Human shape-motion analysis in athletics videos for coarse to fine action/activity recognition using transferable belief model

Emmanuel Ramasso, Costas Panagiotakis, Michèle Rombaut, Denis Pellerin, Georgios Tziritas

Electronic Letters on Computer Vision and Image Analysis 7(4):32-50, 2009

Human Shape-Motion Analysis In Athletics Videos for Coarse To Fine Action/Activity Recognition Using Transferable Belief Model

Emmanuel Ramasso*, Costas Panagiotakis+, Michèle Rombaut*,
Denis Pellerin* and Georgios Tziritas+
* *GIPSA-lab, Images and Signal Department, 38031 Grenoble, France*
+ *Department of Computer Science, University of Crete, P.O. Box 2208, Heraklion, Greece*

Received 9th May, 2008; accepted 26th February, 2009.

Abstract

We present an automatic human shape-motion analysis method based on a fusion architecture for human action and activity recognition in athletic videos. Robust shape and motion features are extracted from human detection and tracking. The features are combined within the Transferable Belief Model (TBM) framework for two levels of recognition. The TBM-based modelling of the fusion process allows to take into account imprecision, uncertainty and conflict inherent to the features. First, in a coarse step, actions are roughly recognized. Then, in a fine step, an action sequence recognition method is used to discriminate activities. Belief on actions are made smooth by a Temporal Credal Filter and action sequences, i.e. activities, are recognized using a state machine, called belief scheduler, based on TBM. The belief scheduler is also exploited for feedback information extraction in order to improve tracking results. The system is tested on real videos of athletics meetings to recognize four types of actions (running, jumping, falling and standing) and four types of activities (high jump, pole vault, triple jump and long jump). Results on actions, activities and feedback demonstrate the relevance of the proposed features and as well the efficiency of the proposed recognition approach based on TBM.

Key Words: Video Analysis, Human Tracking, Action and Activity Recognition, Transferable Belief Model.

1 Introduction

Human motion analysis has many applications in many areas, such as analysis of athletic events, surveillance, content-based image storage and retrieval. The main scientific challenges in human motion analysis are to detect, track and identify people and to recognize the human activity [1] from observations coming from video. The detection and tracking algorithms are challenged by occluding and fast/complicated moving objects, as well as illumination changes.

1.1 Related work

A combination of human shape-motion features estimation, silhouette analysis, skin color detection, template matching, 2–D/3–D human modeling, background modeling have been used on human detection and tracking

Correspondence to: <e_ramasso@yahoo.fr>, <cpanag@csd.uoc.gr>

systems. According to the application, single/multiple or static/moving cameras has been used. The silhouettes are easy to extract providing valuable information about the position and shape of the person. There are model based approaches and systems using Shape-From-Silhouette methods to detect and track the human in 2D [2] or 3D space [3]. When the camera is static, background subtraction techniques can give high accuracy measures of human silhouettes by modeling and updating the background image [4]. Otherwise, when the camera is moving, camera motion estimation methods [5, 6] can locate the independently moving objects.

The system called W4 [7] is based on a statistical-background model to locate people and their parts (head, hands, feet, torso, etc.) using static cameras and allowing multiple person groups. Wang, Hu and Tan [8] emphasize on three major issues of human motion analysis systems, namely human detection, tracking and activity understanding. Figueroa et al. [9] propose a system of tracking soccer players using multiple static cameras. The occlusions have been treated by splitting segmented blobs based on morphological operators and a backward and forward graph representation based on human shape, motion and color features. However, in a real soccer game, there are crowd situations, where the people should be manually tracked. Cheng and Chen [10] propose a method for detecting and tracking multiple moving objects based on discrete wavelet transform and identifying the moving objects by their color and spatial information using a stable camera. The human detection is done using the low band of the wavelet transform of the image due to the fact that most of the fake motions in the background can be decomposed into the high frequency wavelet sub-band.

Many methods have been proposed for action recognition [8] notably based on *classification*, *template matching* and *neural networks*. Generally, the methods are based on the *Bayesian framework* with *Hidden Markov Models* (HMM) [11, 12] and *Dynamic Bayesian Network* (DBN) [12]. Other methods are developed in Artificial Intelligence community notably *Petri Nets* [13]. In previous work, a novel architecture utterly based on the Transferable Belief Model [14] (TBM), an interpretation of Shafer's theory of evidence [15, 16], was proposed [2, 17] for human action and activity recognition in athletic sports videos. The TBM is well-suited for action recognition notably because doubtful transitions between actions are explicitly modeled, conflict between features reflects the need to improve the fusion process and reliability of features depends on the context and can be included in the system. Belief theory has been successfully applied on other pattern recognition problems, e.g. human postures classification [18] and emotions recognition [19].

