A Joint Approach to Shape-based Human Tracking and Behavior analysis

Francesco Monti
Department of Biophysical and Electronic Engineering
University of Genova, Italy.
monti@dibe.unige.it

Carlo S. Regazzoni
Department of Biophysical and Electronic Engineering
University of Genova, Italy.
carlo@dibe.unige.it

Abstract – *In this paper a joint human tracking and recognition system is proposed. While usually these two functions are performed separately, it will be shown that it is possible to improve the estimation performances if these functions are done jointly. For this purpose, a Bayesian estimation framework is presented and implemented using sequential Monte Carlo techniques. Moreover it will be shown how the estimation can be performed efficiently by using the Generalized Hough Transform. The effectiveness of the proposed approach is demonstrated for a variety of image sequences.*

Keywords: Tracking, behavior analysis, estimation, video surveillance, information fusion.

1 Introduction

Tracking humans in video sequences and the analysis of their behavior are two functions that are commonly present in modern surveillance systems. Target tracking is the foundation for all the other analysis modules and usually feeds the other higher level modules like the behavior recognition module.

Among the tracking methodologies, different successful algorithms have been proposed in the past years. The algorithms in literature can be categorized considering the type of information they use to track the objects and to discriminate them from the clutter. Some approaches use measurements that are directly related to the color of the object [1, 2]. Other approaches use, as observations, measurements that are connected to the shape of the object. In [3] the observations are the edges of the contour of the object while in [4, 5] the measurements are keypoints, that can be seen as a sampling of the shape, extracted from an interest point detector. Recently, classifiers have been integrated into the tracker to avoid the drift of the tracker and to allow automatic initialization [6, 7].

Considering the behavior understanding methodologies, the approaches in literature can be divided in two main categories: the holistic approaches and the part-based approaches. Among the holistic approaches, in [8, 9] the optical flow of the stabilized object space-time volume is used as descriptor. In [10] the human movement is modeled as a sequence of silhouettes and a Markov model is used to characterize the different actions. Although the holistic methods are efficient, they have limitations with non stationary backgrounds and camera motion. Part-based approaches use only several "interesting" parts for action modeling and thus avoid problems such as local modifications or occlusions. The Space-Time interest points [11, 12] are commonly used part-based descriptors. Among the methods that use local information for action recognition, some use only the motion information of tracked interest points. These methods don't suffer moving background if the identities of the interest points are maintained and can be used in case of camera motion if the global motion of the object is compensated or not considered. Among these methods it is possible to cite [13, 14].

As shown, different approaches have been proposed in the past to separately track a person and to perform behavior recognition. Differently, the goal of this paper is to propose a system which jointly tracks and analyzes the behavior of the tracked targets. In the proposed framework the tracking sub-module uses, as observations, the position of some interest points generated by the tracked object. The analysis of the motion of these tracked keypoints can be also used to analyze the behavior of the object. In this approach the tracked keypoints can be seen as a sampling of the shape of the object and the behavior analysis can be seen as the analysis of the shape deformation during time. The information produced by the behavior recognition module is used to improve the tracking process and vice-versa in a Bayesian framework. While in the past some works [15, 16] have proposed to use a tracker with multiple observation/behavior models, no works related to the joint tracking and behavior analysis of humans in a surveillance system scenario, with all the difficulties in terms of the generalization of the behavior models required, have been published according to our knowledge. The main contributions of this work are:

- the introduction of a framework for the joint tracking and human analysis;
- an efficient implementation of the system, exploiting a

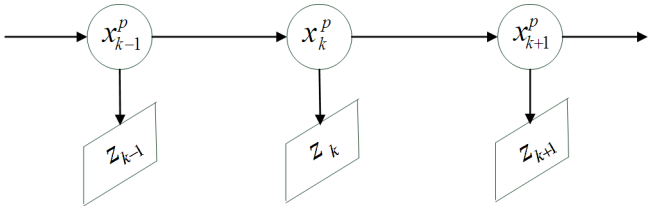

Figure 1: Graphical Model of the shape-based tracker

novel interpretation of the Generalized Hough Transform (GHT) [17].

The structure of the paper is as follows: at first the standalone shape-based tracking algorithm (paragraph 2) and the standalone behavior recognition algorithm (paragraph 3) are presented. These algorithm are the foundation of the proposed system. In paragraph 4 it will be shown how the two sub-modules can be fused together. In paragraph 5 an efficient implementation of the system will be proposed using the GHT. In paragraph 6 the experimental results of the system will be presented. Finally conclusions and future works will be discussed.

2 Target Tracking

From a Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state of the target x_k at time k, given the observations $z_{1:k}$ up to time k.

