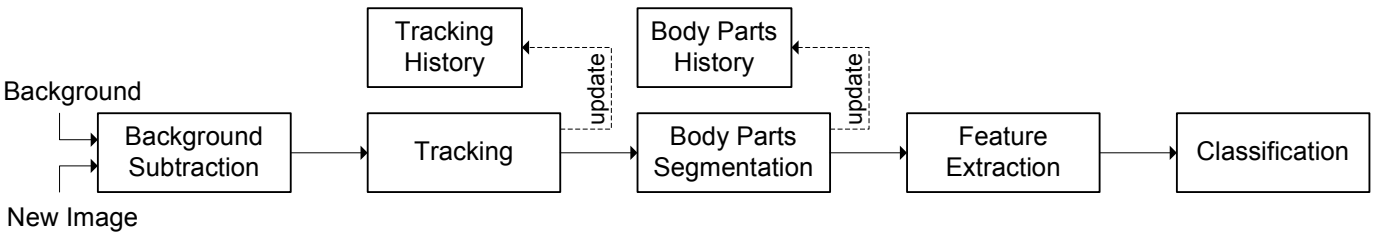


Fig. 1. System diagram.

$$C^t_{\beta_j,j} = C^t_{\beta_j,j} + 1, \qquad j = 1, ..., n \tag{8}$$

Three possible values may be found in the elements of matrix C^t: zero, one and two. Zero means no selection. One represents one selection happens. Two means the track and the measure selects each other both. Five possible results can arise in the matrix C^t:

1) A track is not associated to any measure (All the elements in a row are zero).
2) A measure is not associated to any track (All the elements in a column are zero).
3) A track is associated to more than one measure (More than one element in a row are larger than zero).
4) A measure is associated to more than one track (More than one element in a row are larger than zero).
5) A measure is associated to a track (The element value is two).

If an element value in matrix C^t equals to two, the measure will assign to the track, and all the elements in the same row and column of the distance matrix $\Delta^{t,t-1}$ are updated to infinitude. After that, a new correspond matrix $C^t$is constructed from the updated distance matrix $\Delta^{t,t-1}$. This process will keep on looping until none of the elements value of matrix C^t equals to two. Finally, the foreground measures and existing tracks are classified into three parts: Non-matched track, non-matched measure, matched track and measure.

The above association method assigns one measure to one track and can not handle merging and splitting event, in which one measure may assign to multiple tracks and one track may assign to multiple measures. To solve this problem, it was developed a merging and splitting detection procedure based on the obtained classification results.

B. Merging and Splitting Decision

For those non-matched tracks, a merging detection algorithm is used to decide whether the track is merged by another measure or is missed. If a merging happens, a new group is generated. If the track is missed, the confidence of the track will be decreased, once it drops below a specific threshold, the track will be deleted. For those non-matched measures, a splitting detection module is developed to decide whether the measure is split from an active track or it is a new target. When a splitting event is confirmed, occlusion handling (see Section III C) is performed to label each object correctly. Merging might occur due to a non-matched track overlapped with a measure. This judgment is based on the assumption that there must be overlapped area between the initial merging bounding box and the merged object (Fig. 3, first row). This is a valid assumption when the segmentation process is fast enough. As soon as object touches with each other at time $t+1$, a large bounding box containing all the merged objects will be created and it has large overlapping areas with the merged objects at time t. This assumption was tested with the segmentation being performed at 15*fps* and remained valid. Similar to the merging method above, splitting is detected due to a non-matched measure overlapped with a track (Fig 3, second row). When a group splits, each split object will be labeled correctly with a occlusion handling method.

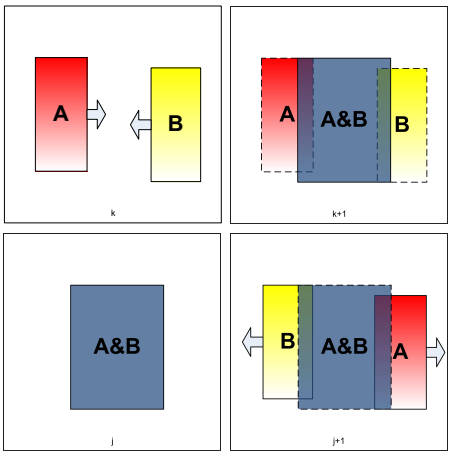


Fig. 3. A scenario of merging and splitting detection. The first row contains the blob merging events and the overlapping areas. The second row contains the splitting events.

C. Occlusion Handling

When a merging event has been detected, the information of the occluded track will be added into the group. After that, the entire group will be tracked as one target. When it splits, its necessary to label each object correctly. To accomplish this task, given that most of the times people wear different color clothes in the torso and legs, whenever a foreground region j containing a single person is detected at time t, a color feature vector c^t_j is extracted,

$$c^t_j = [\mu_{H_{sup}}, \mu_{S_{sup}}, \mu_{V_{sup}}, \mu_{H_{inf}}, \mu_{S_{inf}}, \mu_{V_{inf}}] \tag{9}$$

where $\mu_{H_{sup}}$, $\mu_{S_{sup}}$, $\mu_{V_{sup}}$, $\mu_{H_{inf}}$, $\mu_{S_{inf}}$ and $\mu_{V_{inf}}$ correspond to the mean intensities of the color channels H, S and V for all pixels of the top and bottom half, respectively, of the bounding box contaning the foregound region j. This color feature vector is then used to update a color feature vector C_i belonging to each person i detected,

$$C_i = [\mu_{H_{Tsup}}, \mu_{S_{Tsup}}, \mu_{V_{Tsup}}, \mu_{H_{Tinf}}, \mu_{S_{Tinf}}, \mu_{V_{Tinf}}] \tag{10}$$

where $\mu_{H_{Tsup}}$, $\mu_{S_{Tsup}}$, $\mu_{V_{Tsup}}$, $\mu_{H_{Tinf}}$, $\mu_{S_{Tinf}}$ and $\mu_{V_{Tinf}}$ correspond to the mean intensities of the color channels H, S and V for all pixels of the top and bottom half of the bounding box contaning the foregound region associated with the person i, in every image that the person i was detected isolated.