The most of aforementioned human motion analysis and activity recognition methods suppose static cameras. They have been generally tested using videos with simple action such as walking people. Moreover, generally constrained indoors or outdoors environments are assumed getting high accuracy results of human activity recognition and human detection and tracking.

In Figure 1(b), the silhouette quality is high, since accuracy of human boundary estimation is high and the number of wrong classified pixels is low. A challenging problem appears when the camera is moving and the estimated human silhouettes are of low quality or extremely wrong (see Figure 1(d)).

1.2 Contributions

The presented work focuses on real videos of athletics acquired by a moving camera without initialization and any assumption or knowledge about its motion. Moreover, the videos used present real and unconstrained environments with other moving people. Videos come from various sources of athletics meeting such as broadcast TV, Internet and DVD. These videos present a dynamic environment, almost unconstrained, a varying quality and in which the athlete's motion is extremely fast and complicated. We suppose that the camera tracks the athlete and we test the algorithms of tracking and recognition in individual sports such as pole vault, high jump, triple jump and long jump (called activities) in which we recognize actions (such as running, jumping, falling and standing).

Camera motion as human action feature was a few used [20, 21]. This feature is important since action implies motion as emphasized by a recent work of Irani [22] where optical flow is exploited for action recognition at distance applied to field-view videos such as football. In this paper we rather focus on one athlete and we are interested in decomposing his movement (the proposed algorithms can thus be used for other state sequence

recognition such as gesture). For instance, given a video, one of our objective is to not only recognize one jump among a list of possible ones but also to detect correctly actions. On this point, we have a quite different approach compared to usual HMM-based methods which do not focus on actions within activities (or one needs to use one HMM for each action). Moreover, in usual probability-based methods, generally mixtures of Gaussians are used in "black boxes" where user feedback is almost impossible. In this paper, we propose a high level fuzzy-based description of actions using rules. New rules are easily added. The used of fuzzy description is explained by the fact that we focus on multmedia applications where expert knowledge and user feedback are important and useful. An original method to recognize actions and activities simultaneously is proposed (based on a conference work [23]). The method is online and we propose a new criteria for inference. The proposed algorithm is also exploited for feedback on tracking. The method we propose uses camera motion to obtain a global information about human actions. The detection of actions is then refined using more precise features and sequences. Note that we use the algorithm proposed in [24] in order to detect whether a video concerns individual sports (which is processed) or group of athletes (which is not processed). This algorithm alleviates a great assumption concerning the content of the video but is not described in this paper.

One characteristic of the proposed system is that it works automatically, recognizing action and activities without any initialization or prior knowledge about camera motion and human features, providing also statistical results about athlete motion. Fuzzy rules need expert knowledge which is available in athletics videos. Another contribution of this work is the use of Transferable Belief Model (TBM) [14] for static and dynamic action and activities recognition. Related work concerns only the use of Dempster-Shafer theory [15, 16] for static but not dynamic recognition [25, 26]. So the proposed algorithms are original.

(a) (b) (c) (d)

Figure 1: **(a), (b)** Original image and the silhouette estimated by the method of [7] under stable camera. **(c), (d)** Original image of an individual sport (long jump) the silhouette estimated by the method of [5] under moving camera.

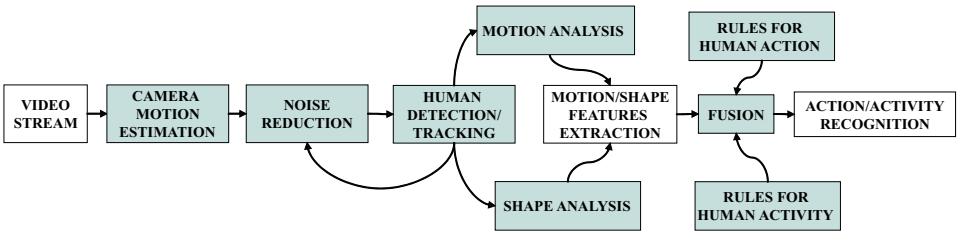


Figure 2: Schema of the proposed system architecture.

The proposed architecture consists of several main modules (Figure 2):

1. Silhouettes are computed using a camera motion estimation method [5], where an affine model is used to describe the camera motion. Such a model is generally sufficient for most of real video sequences. The above method that we use, was implemented by the Vista Team of IRISA.

2. A silhouette noise reduction procedure is executed next and human detection and tracking is performed. Four major human points are recognized and tracked using the human silhouettes.

3. Shape and motion analysis is executed, in order to extract relevant shape and motion features that can be used on action and activity recognition. The pole detection procedure (a shape analysis method), is applied to the human silhouette detecting the pole and extracting features related to it such as its eccentricity and its position. The human major points can be recomputed after a pole removal.