In the proposed shape-based tracking approach, the state of the target x^p_k is the position of the object on the image plane. x^p_k is therefore a vector of two components x, y. Due to the discrete nature of the images, it can take only values in the space $[N_{cols} \times N_{rows}]$ where N_{cols} is the number of columns of the image and N_{rows} is the number of rows.

The observations at time k are the x, y positions of the N interest points detected in the $k-th$ frame using an interest point detector [18]:

$$z_k = \{z^n_k\} \;\; n = 1 \dots N \tag{1}$$

Two sets of conditional independence relations are assumed: that z_k is independent of all other observations and states given x_k and that x_k is independent of $x_1, x_2 ... x_{k-2}$ given x_{k-1} (the first order Markov property). The dependence relations for this kind of approach are shown in Fig. 1. In this figure the standard notation of representing states variables with circles and "evidence" (observations) with squares has been used [19]. In this notation links represent conditional dependence relations.

To perform the tracking it is assumed that the initial probability density function (pdf) of the state vector $p\left(x^p_0\right)$, which is also known as the prior, is available. Then, in principle, the posteriori pdf $p\left(x^p_k|z_{1:k}\right)$ may be obtained, recursively, in two stages: prediction and update.

Supposing that the pdf $p\left(x^p_{k-1}|z_{1:k-1}\right)$ and a model for the target dynamics $p\left(x^p_k|x^p_{k-1}\right)$ is available at time $k-1$, the prediction stage is performed trough the Chapman–Kolmogorov equation:

$$p\left(x^p_k|z_{1:k-1}\right) = \int p\left(x^p_k|x^p_{k-1}\right) p\left(x^p_{k-1}|z_{1:k-1}\right) dx_{k-1} \tag{2}$$

At time step k, a measurement becomes available, and this may be used, via the likelihood function, to update the prediction:

$$p\left(x^p_k|z_{1:k}\right) = c \cdot p\left(z_k|x^p_k\right) p\left(x^p_k|z_{1:k-1}\right) \tag{3}$$

where c is a normalizing constant which depends only on the likelihood function.

To use the Bayesian framework, the dynamic model and the likelihood function have to be defined.

With respect to the dynamic model, the position of an object at time k, given the position at time $k-1$ is considered as uniformly distributed around x^p_{k-1} . This model implies that the target cannot have a velocity greater than a threshold $maxVel$. The dynamic model is defined as:

$$p\left(x^p_k|x^p_{k-1}\right) = \begin{cases} c & if \; \sqrt{[x^p_k]^t . x^p_{k-1}} < maxVel \\ 0 & otherwise \end{cases} \tag{4}$$

where c is a normalizing constant that makes $p\left(x^p_k|x^p_{k-1}\right)$ a pdf.

The goal of the likelihood function is to compute the probability that the observations have been generated by the tracked object, given an hypothesis on the state. It tries to exploit the temporal continuity of the shape of the tracked object and tries to explain the observations at time k using a shape model that was learned during the past time-steps.

The shape of the target which is compared with the observations is composed by a set of M discrete points. Each model point has coordinates Δ^m, expressed in a reference system centered on a reference point that corresponds to the center of the object.

The model point coordinates of the m-th model point can be projected to the image plane with respect to the object global position in the following way: $pos^m\left(x^p_k\right) = \Delta^m + x^p_k$.

The likelihood of the observations is therefore defined as:

$$p\left(z_k|x^p_k\right) = c \sum_{n=1}^{N} \sum_{m=1}^{M} K_{motion}\left(\|z^n_k - pos^m\left(x^p_k\right)\|\right) \tag{5}$$

where $K_{motion}\left(\right)$ is a function which is 0 if its argument is greater or equal than a threshold th and 1 otherwise and c is a normalization constant. The function $K_{motion}\left(\right)$ takes into account the possible distortions of the shape which occur from frame to frame and the observation noise. If only exact matches with respect to the model are allowed, th is set to 0 and only observations which are exactly in the predicted position are taken into account in the likelihood. It is worth noting that the proposed likelihood function is a Bayesian

Algorithm 1 Shape based Tracking

1: Resampling:
Resample the particle set $\left\{x_{k-1}^{p,(n)}\right\}$ based on $\pi_{k-1}^{(n)}$ to obtain $\left\{\tilde{x}_{k-1}^{p,(n)}\right\}$
2: Prediction:
For each $\left\{\tilde{x}_{k-1}^{p,(n)}\right\}$ sample the density of the target dynamics $p\left(x_k^p|\tilde{x}_{k-1}^{p,(n)}\right)$ to obtain $\left\{x_k^{p,(n)}\right\}$
3: Update:
Re-weight each particle by calculating the likelihood $\pi_k^{(n)} = p\left(z_k|x_k^{p,(n)}\right)$