When a split is detected at time t, a distance matrix $\Delta^{c,C}$ is constructed, between each person that was in the group track in $t-1$ and the foregound regions in t that resulted from the division. The distance matrix $\Delta^{c,C}$ is based on the Euclidean distance,

$$\Delta^{c,C}_{ij} = \sqrt{(C_i - c^t_j)(C_i - c^t_j)^\top}. \tag{11}$$

The association between each person i and each foreground region j can then be formulated has an optimization poroblem, and resolved with the Hungarian algorithm [11].

IV. Anatomic Model

In this paper its pretendend to represent each person by an hierarquic human body model. The body model used is

simillar to the one proposed in [8], and consists in the division of each individual in head, upper body and legs. Fig. 4 illustrates the human body representation used.

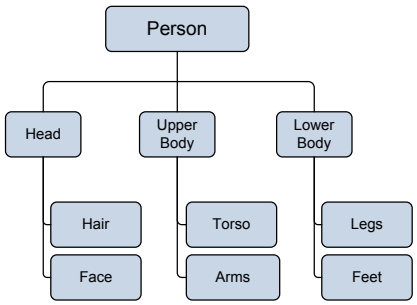


Fig. 4. Human body model.

In order to build the body model, each foreground region detected is segmented in blobs, by clustering individual pixels based on color intensities. Next after the person membership of each blob has been decided, each blob is further classified as belonging to the head, upper body, or lower body of the person.

The construction of the model is made in every frame of the sequence in analysis, in order to extract features from it that allow the classification of the interactions. At the same time it is mantained a model of each individual, that consists on the mean intensities of each color channel H, S and V for each anatomic part for every frame processed. This model is used to decide the person membership of some blobs.

A. Blob Formation

The goal of this module is to segment the foreground regions detected, in blobs made of pixels that share color and spacial features. This module has two main steps, initial blob formation and over segmented blobs merge.

1) Intial Blob Formation: In HSV space, the color values of a pixel at location (x, y) are represented by a random variable $v = [v_H, v_S, v_V]^\top$ with a vector dimension d = 3. According to the method in [12], the color distribution of a foreground pixel v is modeled as a mixture of C Gaussians weighted by prior probability $P(\omega_r)$, given by

$$p(\mathbf{v}) = \sum_{r=1}^{C} p(\mathbf{v}|\omega_r)P(\omega_r), \tag{12}$$

where the rth conditional probability is assumed as a Gaussian, as follows:

$$p(\mathbf{v}|\omega_r) = (2\pi)^{-d/2}|\Sigma_r|^{-1/2}e^{-\frac{(v-\mu_r)^T\Sigma_r^{-1}(v-\mu_r)}{2}} \quad with \;\; r = 1, ..., C. \tag{13}$$

Each Gaussian component θ_j represents the prior probability $P(\omega_r)$ of the rth color class ω_r, a mean vector μ_r of the pixel color component, and a covariance matrix Σ_r of the color components; $\theta_j = \{C,\ P(\omega_j),\ \mu_j,\ \Sigma_j\},\ j = 1, ..., C$. To obtain the Gaussian parameters, an EM algorithm [12] is used. This algorithm estimates the parameters recursively starting from initial estimates and then updating the parameters as follows,

$$\hat{P}(\omega_i) \leftarrow \frac{1}{n}\sum_{k=1}^{n}\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta}), \tag{14}$$

$$\hat{\mu}_i \leftarrow \frac{\sum_{k=1}^{n}\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta})\mathbf{v}_k}{\sum_{k=1}^{n}\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta})}, \tag{15}$$

$$\sum_{i}^{C} \leftarrow \frac{\sum_{k=1}^{n}\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta})(\mathbf{v}_k - \hat{\mu}_i)(\mathbf{v}_k - \hat{\mu}_i)^T}{\sum_{k=1}^{n}\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta})} \tag{16}$$

where,

$$\hat{P}(\omega_i|\mathbf{v}_k, \hat{\theta}) \leftarrow \frac{p(\mathbf{v}_k|\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{\sum_{j=1}^{c}p(\mathbf{v}_k|\omega_j, \hat{\theta}_j)\hat{P}(\omega_j)} \tag{17}$$

The method is initialized choosing a random mean among the values present in the foreground region in analysis for each color channel H, S, and V,

$$\mu_r = [v_H, v_S, v_V]^\top,$$

where

$$\begin{aligned} v_H &\in [min(v_H), max(v_H)] \\ v_S &\in [min(v_S), max(v_S)] \\ v_V &\in [min(v_V), max(v_V)] \end{aligned} \tag{18}$$

The covariance matrix is assumed to be an identity matrix.

$$\Sigma_r = \mathbf{I} \tag{19}$$

Finally all a priori probabilities are assumed as being equal,

$$P(\omega_r) = \frac{1}{C} \tag{20}$$

The training is performed as an iterative update of the parameters mentioned above following the equations (14-16). The iteration stops when the change in the value of the means is less than 1% compared to the previous iteration or when a user-specified maximum iteration number, ζ, is exceeded ($\zeta = 200$). The training depends on the initial guess of the Gaussian parameters. The method starts with 10 Gaussian components ($C = 10$) wich are then used to classify pixels into one of the C classes using a maximum a posteriori classifier (MAP),

$$\omega_L = arg\ max_r\ log(P(\omega_r|\mathbf{v})), \quad 1 \leq r \leq C. \tag{21}$$

The MAP probability $P(\omega_r|\mathbf{v})$ is calculated for every pixel $\mathbf{v}$ in the foreground region and every class r. A pixel v is classified as belonging to the class r that produces the biggest MAP probability.