4. A fusion architecture, based on TBM, is used for action/activity recognition. The input features for the fusion process include camera motion, pole detection and human shape-motion features estimated by the corresponding modules. The results of the fusion process can be used as feedback information improving the results of human tracking.

The rest of the paper is organized as follows: Section 2 presents the human shape-motion analysis method. Section 3 describes the action/activity recognition and feedback method. Finally, Sections 4 and 5 provide experimental results and the discussion, respectively.

2 Human Shape and Motion Analysis

In order to detect action and activities, it is required to extract relevant features. Real videos of athletics are noisy therefore the colour feature is not reliable. In this section, we describe methods to extract human shape and motion features.

In pole vault videos, the athlete's pole can disturb the tracking and action recognition because it is moving with the athlete. To cope with this problem, a robust shape based method for pole detection and deletion is proposed. Then, some major human points are tracked using a shape-motion based technique. These algorithms are applied on binary images obtained from the camera motion estimation method by some simple morphological operations to reduce noise and create quite homogeneous area of moving pixels (the silhouette obtained can be in several pieces).

2.1 Pole detection

The pole is recognized first since it can be easily detected by its shape, which has very high eccentricity comparing with the human members. The eccentricity ($\varepsilon \geq 1$) is defined by the ratio between the two principal axes of the best fitting ellipse, measuring how thin and long a region is. If the detected region has high ε (e.g. more than 20) then it is probably a pole. This feature is relevant in the fusion process to recognize the pole vault videos.

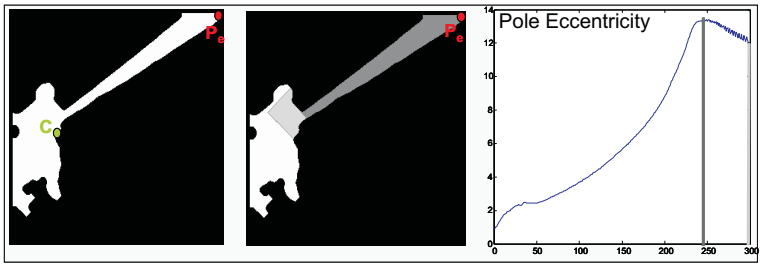


Figure 3: The two main steps of the pole detection method: The detection of points C, P_e (left image) and the region growing algorithm which detects the pole (heavy gray pixels on the middle image). The eccentricity per region is shown on the right image, its maximum value corresponds to the detected pole region.

The eccentricity can be defined by the three second order moments $\mu_{1,1}, \mu_{2,0}$ and $\mu_{0,2}$:

$$\varepsilon = \sqrt{\frac{\mu_{2,0} + \mu_{0,2} + \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4 \cdot \mu_{1,1}^2}}{\mu_{2,0} + \mu_{0,2} - \sqrt{(\mu_{2,0} - \mu_{0,2})^2 + 4 \cdot \mu_{1,1}^2}}} \quad (1)$$

with $\mu_{p,q} = \sum_{(x,y) \in O_t^i} (x - x_c)^p (y - y_c)^q$ where (x_c, y_c) are the coordinates of the mass center of the object (defined as the mass center of the object pixels). Based on this definition, the pole detection procedure (Figure 3) is described as follows:

- First, the highest area object (O_1) is detected, which is defined as follows. The silhouette is possible to consist of more than one objects (see Fig. 1(d)), due to noise effects. For each object of them we compute its area in pixels, then O_1 is defined as the object of maximum area. Then, the end of pole point (P_e) is estimated. P_e is defined as the farthest O_1 point from the mass center (C) of O_1 object under the constraint that it lies above the C as the athlete is running.

- The pole pixels will be detected by a region growing method (RG) starting from P_e point. RG stores in a stack the added points. In each iteration step, RG adds a pixel from boundary of region of added points and the rest object in the stack. This method terminates when the area of region exceeds the 50% of the O_1 area or when the number of pixels of the boundary between the region and O_1 exceeds a threshold. The threshold is a percentage (e.g. 40%) of the square root of the O_1 area approximating the double of O_1 mean width.

- However, the region will have been expanded in the athlete area. Therefore, we have to ignore the last pixels that RG adds, until the region where ε will be maximum (Figures 3 and 4). Let O_2 be the estimated pole region. We compute the distance d between the farthest point (P_f) of O_2 from P_e and P_e itself. Then, ε can be estimated by the ratio $\varepsilon = \frac{\pi d^2}{O_2\ area}$. P_f can be approximated directly by the last point that the RG method adds.