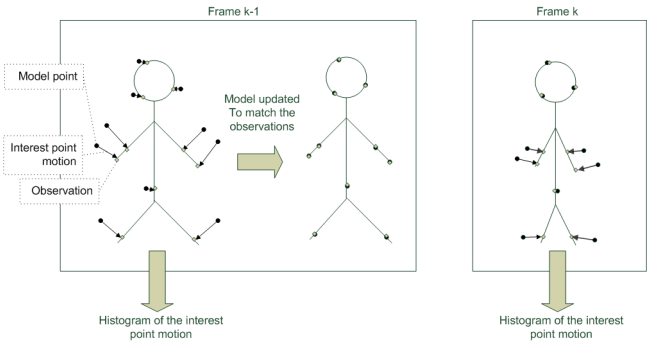

Figure 2: Shape deformation example

formalization of the Generalized Hough Transform (GHT) [17, 20].

This tracker can be easily implemented inside a particle filter framework [21]. The posterior density $p\left(x_{k-1}^p|z_{1:k-1}\right)$ is represented by a set of weighted particles $\left\{x_{k-1}^{p,(n)}, \pi_{k-1}^{(n)}\right\}$. The sampling-based algorithm is summarized in Algorithm 1.

The position at time k is estimated as:

$$\hat{x}_k^p = \sum x_k^{p,(n)} \pi_k^{(n)} \tag{6}$$

If a lot of particles have to be evaluated, the computation of (5) can be time consuming. An efficient way to evaluate (5) will be presented in paragraph 5.

The shape model is learned during the tracking process. At each time-step, after the position of the object $\hat{x}_k^p$ has been estimated as in (6), each observation point is put in correspondence with the model points projected on the image space using $\hat{x}_k^p$. Only associations which are nearer than th are considered as valid. Model points which have a match with an observation are updated so that their new position Δ^m is the position of the nearest observation. Model points that have not been in correspondence with observations since a number of frames are removed from the model. If there are observations that were not in correspondence with model points, new model points are added in the position of the observations. For details about the shape learning process see [4].

3 Behavior Recognition

A behavior recognition system is usually composed by a set of N_b models that are specialized to recognize a specific action. The action is classified by selecting the model which better describes the observations.

The system presented in this paragraph works with the assumption that a tracker like the one shown in paragraph 2 is able to follow the target. In the paragraph 4 it will be shown how the tracker can be fused with the system described in this paragraph to obtain a system that jointly tracks and recognize the actions.

In the proposed approach for recognition, each action is modeled as a sequence of shape deformations. After the target position has been estimated, it is possible to estimate the deformation that the shape has encompassed from frame $k-1$ to k. It is important to note that, considering the shape model and the observations, each estimated position induces a different shape deformation. The standalone behavior recognition module considers the position as fixed to the one estimated by the tracker (as shown in (6)) and computes the shape deformation accordingly. Each model point, given the position, can be put into correspondence with the observations and its motion from frame $k-1$ to k can be computed as shown in Fig. 2. The shape deformation is then modeled as an histogram of the deformations of the model points in the current frame. The motion information is discretized using a function $map : \mathbb{Z}^2 \rightarrow \mathbb{N}$ which maps the x, y components of the interest point motion vector v to a histogram bin. Different choices of mapping functions are possible. In this paper the following simple function is used $map(\mathbf{v}) = fix(\sqrt{\mathbf{v}^t\mathbf{v}})$ which simply truncates the magnitude of the feature motion vector to map it to a histogram bin. The motion histogram is composed of th bins.

The behavior descriptor is created considering all the matches between the model points $m = 1 \ldots M$ and the observations $n = 1 \ldots N$ as follows:

$$\begin{aligned} if \quad & K_{motion}\left(\|z_k^n - pos^m\left(\hat{x}_k^p\right)\|\right) = 1 \\ & map\left(z_k^n - pos^m\left(\hat{x}_k^p\right)\right) \rightarrow binID \\ & d_k\left(\hat{x}_k^p\right)[binID] = d_k\left(\hat{x}_k^p\right)[binID] + 1 \end{aligned} \tag{7}$$

where $d_k\left(\hat{x}_k^p\right)$ is the descriptor. For ease of visualization in case it is not necessary, the dependence of d_k from $\hat{x}_k^p$ will be removed. As shown in Fig. 2 the representation used for action classification is really sparse.

This descriptor describes the instantaneous shape deformation. The proposed descriptor shares some similarities with the one defined in [14] (considering that it uses a histogram of counts computed from the movement of sparse interest points) and the one defined in [8] (in which the optical flow of the tracked object is used as action descriptor).

During the training phase of the model, a number of histograms is collected using some sequences that are correctly tracked. By clustering the collected shape deformation histograms into N_o clusters it is possible to create a discrete alphabet of deformations D that contains the cluster centers.