The method above labels the foreground pixels with the same color as being in the same class, even though they are not connected. In an ideal situation, only the pixels connected to each other would be labeled as being in the same class. Therefore, its necessary to relabel the pixels with different classes if they are disconnected in an image. The connected component analysis is used to relabel the disjoint blobs, if any, with distinct labels, resulting in over-segmented small regions.

The number of disjoint blobs generated by the relabeling process may vary from frame to frame depending on the input image. The fluctuation of blob numbers causes difficulty. To maintain consistency, its necessary to merge the over-segmented regions into meaningful and coherent blobs. This requires a high-level image analysis that takes into account the relationship between the segmented regions.

2) Over Segmented Blobs Merge: Merging over-segmented blobs is a region growing procedure [13] controlled by features of the blobs. It were extracted two types of features, primary features that describe the blob and secondary features that describe the relation between adjacent blobs, that allow to describe the blob A_j as follows,

1) Primary features: concerning only one blob.
   - Blob label: $M(A_j) \in Z^+$.
   - Blob size: $\alpha(A_j) = |A_j|$, where $|A_j|$ is the number of elements in the blob.
   - Color: $[\mu_H, \mu_S, \mu_V]^T$ the mean intensities of H, S, V color components of the blob.
   - Blob position: $[\bar{I}, \bar{J}]^T$, the median position of the blob (i.e., the median values of horizontal and vertical projections of the blob in spatial coordinates).
   - Border pixel set: $\Psi(A_j)$, 8-connected outermost pixels corresponding to the contour of A_j.
2) Secondary features: determined by two adjacent blobs.
   - Adjacency lis: $\Gamma(A_j) = \{k \in Z^+ | A_k$ is adjacent to $A_j\}$ $k \neq j$.
   - Border ratio of A_j with respect to A_k: $\beta_j(A_k) =$ number of pixels in $\Psi(A_j)$ connected to A_k / $\Psi(A_j)$.
3) Its included the following skin label:
   - Skin label: $\varsigma(A_j)$

$$\varsigma(A_j) = \begin{cases} 1 & \text{se } |A_j \cap S| \geq |A_j| \times 0.75 \\ 0 & \text{caso contrário} \end{cases}$$

where S is the set of pixels of the foreground identified as skin.

Skin information is very useful in recognizing body parts. Skin color is determined by a single melanin pigment, and only its density differs between different ethnic groups. Its adopted a simple threshold model for skin color detection using the normalized RGB color space,

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B} \tag{22}$$

where a pixel v belonging to the foreground belongs to the set of skin pixels S if,

$$v \in S \text{ if } (\lambda_{g1}(r) \geq g \geq \lambda_{g2}(r)) \wedge (\lambda_{r1}(g) \geq r \geq \lambda_{r2}(g)) \tag{23}$$

where

$$\lambda_{g1}(r) = -0.3 \times r + 0.48 \tag{24}$$

$$\lambda_{g2}(r) = -042 \times r + 0.46 \tag{25}$$

$$\lambda_{r1}(g) = \frac{g - 0.237}{0.3} \tag{26}$$

$$\lambda_{r2}(g) = \frac{g + 0.13}{0.67} \tag{27}$$

The values of the thresholds λ_{g1}, λ_{g2}, λ_{r1}, and λ_{r2} were obtained by graphical analysis of skin pixels gathered from experimental data, and then tested to verify the robustness of the same.

Two blobs A_j and A_r are then merged if the following blob-merging criteria are satisfied,

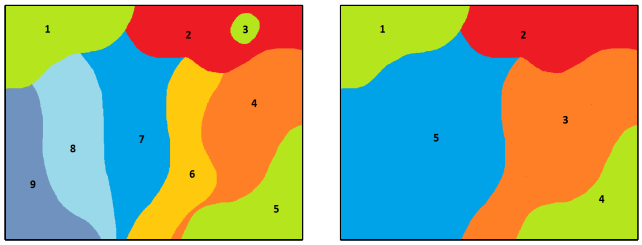
1) Adjacency criterion: two blobs should be adjacent.
2) Color similarity criterion: two blobs should be similar in color, where the similarity is defined by the Mahalanobis distance δ_ϕ of color feature ϕ between the blobs A_j and A_r, as follows:

$$\delta_\phi = (\phi_j - \phi_r)^T (\Sigma_\phi)^{-1} (\phi_j - \phi_r), \tag{28}$$

$$\phi = [\mu_H, \mu_S, \mu_V]^T, \tag{29}$$

   where Σ_ϕ is the covariance matrix of color values for all the blobs in the image. If δ_ϕ is less than a threshold λ_ϕ, blobs A_j and A_r are similar in color.
3) Border-ratio criterion: two blobs should share a large border. $(\beta_j(A_r) \geq Th_\beta) \vee (\beta_r(A_j) \geq Th_\beta)$ where Th_β is a treshold.
4) Small blob criterion: A small blob A_j less than a treshold Th_α that shares with another blob A_r more than 80% of its border $\beta_j(A_r) > 0.8$ does not need to follow criterion 2, unless its a skin labeled blob, $\varsigma(A_j) = 1$.

Fig. 5 illustrates the process of merging over segmented blobs. Blobs 4 and 6, and blobs 7, 8 and 9 follow the first three criterions and are merged. Blob 3 obeys the criterion 4 and is merged to blob 2.



(a) (b)

Fig. 5. Example of a over segmented blob merge process.