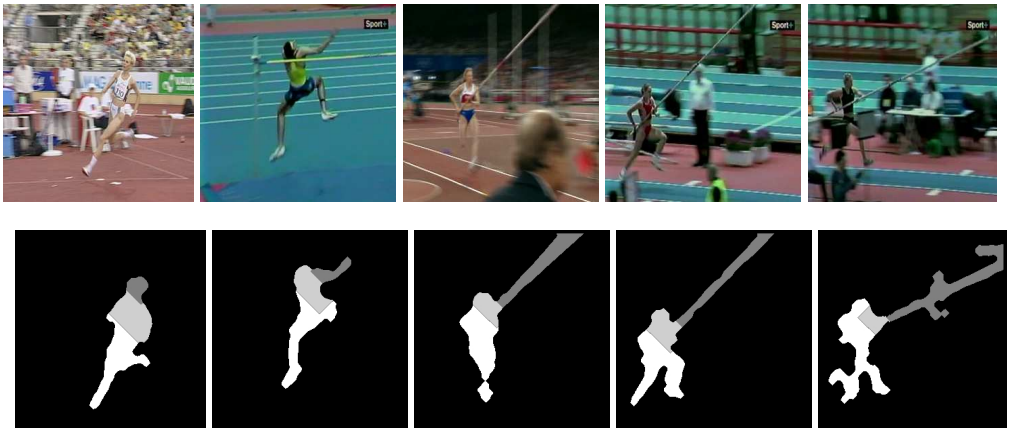
- Finally, the estimated pole region (O_2) is characterized as pole if its shape is like the pole's shape. We measure this similarity using the region eccentricity. If ε is higher than a threshold (e.g. 20) and the region length is at least 25% of the O_1 length then the O_2 object will be a pole.

The proposed pole detection method detects the pole with high accuracy (about 90% without false alarms) and robustness to silhouette noise (see Figure 4(e)). We can recognize if the detected region is pole (gray pixels of Figure 4(e)) using a threshold on detected region eccentricity ($\varepsilon \leq 20$). When the eccentricity is high then the pole is deleted. The strong point of this method is that it is simple and low cost. The results on our database show a great performance of this detector. Notice that using P_e and P_f points, we can compute the slope of the detected pole (not used in this paper).

2.2 Points detection and tracking

Real athletics videos can be of bad quality (provided my home's TV recorder or on Internet) therefore details on athletes are not available, only the rough positions of "major" points can be obtained. We assume that the head center, the mass center, the left end of leg and the right end of leg (see Figure 5) are sufficient for global action recognition (such as running, jumping, falling or standing-up). Moreover these points remain quite visible along a video sequence.

These four major points are detected and tracked using human silhouettes as input. The method is divided into two procedures: detection and tracking. This method is an extension of [17], where three major human points are detected and tracked.



(a) (b) (c) (d) (e)

Figure 4: Results of pole detection procedure. The light gray pixels denote those that ignored (last added) by the RG method and the gray pixels denote the detected pole region. **(a)** $\varepsilon = 6.08$, **(b)** $\varepsilon = 12.24$, **(c)** $\varepsilon = 31.27$ **(d)** $\varepsilon = 50.01$, **(e)** $\varepsilon = 31.32$.

2.2.1 Detection

In this step, the four major human points are automatically detected. This procedure is executed just once, in the first silhouette frame of the sequence. The previous position of the four major human points is unknown, so we assume that the human stands vertically in the first frame (the head lies above the mass center). The algorithm named "Human Points' Detection" is executed as it is described hereafter.

- First, the mass center point is computed. This point is defined as the mass center of the foreground pixels. The other major human points belong on human boundary. We compute them under this restriction using the precomputed mass center. Thus their search space is reduced.

- Next, the human body major axis (Figure 5(b)) is computed using second order moments:

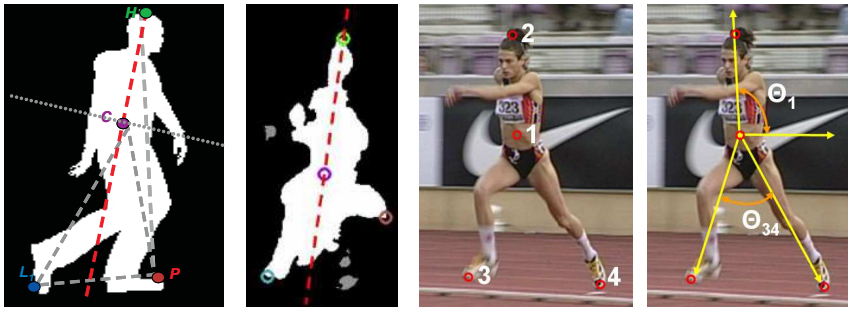
$$\theta = \arctan\left(\frac{2 \cdot \mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right), \quad \theta \in [0, 180] \tag{2}$$

  The head point (H) is defined as the farthest major axis point from the mass center (C), that lies above the mass center.