It is therefore possible to model a continuous human action as a sequence of discrete deformation symbols.

The symbols are extracted by computing the distance between the deformation histogram and all the prototypes in D and by taking the one with minimum distance:

$$O_k\left(\hat{x}_k^p\right) = \underset{j=1\ldots N_o}{\operatorname{argmin}} \sqrt{\left[d_k\left(\hat{x}_k^p\right)\right]^t \cdot D^j} \tag{8}$$

where $O_k\left(\hat{x}_k^p\right)$ is the symbol and D^j is the $j-th$ prototype. For ease of visualization in case it is not necessary, the dependence from $\hat{x}_k^p$ will be removed.

These sequences of discrete symbols are modeled probabilistically trough an HMM [22]. The HMM is the most popular stochastic algorithm for discrete sequences modeling because of its versatility and mathematical simplicity. HMMs make it possible to deal with time-sequential data and can provide time-scale invariability in recognition. An HMM consists of a number of N_s states, each of which is assigned a probability of transition from one state to another state. Considering the discrete nature of the states, these probabilities can be described by using a $N_s \times N_s$ matrix A:

$$A_{ij} = p\left(x_k^b = i | x_{k-1}^b = j\right) \tag{9}$$

where x_k^b is the behavior state of the object at time k. With time, state transitions occur stochastically. Like Markov models, states at any time depend only on the state at the preceding time. One symbol is yielded from the HMM according to the probabilities assigned to the states and to the probability that the symbol is observed given the states.

The probability that one particular symbol is observed given the state can be described by using a $N_s \times N_o$ matrix S:

$$S_{ij} = p\left(O_k = i | x_k^b = j\right) \tag{10}$$

The dependence network is shown in Fig. 3. Given an hidden state x_k^b that evolves with time, the goal of the HMM is to find the likelihood of the set of observation symbols $\{O_1, O_2, \ldots O_k\}$ given the model. The action/model β which better describes the sequence of symbols is therefore selected as $\hat{\beta} = \underset{j=1\ldots N_b}{\operatorname{argmax}}\, p\left(O_1, O_2, \ldots O_k | j\right)$. This computation is efficiently performed using the standard Forward-Backward algorithm [22]. Each of the HMM models requires a training phase in which the parameters A_{ij}, S_{ij} and the prior probability $p\left(x_0^b\right)$ have to be learned from data. The learning phase can be efficiently performed by feeding each HMM with the sequences of the action the HMM is meant to recognize as shown in [22].

4 Joint Tracking and Behavior Recognition

In this paragraph it will be shown how the sub-modules described in the previous paragraphs can be fused to improve both the tracking and the action recognition. A joint position and behavior state is used. The position state, x_k^p,

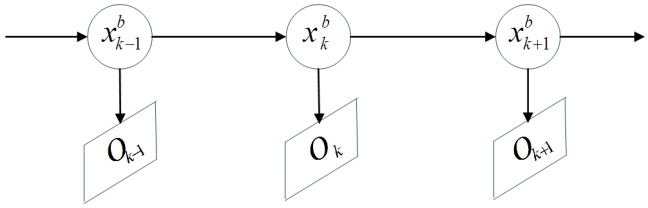

Figure 3: Graphical Model for the Action Recognition HMM

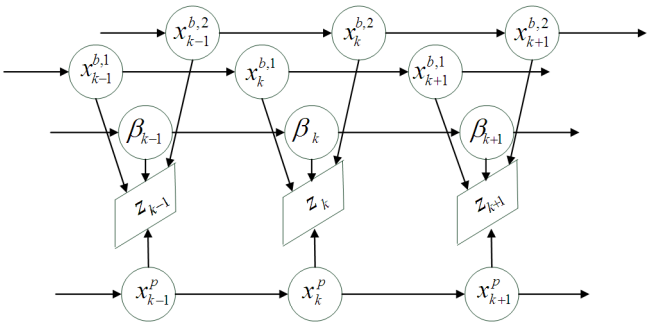

Figure 4: Graphical Model of the Joint Tracker and Behavior Recognition System

is the same described in paragraph 2. Considering that the behavior sub-module models N_b different behaviors, the behavior part of the state vector is composed by a vector of $N_b + 1$ components. The first component β_k is the ID of the model behavior and can take only discrete values in the space $\{1 \ldots N_b\}$. The other components $\left\{x_k^{b,1}, \ldots, x_k^{b,N_b}\right\}$ are the internal states of each of the sub-modules. Each of them can take only discrete values in the space $\{1 \ldots N_s\}$.

The full state vector is therefore $x_k = \left\{x_k^p, \beta_k, x_k^{b,1}, \ldots, x_k^{b,N_b}\right\}$.