B. Overlap Management

When a merge is detected in the tracking module, this module is responsable for segmenting each of the individuals overlapped. The segmentation of each individual is achieved by assigning each of the blobs that belong to the foreground region containing the overlapped persons the identity of one of them. In order to assign each of the blobs, it was developed a system composed of three steps, blob tracking, spacial classification and model correspondence. The blobs go trough each of the steps until they are assigned an identity. If in the end of the process the identity of one individual has not been assigned to one of the blobs, this blob is considered unclassified and its not processed in the remaining of the global system.

1) Blob Tracking: There are some large blobs that usually correspond to large homogeneous areas of body parts (torso, arms or legs, etc.) that can be reliably tracked. So the first step in identifying the identity of the blobs consists in a blob tracking process. This process involves the following problems:

1) A different number of blobs may be involved at each time frame.
2) A single blob at time $t-1$ may split into multiple blobs at time t.
3) Multiple blobs at time $t-1$ may merge into a single blob at time t.
4) Some blobs at time $t-1$ may disappear at time t.
5) New blobs may appear at time t.
6) Its necessary to mantain the identity of the blobs throughout the sequence.

These phenomena complicate the blob tracking. Its needed to not only allow many-to-many mapping, but also avoid situations where scattered blobs in time $t-1$ are associated with a single blob at time t or situations, where a single blob at time $t-1$ is associated with scattered blobs in time t.

To accomplish this task, it was used a method proposed in [14]. It was chosen to use a different tracking method from the one already presented, because in the blob tracking problem, its necessary to give a bigger weight to the large blobs, and the method presented before is not suitable for dealing with blob merging and splitting.

Let´s denote the blobs already tracked up to frame $t-1$ as tracks R^{t-1}, and the new blobs formed at frame t as blobs B^t. Let the ith track at frame $t-1$ be track $R_i^{t-1} \in R^{t-1}$, and the jth blob at frame t be blob $B_j^t \in B^t$.

The task of blob-level tracking is to associate a blob B_j^t at frame t with one of the already tracked blobs R^{t-1} at frame $t-1$. Matching between the two sets is represented by edge matrix E.

$$E^{t-1,t} = \begin{pmatrix} \epsilon_{11}^{t-1,t} & \epsilon_{12}^{t-1,t} & \dots \\ \epsilon_{21}^{t-1,t} & \epsilon_{22}^{t-1,t} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{30}$$

where

$$\epsilon_{ij}^{t-1,t} = \begin{cases} 1 \text{ if the track } R_i^{t-1} \text{ its associated to the blob } B_j^t. \\ 0 \text{ otherwise.} \end{cases} \tag{31}$$

The blob association between R_i^{t-1} and B_j^t is performed by comparing the similarity between feature vectors m_i^{t-1} and m_j^t that describe each blob,

$$m_i^{t-1} = [\alpha, \mu_H, \mu_S, \mu_V, \bar{I}, \bar{J}] \text{ for } R_i^{t-1}, \tag{32}$$

$$m_j^t = [\alpha, \mu_H, \mu_S, \mu_V, \bar{I}, \bar{J}] \text{ for } B_j^t, \tag{33}$$

where α is blob size, μ_H,, μ_S,, μ_V, are mean intensities of the H, S, and V color components of the blob, and $\bar{I}$, $\bar{J}$ are the median position of the blob, respectively. The median position of the blobs are the median values of horizontal and vertical projections of the blob in spatial coordinates.

Given the covariance matrices Π_{t-1} and Π_t of these features for all the tracks in the image at time $t-1$ and all the blobs at time t, respectively, the Mahalanobis distance $\Delta_{ij}^{t-1,t}$ defines the dissimilarity between the ith track R_i^{t-1} at time $t-1$ and the jth blob B_j^t at time t as follows:

$$\Delta_{ij}^{t-1,t} = (m_i^{t-1} - m_j^t)^T (\Pi_{t-1} + \Pi_t)^{-1} (m_i^{t-1} - m_j^t). \tag{34}$$

The blob tracking process has two phases, a first phase where only one-to-one associations are allowed, and a second phase where one-to-many associations are performed.

The first phase is subject to one-to-one association constraints:

$$\sum_{j=1}^{|B^t|} \epsilon_{ij}^{t-1,t} = 1, \ \forall i = 1, ..., |R^{t-1}| \tag{35}$$

$$\sum_{i=1}^{|R^{t-1}|} \epsilon_{ij}^{t-1,t} = 1, \ \forall j = 1, ..., |B^t| \tag{36}$$

This initial phase will create four sub-sets ${}^1R^{t-1} \subseteq R^{t-1}$, ${}^0R^{t-1} \subseteq R^{t-1}$, ${}^1B^t \subseteq B^t$ and ${}^0B^t \subseteq B^t$, that correspond to the set of tracks and blobs with and without correspodence. Below its described the process used to perform the one-to-one association:

1) Compute dissimilarity $\Delta^{t-1,t}$:

$$\Delta^{t-1,t} = \begin{pmatrix} \Delta_{11}^{t-1,t} & \Delta_{12}^{t-1,t} & \dots \\ \Delta_{21}^{t-1,t} & \Delta_{22}^{t-1,t} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{37}$$

   If the value of $\Delta_{ij}^{t-1,t}$ is bigger than a given treshold $Th\Delta$, the value of $\Delta_{ij}^{t-1,t}$ its updated to infinity.
2) Backward search with auxiliary $\epsilon_{Bij}^{t-1,t}$:
   Search for each blob $j \in B^t$, starting form the largest blob, the track $i \in \{1, ..., |R^{t-1}|\}$ that has minimum dissimilarity $\Delta_{ij}^{t-1,t}$, under the one-to-one constraint in Eq. 35.