- Then, the end of leg points search space is reduced to the silhouette boundary points S that are found under the mass center. This property can be expressed by the following constraint $\vec{CH} \cdot \vec{CL} < 0.1 \cdot |CH|^2$, $L \in S$. The first end of leg point (L_1) can be computed by getting the farthest foreground pixel from the C, that lies below the C.

- The next end of leg point (L_2) should have the following properties: high distances from C, H and L_1. Moreover, the triangle PCL_1 should be close to an isosceles triangle, where P denotes a candidate L_2 point. The last two constraints are equal to the triangle area ($E(PCL_1)$) maximization. Thus, the maximization of product ($|PH| \cdot |PC| \cdot E(PCL_1)$) provides the L_2 point.

Figure 5(a) illustrates graphically the predefined symbols. Finally, it is trivial to distinguish the leg points L_1, L_2 to the left and right end of leg points using the human major axis.



(a) (b) (c) (d)

Figure 5: **(a),(b)** Estimated human points: head center (green point), mass center (magenta point), left end of leg (blue point) and right end of leg (brown point). The human body major axis is shown as a red dashed line. **(c)** The four major human points. **(d)** The two characteristics angles: the human major axis angle (Θ_1) and the angle between legs (Θ_{34}).

2.2.2 Tracking

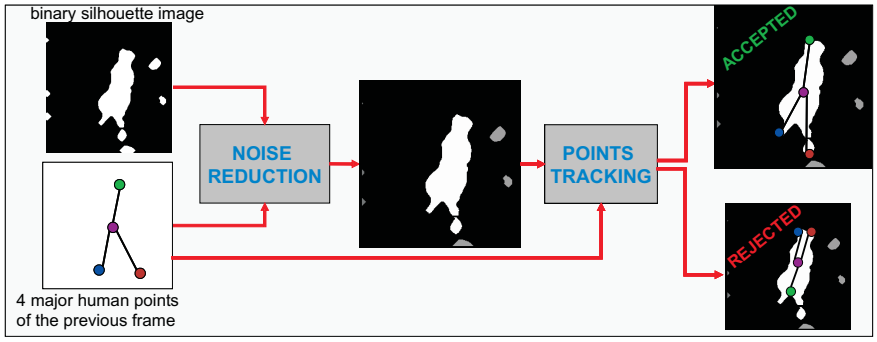


Figure 6: Individual points tracking schema.

In this step, the four major human points are tracked. This procedure is executed in every frame of the sequence, apart from the first one, taking as input the position of the four major human points in the previous frame and the current silhouette image (Figure 6). Finally, the position of the four major human points in the current frame is estimated.

First, a noise reduction procedure is applied which reclassifies the binary silhouette image pixels in order to reduce the number of wrongly classified pixels. For that, we compute the minimum distance of each foreground (white and moving) object from the previous position of the four human points. We then multiply it by the percentage of foreground pixels that belong to a line segment started at the mass center of the foreground object and ended on the specific major human point. If this distance is higher than a threshold then the foreground pixels will be classified to background class (gray pixels of Figure 5(b)). Next, we reclassify all background pixels that belong to human silhouette holes to foreground class.

The four major human points can be detected by “Human Points’ Detection” algorithm which has been described in the previous section. This method produces two pairs of solutions for the head point and the leg points, as it is unknown if the head point lies above or under the mass center. We choose the closest pair compared to the estimated pair in the previous frame (see Figure 6).

2.3 Human shape-motion features

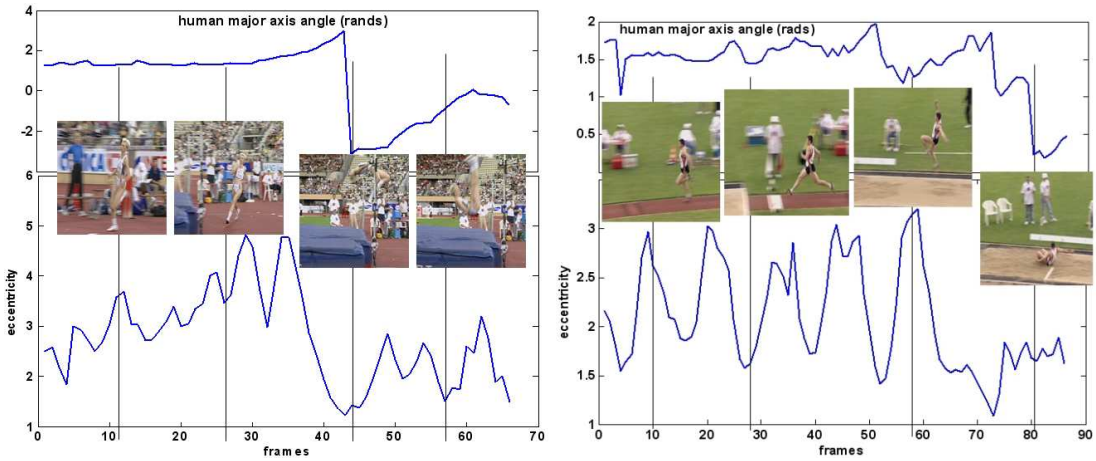


Figure 7: The human eccentricity and the human body axis angle in a high jump and a long jump sequence.