The position and behavior of the target are considered as a priori independent. Each state variable evolves according to its own dynamics and is a priori uncoupled from the other state variables. The observation at time step k can depend on all the state variables at that time step. The graphical model which describes the system is shown in Fig. 4 where only two models ($N_b = 2$) have been considered for ease of visualization. The state variables at one time step, although marginally independent, become conditionally dependent given the observation sequence [23]. This can be determined by applying the semantics of directed graphs, in particular the d-separation criterion [19], to the graphical model in Fig. 4.

Inference on the graphical model can be performed with the two steps Bayesian approach shown in paragraph 2.

4.1 State Prediction

The prediction is performed by modifying the (2) considering that each state evolves according only to its past history.

The object position dynamic model is described in Eq. (4).

The states of the behavior models evolves according to their internal state transition probabilities $S^1, \ldots, S^{N_b}$ that have to be learned offline.

The probabilities of an object to switch from a behavior to the other are fixed using common sense rules. Therefore a $N_b \times N_b$ matrix B which contains the state transition probabilities is considered as available. Each element of the matrix B_{ij} defines the a priori probability that the object switches from the i-th behavior to the j-th.

$$B_{ij} = p\left(\beta_k = i | \beta_{k-1} = j\right) \tag{11}$$

4.2 Likelihood Evaluation

The likelihood is conditioned on all the state variables. It is considered as composed by two contributions: one shape-based and one behavior-based.

The first contribution (shape based observation model) tries to exploit the temporal continuity of the shape of the tracked object and tries to explain the observations at time k using a shape model that was learned during the past time-steps. The second contribution (behavior based observation model) tries to explain the observations considering that each kind of behavior typically generates a particular pattern of interest point motion.

The combined likelihood is modeled as the product of the two contributions:

$$\begin{aligned} p\left(z_k | x_k\right) = p_s\left(z_k | x_k^p\right) \cdot \\ \cdot p_b\left(z_k | x_k^p, \beta_k, x_k^{b,1}, \ldots, x_k^{b,N_b}\right) \end{aligned} \tag{12}$$

The likelihood in Eq. (5) is taken as $p_s\left(z_k | x_k^p\right)$.

The behavior based observation model is based on the rationale that different behaviors induce, in time, different shape deformation.

At each time step, considering the shape model and the observations, each candidate position x_k^p induces a different deformation.

$$\begin{aligned} & p_b\left(z_k | x_k^p, \beta_k, x_k^{b,1}, \ldots, x_k^{b,N_b}\right) = \\ & = \sum_{l=1}^{N_b} p\left(z_k | x_k^{b,l}, x_k^p\right) \delta\left(l - \beta_k\right) = \\ & = \sum_{l=1}^{N_b} p\left(O_k\left(x_k^p\right) | x_k^{b,l}\right) \delta\left(l - \beta_k\right) \end{aligned} \tag{13}$$

where $O_k\left(x_k^p\right)$ is the symbol induced by the hypothesis x_k^p considering the observations, $p\left(O_k\left(x_k^p\right) | x_k^{b,l}\right)$ is the output symbol probability of the $l-th$ model and $\delta()$ is a Kronecker delta. If, as in the system described in paragraph 3, the recognition process follows the tracking, the symbol is fixed to $O_k\left(\hat{x}_k^p\right)$, while in the proposed joint approach the induced symbol is used to enforce the estimation on x_k^p. This tracker can be easily implemented inside a particle filter framework. The posterior density $p\left(x_{k-1} | z_{1:k-1}\right)$ is represented by a set of weighted particles $\left\{x_k^{p,(n)}, \beta_k^{(n)}, x_k^{b,1,(n)}, \ldots, x_k^{b,N_b,(n)}, \pi_{k-1}^{(n)}\right\}$. The sampling-based algorithm is summarized in Algorithm 2.

Algorithm 2 Joint Shape and Behavior based Tracking

1: Resampling:
Resample the particle set $\left\{x_{k-1}^{(n)}\right\}$ based on $\pi_{k-1}^{(n)}$ to obtain $\left\{\tilde{x}_{k-1}^{(n)}\right\}$

2: Prediction:
For $n = 1 \ldots N_{particles}$

- Sample the density of the target dynamics $p\left(x_k^p | \tilde{x}_{k-1}^{p,(n)}\right)$ to obtain $\left\{x_k^{p,(n)}\right\}$
- Sample the density of the behavior transition $p\left(\beta_k | \tilde{\beta}_{k-1}^{(n)}\right)$ to obtain $\left\{\beta_k^{(n)}\right\}$
- For $b = 1 \ldots N_b$
  - Sample the density of the internal behavior model state transition $p\left(x_k^{b,(n)} | \tilde{x}_{k-1}^{b,(n)}\right)$
- end

end

3: Update:
Re-weight each particle by calculating the likelihood $\pi_k^{(n)} = p\left(z_k | x_k^{(n)}\right)$

The position and behavior at time k are estimated as:

$$\hat{x}_k^p = \sum x_k^{p,(n)} \pi_k^{(n)} \tag{14}$$

$$\hat{\beta}_k = \underset{j=1 \ldots N_b}{\operatorname{argmax}} \sum \pi_k^{(n)} \delta\left(j - \beta_k^{(n)}\right) \tag{15}$$

With these estimates the shape model is learned from frame to frame as explained in paragraph 2.