$$\epsilon_{Bij}^{t-1,t} = \begin{cases} 1 \text{ if a track } i \text{ is found.} \\ 0 \text{ otherwise.} \end{cases}$$

3) Forward search with auxiliary $\epsilon_{Rij}^{t-1,t}$:
   For each track $i \in R^{t-1}$, starting from the largest track, $j \in \{1, ..., |B^t|\}$ that has minimum dissimilarity $\Delta_{ij}^{t-1,t}$, under the one-to-one constraint in Eq. 36.

$$\epsilon_{Rij}^{t-1,t} = \begin{cases} 1 \text{ if a blob } j \text{ is found.} \\ 0 \text{ otherwise.} \end{cases}$$

4) Combine the backward and forward searches:

$$\epsilon_{ij}^{t-1,t} = \begin{cases} 1 \text{ if } (\epsilon_{Bij}^{t-1,t} = 1) \wedge (\epsilon_{Rij}^{t-1,t} = 1). \\ 0 \text{ otherwise.} \end{cases}$$

The first phase is a strict one-to-one mapping between tracks at frame $t-1$ and blobs at frame t, leaving some tracks ${}^0R^{t-1}$ and some blobs ${}^0B^t$ without correspondence. For example, let´s assume that two blobs have their minimum dissimilarity

with a particular track, but their minimum dissimilarity measures are slightly different. Only the blob that has the smaller dissimilarity value gets associated with that track in the first phase, because one-to-many mapping is not allowed in the first phase. The second blob is retained for the second phase. The following additional associations are performed in the second phase:

1) Association between ${}^1R^{t-1}$ and ${}^0B^t$.
2) Association between ${}^1B^{t-1}$ and ${}^0R^t$.

Associations between ${}^0R^{t-1}$ e ${}^0B^t$ are not performed, because if there had been any association, it would have been established in the previous one-to-one association, as an association between ${}^1R^{t-1}$ and ${}^1B^t$.

The associations in this phase is performed by searching for each track $R_i^{t-1} \in {}^0R^{t-1}$ the blob $B_j^t \in {}^1B^t$ that has minimum dissimilarity $\Delta_{ij}^{t-1,t}$, and search for each blob $B_j^t \in {}^0B^t$ the track $R_i^{t-1} \in {}^1R^{t-1}$ that has minimum dissimilarity $\Delta_{ij}^{t-1,t}$, and make the respective association. Not every element of ${}^1R^{t-1}$ and ${}^1B^t$ can be associated to ${}^0B^t$ and ${}^0R^{t-1}$. A track $R_{i=a}^{t-1} \in {}^0R^{t-1}$ can only be associated to a blob $B_{j=b}^t \in {}^1B^t$ if the track $R_{i=c}^{t-1} \in {}^1R^{t-1}$ to which the blob is already associated, $\epsilon_{i=c,j=b}^{t-1,t} = 1$, is contained in his adjacency list $R_{i=c}^{t-1} \subseteq \Gamma(R_{i=a}^{t-1})$. In the end of this process there are four possible outcomes for each blob $B_j^t \in B^t$,

- A blob is associated to one track.
- A blob is associated to many tracks.
- Many blobs are associated with one track
- A blob is not associated.

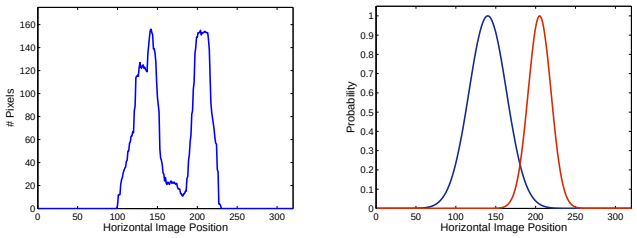
For each one of this situations its necessary to evaluate if its possible to assign the identity of one of the persons to each blob. If its not possible the blob is processed in the following steps of the overlap management module. When a blob is associated with a single track, its only possible to assign it an identity if the identity of the track to whom its associated is known. When a blob is associated to many tracks its only possible to assign it an identity if the identity of the several tracks is known, and they all belong to the same person. When a track is associated with many blobs, only if the identity of the track is known all the blobs can inherit his identity. When a blob is not associated its impossible to assign it an identity. It can be considered that the blob tracking process splits the set of blobs B^t in three sub-sets I^k, with $k = 0, 1, 2$. Where I^0 its the set of blobs with no identity assigned, and I^1, I^2 are the sets of blobs assigned to each one of the persons overlapped.

2) Spacial Classification: The next step in the overlapp management module is a spacial classification. This classification is made based in the assumption, that in most situations where two persons are overlapped, they keep some distance between them, and have only small portions of their body in contact.

This process starts by modeling the vertical projection profile of the foreground pixel set with a 1D mixture of two Gaussians, trained with the EM algorithm. Fig. 6 illustrates a vertical projection of a foreground pixel set and the Gaussians estimated. Next, its necessary to assign to each of the Gaussians the identity of one of the persons overlapped. To each of the Gaussians $\mathcal{N}_i$, its calculated the probability of belonging to the person k given by,

$$P(\mathcal{N}(\mu_i, \sigma_i) \Rightarrow k) = \frac{1}{N_{I^k}} \sum_{l=1}^{N_{I^k}} \frac{1}{N_{B_l}} \sum_{r=1}^{N_{B_l}} x_r \frac{1}{2\pi\sigma_i} e^{-\frac{(x_r-\mu_i)^2}{2\sigma_i^2}} \quad with\ i = 1, 2\ \ and\ \ k = 1, 2 \tag{38}$$

where N_{I^k} is the number of blobs that belong to I^k, N_{B_l} is the number of pixels in the blob $B_l \in I^k$ and x_r the horizontal position of r-th pixel that belongs to blob l. After calculating each of this probabilities, the association between each Gaussian i and each person k can be formulated has an optimization problem, and solved with the Hungarian algorithm [11].