Using the results of pole detection and points tracking, we can compute shape-motion features useful for action recognition. The estimated pole eccentricity (ε) is relevant shape feature since we can recognize if the detected region is a pole. It can also be used to detect dropping bar during jumping or falling stages in high jump and pole vault. The time-varying human silhouette ε is an important shape feature, because it is related with the human action. For example, the human eccentricity ε is lower during the jumping of a high jump sequence and it is useful for gait period estimation. We compute these features using the estimated silhouette (after the noise removal) by the points tracking procedure.

The motion based features are computed from the major points trajectories. One important feature concerns the vertical translation of the mass center (P_{msvt}). Then, the angle between the human major axis and the horizontal axis (Θ_1) (see Figures 5(d) and 7) is of key of importance for action discrimination. If this angle is about 90^o, the human is standing or running, whereas important variation occur during the jumping and falling in high jump and pole vault. Moreover, the angle between the legs (Θ_{34}) (see Figure 5(d)) is another relevant feature. Indeed, the gait period can be measured from its trajectory providing an estimation of the human speed. The camera motion features are also exploited for action recognition: the camera horizontal translation (P_{cht}), the camera vertical translation (P_{cvt}), and the camera zoom (P_{cz}).

Finally, a set of 6 features are automatically computed at each frame and are used for action recognition. They are listed in Table 1.

Vertical translation of the mass center	P_{msvt}
Angle between the human major axis and the horizontal axis	Θ_1
Angle between the legs	Θ_{34}
Camera horizontal translation	P_{cht}
Camera vertical translation	P_{cvt}
Camera zoom	P_{cz}

Table 1: Features used for action recognition.

3 Human action and activity recognition

The features described previously are now combined within the axiomatically well-founded Transferable Belief Model (TBM) framework proposed by Smets and Kennes [14]. The goal is to obtain a global belief on actions which takes features imprecision, uncertainty and conflict into account.

Since usually probabilistic methods are applied (in Computer Vision applications), the reader may refer to both Philippe Smets' homepage and Thierry Denoeux's homepage in order to be convinced about the relevance of TBM: many applications (medical, diagnosis, target identification, ...) and comparisons with fuzzy and probabilistic methods are proposed. Roughly, TBM is a more general framework than probability theory and is based on Shafer's Theory of Evidence [15]. It relies on belief functions which allows to explicitly model doubt whereas doubt is implicit in probability. The Bayes theorem was also generalized in TBM [27] yielding to many possibilities for TBM-based networks. The TBM also emphasizes conflict between hypotheses which is an original and strong advantage compared to other formalisms.

The system works as follows: 1) features are converted into belief on symbols and such that doubt is explicit [17], 2) for each action, separately, beliefs are combined according to predefined rules using TBM framework [17], 3) beliefs on each action are made smooth and coherent using the Temporal Credal Filter [28], 4) a sequential data analysis method based on TBM and called Belief State Scheduler is applied to recognize sequences of actions [23], and finally 5) a quality criterion is computed for each action and each activity in an online manner in order to infer action and activities at each time. Note that we only describe briefly the system and the reader may refer to [17, 28, 23] for more details and illustrations. In this paper, we also propose a coarse-to-fine activity recognition: instead of combining every features blindly, we exploit features characteristic in order to combine them hierarchically.

3.1 From numerical features to belief on actions

An action A is described by two states gathered in the frame of discernment (FoD) $\Omega_A = \{R_A, F_A\}$ with R_A (resp. F_A) stands for "action A is right" (resp. "A is *false*"). A basic belief assignment (BBA) on an A according to a feature P is defined on the set of propositions $2^{\Omega_A} = \{\{\emptyset\}, \{R_A\}, \{F_A\}, \{R_A \cup F_A\}\}$ (for sake of simplicity the braces will be omitted, i.e. $\{F_A\}$ will be written F_A) by $m_P^{\Omega_A} : 2^{\Omega_A} \rightarrow [0,1]$, $X \rightarrow m_P^{\Omega_A}(X)$ and by construction $m_P^{\Omega_A}(\emptyset) = 0$, and $\sum_{X \subseteq \Omega_A} m_P^{\Omega_A}(X) = 1$. The proposition $R_A \cup F_A$ explicitly represents the doubt concerning the real state of an action: it does not imply any additional claims regarding the subsets, i.e. neither R_A nor F_A. This is a fundamental difference with a probability measure which is additive.