4.3 Parameter Learning

To perform target tracking, a set of parameters have to be estimated offline. In particular the A_{ij} and the B_{ij} for each behavior model have to be learned from a set of correctly tracked sequences with known behavior. If the tracked position and the behavior is considered as known, the graphical model shown in Fig. 5 represents the training phase. A key difference with the graphical model of Fig. 4 is the presence of squares (known values) where circles (state values) were present. This structure of the graphical model and the form of Eq. (13) allow to say that learning the parameters of the joint tracker is equivalent to learning N_b HMM models [23]. Each HMM model has to be trained considering only the sequences containing the action the model is meant to recognize as explained in paragraph 3.

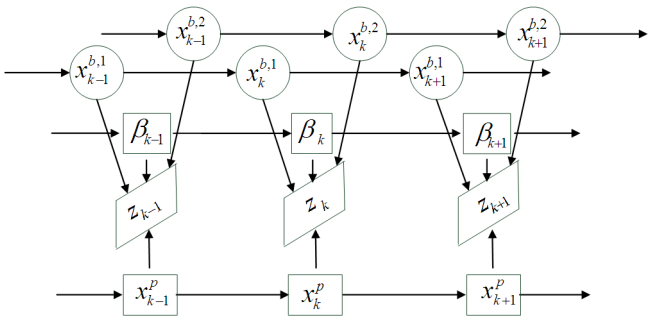

Figure 5: Graphical Model of the Joint Tracker and Behavior Recognition System during the learning phase

Algorithm 3 GHT-based implementation

1: Initialize a N_{rows} by N_{cols} empty voting space $\rightarrow V$
2: **for** $n = 1 \dots N$ **do**
3: **for** $m = 1 \dots M$ **do**
4: $V\left(z_k^n - \Delta^m\right) = V\left(z_k^n - \Delta^m\right) + 1$
5: **end for**
6: **end for**

5 Efficient GHT implementation

The computation of the shape-based likelihood and of the deformation descriptors can be time consuming if the number of particles to be evaluated is big. In this paragraph it will be shown how this process can be optimized using the Generalized Hough Transform (GHT) [17]. The purpose of the GHT is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm. The GHT has the capability to concentrate the information that is distributed in the image in the parameter space and, as it will be shown, some computations are more easily performed in the parameter space. Algorithm 3 shows how the GHT is applied to obtain the parameter space V that is used in the following.

The shape based likelihood function (5) is easily computed from V. At first the matrix L is computed convolving V with K_{motion}: $L = V \otimes K_{motion}$. It is possible to see that $p\left(z_k|x_k^p\right)$ can be evaluated by taking the element in the matrix L that corresponds to x_k^p.

$$p\left(z_k|x_k^p\right) = L\left[x_k^p\right] \tag{16}$$

The computation of V depends only on the number of the observations N and of the model points M and reduces the computational load in case a big number of particles have to be evaluated (once L has been computed, the evaluation of the likelihood is a look-up operation).

The matrix V can be useful also to compute the descriptor $d_k\left(x_k^p\right)$. To perform this computation it is worth analyzing the meaning of V from a shape distortion point of view. It is possible to observe that a look-up in V in the position $V\left[x_k^p\right]$ gives the number of observations-model points matches that, considering x_k^p as the position of the object, have occurred with zero distortion (see Algorithm 3 and eq. (5)). A look-up in V in the position $x_k^p + d$ gives the number of observations-model points matches that, considering x_k^p as the position of the object, have occurred with an interest point motion vector $v = d$. This property of the matrix V can be exploited to compute $d_k\left(x_k^p\right)$. To this purpose the supporting matrices C_r $(r = 1 \dots th)$ have to be computed in the following way:

$$C_r\left[i + th, j + th\right] = \begin{cases} 1 & if\ r - 1 \leq \sqrt{i^2 + j^2} < r \\ 0 & otherwise \end{cases} \tag{17}$$

with $i, j = -th \dots th$. These matrices represents circular rings of increasing radius and will be used to perform efficiently the histogram computation of (7). The matrices C_r have to be convolved with V to obtain th matrices $F_r = V \otimes C_r$. The descriptor is then computed as a simple look-up:

$$d_k\left(x_k^p\right)[i] = F_i\left(x_k^p\right)\ i = 1 \dots th \tag{18}$$

In this way the number of operations to be computed is not dependent on the number of particles to be evaluated. Every particles will need, assuming the th matrices F_r have been computed, only a look-up.