(a) (b)

Fig. 6. Projection of a foreground region (a) and the Gaussian mixture trained (b)

Once the association between the Gaussians and the persons is made, the assignement of the individuals to each blob $B_j \in I^0$ can start. In order to do that, its calculated for each blob $B_j \in I^0$ the probability of belonging to the individual k is calculated,

$$P(B_j \Rightarrow k) = \frac{1}{N_{B_j}} \sum_{m=1}^{N_{B_j}} x_m \frac{1}{2\pi\sigma_k} e^{-\frac{(x_m-\mu_k)^2}{2\sigma_k^2}} \quad with\ \ k = 1, 2 \tag{39}$$

where N_{B_j} is the number of pixel in blob B_j and x_m the horizontal position of the m-th pixel of blob B_j. After the probability of the blob belonging to each individual is calculated, the blob is associated with the individual that has the biggest probability, as long as the probability is bigger than 0.7 and the absolute difference between probabilities is bigger than 0.3 . If this conditions are not met, the blob is processed in the last step of the overlap management module.

3) Model Correspondence: The last step in the overlap management module is a model correspondence, where a feature vector b_j describing each blob that has not been classified in the previous steps is compared with a feature vector v_i^k that describes each individual k body model,

$$b_j = [\mu_H(b),\ \mu_S(b),\ \mu_V(b)] \tag{40}$$

$$v_i^k = [\mu_H(v),\ \mu_S(v),\ \mu_V(v)] \tag{41}$$

where μ_H, μ_S e μ_V are the mean intensities of the color channels H, S and V.

Given the covariance matrices Π_b and Π_v of these features for all the blobs without correspondence and all body parts that make the body model of each individual respectively, the

Mahalanobis distance Δ_{ij} between the body part i and blob j is given by,

$$\Delta_{ij} = (b_j v_i^k)^T (\Pi_b + \Pi_v)^{-1} (b_j v_i^k). \quad (42)$$

Each value of Δ_{ij} is compared with a treshold, if its value is smaller the respective association is performed. In this process the feature vectors that correspond to skin body parts are excluded, because skin body parts don´t allow to descriminate between each of the individuals.

The blobs to whom is not assigned an identity is this step are considered not classified, and are not processed in the next modules of the overall system.

C. Body Parts Atribuition

Once the person membership of each blob is decided, each blob is further classified as belonging to the head, upper body, or lower body of the person according to the blob position´s proximity to each zone of the bounding-box regions. The skin predicate $\varsigma(A_j)$ of blob A_j determines whether the blob A_j belongs to a skin or non-skin part within the body-part membership. The bounding box areas are defined based on basic knowledge of human proportions, so:

- The head zone is defined as the vertical range between the bounding box top and κ_1 times the height of the person silhouette.
- The upper body zone is defined as the vertical range between κ_1 times the height and κ_2 times the height of the persons silhouette.
- The lower body zone is defined by the vertical range of the rest of the silhouette.

In Fig. 7 its illustrated a bounding box division. After estimating each of the areas, its calculated for every blob the percentage of its area that overlaps each zone. Each blob is then associated with the zone that overlaps the biggest share of his area. For example, a blob that has 80% of its area in the head zone and 20% in the torso zone is classified as head.

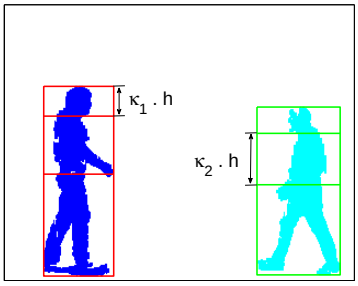


Fig. 7. Division of the bounding box containing the silhouette of each person.

The values of the parameters κ_1 and κ_2 are initialized with values 0.16 and 0.45 respectevely and are dynamically updated according to the change of the appearance of each person in the current frame. The updated values are used as the initial values to estimate the body parts in the next frame.

V. Feature Extraction

In this article the goal is to recognise four human interactions using only one second of information described by the feature vector x. The fact that humans can easily recognize an interaction pattern from a video sequence with one second, shows that sufficient information is contained in one second video sequences.

In each video sequence the beginning of the one second of information used to classify the interaction is given by the distance between the two persons. When the distance between the horizontal projection of the centroid from each of the persons foreground region is inferior to a given treshold the data gathering starts. The treshold used was trained with experimental trials. To make the system image size independent the distance between each person is normalized by the averge height. In Fig. 8 its illustrated the distance between the two persons in a video sequence and the one second interval used to classify the interaction.

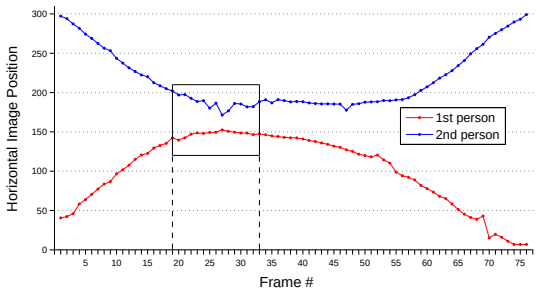


Fig. 8. Horizontal trajectory of two persons, and the one second interav used to classify the interaction.

The frames of a video can be seen as a sequence of matrices F_1, F_2, ..., F_K where K is the number of frames. So, each one second sequence, used to classify the interactions can be seen as,

$$F = \{F_1, F_2, ..., F_N\} \quad (43)$$

where N is the sample rate of the video in analysis. Each one second sequence is then described with a faeture vector $x \in \mathbb{R}^n$ $(n = 5)$ where x_i is the i-th feature. For each of the models being tested in this study, the features that make the feature vector x are different. The features extracted for each model are described in detail below.