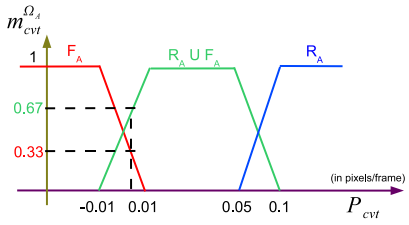
A fuzzy-set inspired method [17] (using trapezoids) is used to convert each numerical feature (described in Section 2.3) into sources of belief. An illustration is depicted Figure 8(a). Trapezoids learning can be made using expert knowledge (if features are understandable as it is the case in this paper) or statistics. Belief synthesizing is performed frame by frame. In usual probabilistic methods, the couterpart of the TBM-fuzzy-set is the mixture of Gaussians.

3.2 Transferable Belief Model fusion

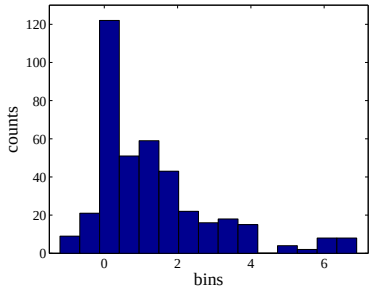
Belief of features are combined in the TBM framework [14] in order to obtain a global belief on actions which takes features imprecision, uncertainty and conflict into account. The fusion process is performed frame by frame for each action independently by rules of combination defined for two distinct BBAs $m_{P_1}^{\Omega_A}$ and $m_{P_2}^{\Omega_A}$ by:

$$m_{P_1}^{\Omega_A} \textcircled{\triangle} m_{P_2}^{\Omega_A}(E) \quad = \quad \sum_{C \triangle D = E} m_{P_1}^{\Omega_A}(C).m_{P_2}^{\Omega_A}(D) \tag{3}$$

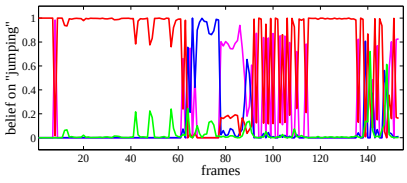
with $\triangle = \cap$ (resp. $\cup$) for the conjunctive (resp. disjunctive) rule of combination. The rules of combination can be used in logical rules such as "*if* ... AND ... OR ... *then* ..." for describing actions by means of features states. These logical rules are then translated into belief combinations where the logical AND is replaced by the



(a) Doubt is explicitly described by means of $R_A \cup F_A$. Ex: if $P_{cvt} = 0.005$ then $m^{\Omega_A}_{cvt}(R_A) = 0$, $m^{\Omega_A}_{cvt}(R_A \cup F_A) = 0.67$ and $m^{\Omega_A}_{cvt}(F_A) = 0.33$.



(b) Histogram of camera vertical motion feature values over a pool of high jumps. This is used for trapezoids thresholds estimation.



(c) Belief on jumping after fusion of camera vertical translation and variation of center of gravity. Legend: action true in blue, action false in red, conflict in magenta and doubt in green.

Figure 8: A BBA for an action "jumping" (J) in a high jump obtained by the fusion of camera vertical motion and center of gravity variation features.

⋒-rule and the logical OR by the ⋓-rule assuming the same FoD [14]. One can also use hand-made table of rules, this approach is well suited when expert knowledge is available as in [29] for medical diagnosis.

We propose to use first a coarse definition of actions using the following rules (see Section 2.3 for symbols):

IF ($\mathrm{P_{cht}}$ is high **OR** $\mathrm{P_z}$ is high **OR** $\mathrm{P_{msvt}}$ is almost null) **THEN** ($\mathrm{R_n}$ is true)

IF ($\mathrm{P_{cvt}}$ is highly positive **OR** $\mathrm{P_{msvt}}$ is highly positive) **THEN** ($\mathrm{J_n}$ and $\mathrm{U_n}$ are true)

IF ($\mathrm{P_{cvt}}$ is highly negative **OR** $\mathrm{P_{msvt}}$ is highly negative) **THEN** ($\mathrm{F_n}$ is true)