6 Experimental results

In this paragraph the experimental results of the proposed method are reported. The tracking capabilities of the method have been evaluated considering a typical human tracking scenario. Three different behavior models have been trained using the KTH database [24]. The KTH database consists of 25 subjects performing 6 different actions: boxing, hand-clapping, jogging, running, walking, hand-waving. In this work only the actions walking, jogging and running have been considered. The total number of videos is 300. Considering that, on average, in each video the actor enters and exits from the field of view 4 times, the total number of sequences that were used is 1200. The models have been learned considering as training data 50% of the full dataset. The parameters B_{ij}, $maxVel$ and th have been fixed in the following way: $B_{ij} = 0.8$ if $i = j$ and 0.1 otherwise; $maxVel = 30$; $th = 5$. An example of a "walking" sequence is shown in Fig. 6. In this figure the bounding box of the tracker is plotted using a solid line, the convex hull of the object model is shown using a dashed line, the tracked interest points are shown using blue dots and finally, in red, it is possible to see the estimated $\hat{x}_k^p$. In Fig. 7 one step of the Bayesian estimation process is shown. All the distributions have been marginalized to remove the internal behavior model state variables for ease of visualization. With this marginalization, the pdf has three dimensions: x, y and β. In the figure each plot on the columns corresponds to

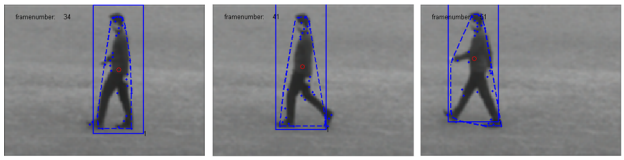

Figure 6: Example of walking sequence

a value of β while the columns and the rows of each plot correspond to x, y. The target truth position is at the center of the plot. In the first row the shape-based likelihood $p_s\left(z_k|x_k^p, \beta_k\right)$ is shown. As $p_s\left(z_k|x_k^p, \beta_k\right) = p_s\left(z_k|x_k^p\right)$, the plot is the same on all the columns. The second row shows the behavior-based likelihood $p_b\left(z_k|x_k^p, \beta_k\right)$. As it is possible to see, it has high values for $\beta_k = walk$ as it should be, considering the the sequence under analysis is a "walking" sequence. The third row shows the posterior $p\left(x_k^p, \beta_k|z_{1:k}\right)$. Also in this case it is possible to see that high values of posterior are found for $\beta_k = walk$. The capabilities of the joint tracking framework can be understood by observing that the shape-based likelihood in Fig. 7 is not monomodal. The presence of different peaks shows that at time k multiple position hypotheses x_k^p are compatible with the observations if only the continuity of the shape is considered (as the standalone tracker of paragraph 2 would do). If however this shape-based likelihood is multiplied with the behavior-based likelihood (second row of Fig. 7), the resulting likelihood is monomodal. This is due to the fact that some hypotheses, that were plausible considering only the shape, are removed if the shape information is fused with the behavior information.

Some action recognition results are shown in Fig. 8. Three different sequences containing an actor performing the three different actions where used to test the algorithm. In the figure, for each frame it is possible to see the outcome of the action estimation $\hat{\beta}_k$. As it is possible to see, the estimation is correct for nearly all the frames. The results of the tracking and behavior recognition have been then evaluated on the remaining 50% of the KTH database. A 85% correct recognition (result alligned with the state of the art) was obtained, but the interesting results is that, on the same 600 test sequences, while using only the standalone tracker there was a failure rate of the 5%, with the joint system, the failure rate decreased to 2%. Finally a result on a sequence from a different source is shown in Fig. 9. As it is possible to see, both the tracking and the behavior recognition were successful. It is worth saying that the standalone tracker failed since it was attracted by a false hypothesis generated by the clutter. This hypothesis was rejected due to the behavior based likelihood.