A. Silhouette Model

The feature vector x components for the silhouette model are

- x_1: Active area variation.
- x_2: Movement uniformity.
- x_3: Spacial ocupation.
- x_4: Mean of the distance between persons.
- x_5: Standart deviation of the distance between persons.

In Fig. 9 its shown the mean of the feature vector x for each class of interaction. Each of the features and their extraction are described below.

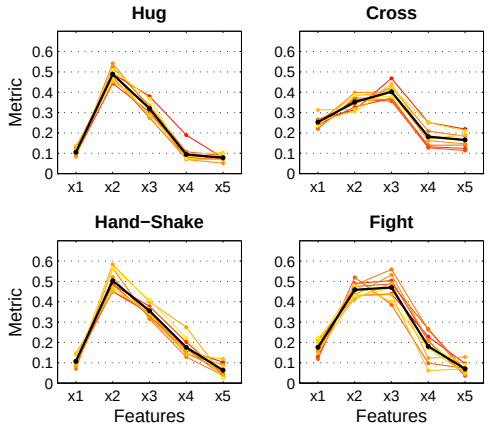


Fig. 9. Signature of each of the interactions classes when the silhouette model is used.

1) Active Area Variation: Active area variaton is the mean of the variation in the video sequence. The variation in the video sequence is given by the ratio between the pixels that are active in F_k and the pixel that weren´t active in F_{k-1}. If we name ${}^1F_k$ the set of pixels active in F_k the active area variation is given by,

$$x_1 = \frac{1}{N}\sum_{k=1}^{N}\frac{\#{}^1F_k - \#({}^1F_k \cap {}^1F_{k-1})}{\#{}^1F_k} \tag{44}$$

2) Movement Uniformity: The movent uniformity its extracted from a Motion Energy Image (MEI), similar to the one proposed in [15], where the foreground regions of each frame of the sequence are aggregated. The MEI representation can be interpreted as a matrix J given by,

$$J = \frac{1}{N}\sum_{k=1}^{N}{}^1F_k \tag{45}$$

The uniformity of movement is then given by the standart deviation of all the elements in J,

$$x_2 = \sqrt{\frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}(J_{ij}-\bar{J})^2} \tag{46}$$

where $\bar{J}$ is the mean of all elements in V given by,

$$\bar{J} = \frac{1}{m\times n}\sum_{i=1}^{m}\sum_{j=1}^{n}V_{ij} \tag{47}$$

3) Spacial Ocupation: In each frame $F_k \in F$ its extracted a bounding box that contains all silhouettes present in the scene. The spacial ocupation is then calculated as the mean, over all frames $F_k \in F$, of the bounding box width divided by its height.

4) Distance Between Persons: The distance between persons is calculated using a uni-dimensional Guassian misture trained in a similar way to the one presented in section IV-B2. Once the Gaussians are trained the distance between persons is given by the absolute difference between the means of both Gaussians. This distance is calculated in every $F_k \in F$ and then its mean and standart deviation over the sequence are used as features.

B. Anatomic Model

The feature vector x components for the anatomic model are

- x_1: Mean of the distance between heads.
- x_2: Standart deviation of the distance between heads.
- x_3: Mean of the distance between hands.
- x_4: Standart deviation of the distance between hands.
- x_5: Distance between arm angles.

In Fig. 10 its shown the mean of the feature vector x for each class of interaction. Each of the features and their extraction are described below.

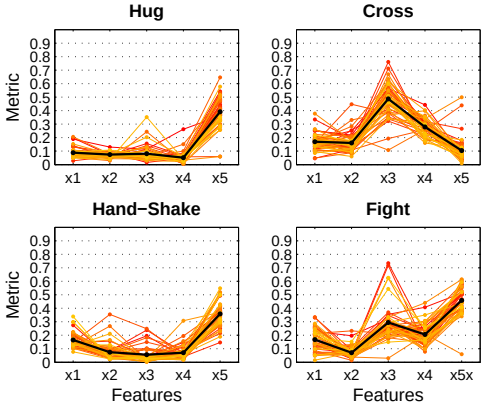


Fig. 10. Signature of each of the interactions classes when the anatomic model is used.

1) Distance Between Heads: The distance between heads its the absolute difference between both persons horizontal projection of the centroid of the heads, normalized by the average height. This distance is calculated in every $F_k \in F$ and then its mean and standart deviation over the sequence are used as features.

2) Distance Between Hands: For each person its estimated the hand of the outermost arm position. This position is given by the maximum curvature point of the convex hull that cointains the torso and is coicident with a skin marked blob. The distance between hands is then given by the norm between the hand position of both subjects. This distance is calculated in every $F_k \in F$ and then its mean and standart deviation over the sequence are used as features.

3) Difference Between Arm Angles: For each person its estimated the angle between the outermost arm and the torso. This angle is defined has the angle between a vertical line that crosses the shoulder and a line connecting the shoulder to the hand of the outermost arm. The position of the shoulder is given by the maximum curvature point of the convex hull that contains the torso that is closer to the head. After estimating the arm angle for both subjects, the difference between both angles is calculated and normalizaed by 360º. This value is only estimated in the first frame of the sequence F and its used as a component of feature vector x.

VI. Classification

A K-nearest neighbor method is used to classify a new interaction vector x, into one of the four interaction classes.

That is, x, is classified as an example belonging to class k if the class k produces overall minimum Euclidian distances from its stored model vectors to the input vector x, compared with the other classes. The choice off using this classifier was made since this classifier is easy to implement, and easy to train, not being need an extensive training data set for the classifier to have a good performance.

VII. Experimental Results

The system was coded in MatLab from Mathworks Inc. and run on a personal computer Vaio VGN-FW21M from Sony installed with the Microsoft Windows 7 operating system. The computer was equipped with a Intel Core Duo P8600 processor from Intel Corporation with 2.4 GHz clock speed. The video data used to test the system was captured with the webcam Motion Eye 1.3MP built-in the computer used. The computer was positioned in a stable base with the webcam viewing direction parallel to the ground. It were captured interactions between two persons in outdoor enviroment with a complex background. The video data was converted to image sequences of Windows AVI files, without any kind of compression, in 15 frames/s sequences of 320x240 pixel color images. The color quality was 24 bit (i.e., for each of the color channels R, G and B).

The system was tested with four interactions between two persons: hug, hand-shake, cross and fight. For each of the interactions it were used 10 pairs of persons, from a set of 7 volunteers, making a total of 40 sequences (4 interactions $\times$ 10 pairs of persons). All sequences start with no one in the image, with the persons starting from opposite positions and walking towards each other until they interact. The subjects wore various casual clothes, and were instructed to interact each other in a natural manner for the nine interaction types. An interaction sequence contains a single instance of an interaction between two persons, and varies in duration in the range of $2-9$ seconds depending on the persons and interaction type. The total number of frames captured for each interaction can be seen on Table I.

TABLE I
Number of frames captured for each interaction.

	hug	cross	hand-shake	fight	total
# frame	936	755	482	735	3108

For each of the models in study, the evaluation of the classifier was done using the leave one out method, meaning that the set of data used for training the classifier is constituted by all sequences available except the sequence used for testing. For the silhouette model it was used a K-nearest neighbor classifier that uses 3 nearest neighbors out of 39 to classify the interaction. For the anatomic model, since the feature extraction is dependent of the EM algorithm inicialization when building the anatomic model, the sequences were processed with five different inicializations and all of them were used in the test. So in this case we consider 200 sequences, five incializations of the 40 sequences in the data set. The classifier, when tested with the anatomic model, uses the 15 nearest neighbors out of 195 to classify the interaction. Table II presents the results for the interaction recognition when using the silhouette model, while Table III presents the results for the interaction recognition when using the anatomic model.

TABLE II
Results of the interaction recognition when the silhouette model is used.

	hug	cross	hand-shake	fight
hug	90	0	10	0
cross	0	100	0	0
hand-shake	0	0	100	0
fight	20	0	0	80

TABLE III
Results of the interaction recognition when the anatomic model is used.

	hug	cross	hand-shake	fight
hug	86	0	12	2
cross	0	96	0	4
hand-shake	12	2	84	2
fight	2	4	6	88

VIII. Conclusion

In this paper it was implemented and tested an automatic classifier for interactions between two persons, and a comparison was made between two human body models to be used with that end. It were made some assumptioms to simplify the problem, and in view of those assumptions the systen showed that it is effectively capable of identify interaction between two persons. Both of the human models used presented a high recognition rate, with the silhouette model presenting a slighter higher recogniton rate 92.5% than the anatomic model 90.8%. The anatomic model is more descriptive than the silhouette model, and presents a better potential to be used on future works where some assumptions about the pose of the persons interacting made in this paper are eliminated. The silhouette model, on the other hand proved to be more robust and presents a better potential to be used on far-view scenarios, where color features and body parts are difficult to extract.

References

[1] K. Jia and D.-Y. Yeung, "Human action recognition using local spatio-temporal discriminant embedding," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[2] A. Yilmaz and M. Shah, "Actions sketch: a novel action representation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 984 – 989 vol. 1, june 2005.

[3] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–819 – II–826 Vol.2, jun. 2004.

[4] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa, ""shape activity": a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection," *Image Processing, IEEE Transactions on*, vol. 14, pp. 1603 –1616, oct. 2005.

[5] S. Park and J. Aggarwal, "Semantic-level understanding of human actions and interactions using event hierarchy," in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on*, p. 12, june 2004.

[6] Y. Du, F. Chen, and W. Xu, "Human interaction representation and recognition through motion decomposition," *Signal Processing Letters, IEEE*, vol. 14, pp. 952 –955, dec. 2007.

[7] H.-I. Suk, B.-K. Sin, and S.-W. Lee, "Analyzing human interactions with a network of dynamic probabilistic models," in *Applications of Computer Vision (WACV), 2009 Workshop on*, pp. 1 –6, dec. 2009.

[8] S. Park and J. Aggarwal, "Segmentation and tracking of interacting human body parts under occlusion and shadowing," in *Motion and Video Computing, 2002. Proceedings. Workshop on*, pp. 105 – 111, dec. 2002.

[9] K. Sato and J. K. Aggarwal, "Temporal spatio-velocity transform and its application to tracking and interaction," *Comput. Vision Image Understand*, vol. 96, pp. 100 –128, 2004.

[10] T. Yang, Q. Pan, J. Li, and S. Li, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 970 – 975 vol. 1, jun. 2005.

[11] K. H., "The hungarian method for solving the assignment problem," *Naval Research Logistics Quart.*, vol. 2, pp. 83 – 97, 1955.

[12] P. Duda, R. andHart and E. Stork, *Pattern Classification second ed.*, ch. 10, pp. 517 –583. Wiley, New York, 2001.

[13] L. Salgado, N. Garcia, J. Menendez, and E. Rendon, "Efficient image segmentation for region-based motion estimation and compensation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, pp. 1029 –1039, oct. 2000.

[14] Y. Bar-Shalom and W. Blair, *Multitarget-multisensor tracking: applications and advances*, vol. 3, pp. 199 –231. Norwood, MA, 2000.

[15] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 257 –267, mar 2001.