7 Conclusions and future work

In this work a system for the joint human tracking and behavior recognition has been proposed. The reported results demonstrated that tracking results are improved if these functions are performed together. Future works will include

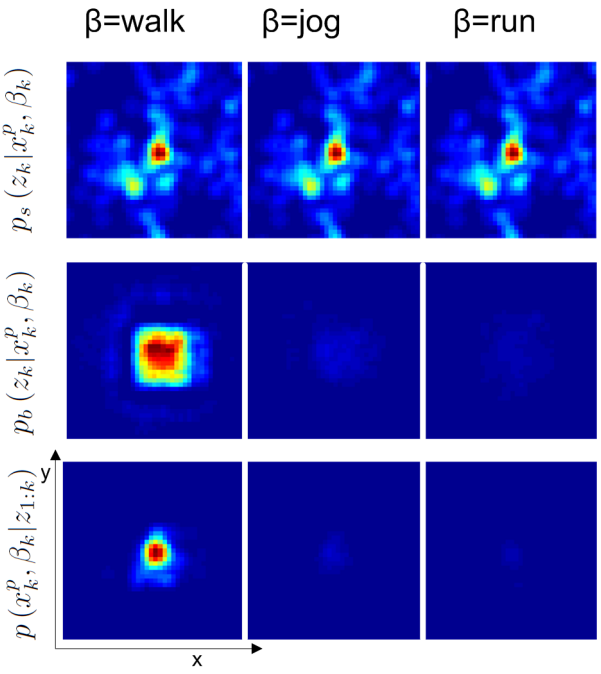

Figure 7: Example of joint tracking and behavior analysis for the sequence in Fig. 6.

the inclusion of the shape learning phase in the Bayesian estimation Framework. Another extension to the current system could be the inclusion of a module to compute behavior transition probability and position dynamics using high level contextual information.

References

[1] C. Hue, J. Cadre, and P. Perez, "A particle filter to track multiple objects," in *Proc. IEEE Workshop on Multi-Object Tracking*, 2001, pp. 61–68.

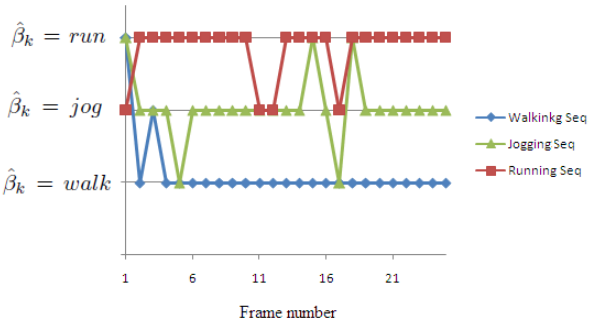

Figure 8: Behavior recognition results. Three sequences with different actions.

Figure 9: Tracking in case of clutter

[2] L. Li, W. Huang, I. Y.-H. Gu, R. Luo, and Q. Tian, "An efficient sequential approach to tracking multiple objects through crowds for real-time intelligent cctv systems," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 38, pp. 1254–1269, 2008.

[3] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Trans. on PAMI*, vol. 26, pp. 1531–1536, 2004.

[4] M. Asadi and C. Regazzoni, "A comparison of different approaches to nonlinear shift estimation for object tracking," in *Proc. ICIP*, 2007.

[5] T. Mathes and J. Piater, "Robust non-rigid object tracking using point distribution models," in *Proc. BMVC*, 2005.

[6] S. Avidan, "Support vector tracking," *IEEE Trans. on PAMI*, vol. 26, pp. 1064–1072, 2004.

[7] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Trans. on PAMI*, vol. 30, pp. 1728–1740, 2008.

[8] Y. Wang and G. Mori, "Human action recognition by semilatent topic models," *IEEE Trans. on PAMI*, vol. 31, no. 10, pp. 1762–1774, 2009.

[9] K. Schindler and L. van Gool, "Action snippets: How many frames does human action recognition require?" in *Proc. CVPR*, 2008, pp. 1–8.

[10] Y. Liang, S. Shih, A. Shih, H. Liao, and C. Lin, "Learning atomic human actions using Variable-Length markov models," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, pp. 268–280, 2009.

[11] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," in *First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2006.

[12] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using Spatial-Temporal words," *Int. J. Computer Vision*, vol. 79, pp. 299–318, 2008.

[13] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Proc. ICCV*, 2009.

[14] M. Ahmad and S. W. Lee, "Human action recognition using shape and CLG-motion flow from multi-view image sequences," *Pattern Recognition*, vol. 41, no. 7, pp. 2237–2252, 2008.

[15] M. Isard and A. Blake, "A mixed-state condensation tracker with automatic model-switching," in *Proc. ICCV*, 1998.

[16] Y. Wu, G. Hua, and T. Yu, "Switching observation models for contour tracking in clutter," in *Proc. CVPR*, 2003.

[17] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes." *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. ECCV*, 2006.

[19] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, September 1998.

[20] N. Toronto, B. Morse, D. Ventura, and K. Seppi, "The Hough transform's implicit bayesian foundation," in *Proc. ICIP*, 2007, pp. 377–380.

[21] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, pp. 174–188, 2001.

[22] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[23] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, no. 2, pp. 245–273, November 1997.

[24] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proc. ICPR*, vol. 3, 2004, pp. 32–36.