Some reliability factors are also integrated in equation (3) (before combination by the rules) as described in [23]. Reliability is an important tool for action recognition in video (not really exploited until now) because it gives a penalty on belief provided by sources that work in non-optimal conditions. Reliability factors are automatically computed from data at each frame of the video and in an online manner to take into account the reliability that may vary, in particular according to the video quality. A coefficient of reliability, denoted $\alpha \in [0, 1]$ (its dual is called the discount factor with value $(1 - \alpha)$), is applied on a belief $m^{\Omega_A}_P$ (defined on Ω_A for an action A) and a new belief m^{α,Ω_A}_P is obtained as follows:

$$\begin{aligned} m^{\alpha,\Omega_A}_P(X) &= \alpha.m^{\Omega_A}_P(X) \qquad \forall X \subsetneq \Omega_A \\ m^{\alpha,\Omega_A}_P(\Omega_A) &= (1-\alpha) + \alpha.m^{\Omega_A}_P(\Omega_A) \end{aligned} \tag{4}$$

The belief of each proposition is discounted and the remaining of the belief mass is transferred onto ignorance ($R_A \cup F_A$). For example, let $m^{\Omega_A}_P(\emptyset) = 0.12$, $m^{\Omega_A}_P(T_A) = 0.55$, $m^{\Omega_A}_P(F_A) = 0.07$ and $m^{\Omega_A}_P(\Omega_A) = 0.26$. Let the discount factor be $\alpha = 0.74$ at time t. The discounted BBA is $m^{\alpha,\Omega_A}_P(\emptyset) = 0.09$, $m^{\alpha,\Omega_A}_P(T_A) = 0.41$, $m^{\alpha,\Omega_A}_P(F_A) = 0.05$ and $m^{\alpha,\Omega_A}_P(\Omega_A) = 1 - 0.74 + 0.19 = 0.45$.

Expert knowledge or statistics can be used to compute this coefficient. In [30], discounting factors are computed using distance measures and risk minimization. Our methodology consists rather in computing automatically the discount coefficients from data at each frame. Two discount coefficients are automatically computed: one for tracking (α_{dist}) and one for camera motion estimation (α_{sup}). The computation, at each frame, of those coefficients are as follows:

– α_{dist}: the distance between the center of gravity and the head is assumed to be close between two successive frames. The distance is normalized into $[0, 1]$ (by using the size of the image) and is used as a coefficient of reliability. When the distance is constant, the coefficient is close to 1 so the reliability

is high and vice-versa. This coefficient reflects the quality of the tracking: when other moving objects appear, the binary silhouette can be of bad quality and so does the tracking.

– α_{sup}: the number of pixels of the silhouette is assumed to be quite constant between two successive frames. This number is computed after the binarization. The relative difference between two successive frames is converted into a quality coefficient using a fuzzy set with core $[0.9, 1.0]$ and support $[0.75, 1.0]$. With this conversion, if the variation is greater than 25% then the motion estimation is not reliable ($\alpha_{sup} = 0$). This coefficient allows to discount features coming from the camera motion estimation.

A system originally based on TBM was recently proposed for belief filtering and data sequence analysis using belief functions and TBM framework. We describe hereafter the main points of the system and the reader may refer to [28, 17, 23] for details and illustrations.

3.3 Temporal Credal Filter for action state filtering

The Temporal Credal Filter (TCF) proposed in [28] makes belief on actions temporally consistent (the resulting belief has no conflict and made smooth) and separates action states (assumed to be true or false). The TCF works on-line on each action independently taking as input the BBA obtained from features fusion and the previous TCF output. The system is described in Figure 9(a). In this section, the main points of the TCF process are recalled [28].

The TCF uses a model of belief evolution $\mathcal{M} \in \{\mathcal{T}, \mathcal{F}\}$, one for each state ($\mathcal{T}$ for T_A^f and $\mathcal{F}$ for F_A^f). Only one model is applied at each time f and each model assumes that the BBA of the current TCF output $m^{\Omega_A^f}$ at frame f is close to the previous one $m^{\Omega_A^{f-1}}$ (this is a common hypothesis in filtering, in particular for our application since human motions are continuous). A model of evolution can be viewed as an equivalent to conditional probability tables but in the TBM context.

1-Prediction: A model of evolution is used to predict the current state of each action $\hat{m}_{\mathcal{M}}^{\Omega_A^f}$ (at time f) by combining the BBA of the current model of belief evolution and the previous TCF output $m^{\Omega_A^{f-1}}$ resulting in two possible BBA [28]: either $\hat{m}_{\mathcal{T}}^{\Omega_A^f}$ if the current model is $\mathcal{T}$ or $\hat{m}_{\mathcal{F}}^{\Omega_A^f}$ if the current model is $\mathcal{F}$. These BBAs are given by: $\hat{m}_{\mathcal{T}}^{\Omega_A^f}(T_A^f) = \gamma_{\mathcal{T}} \cdot m^{\Omega_A^{f-1}}(T_A^{f-1})$, $\hat{m}_{\mathcal{T}}^{\Omega_A^f}(\Omega_A^f) = \gamma_{\mathcal{T}} \cdot m^{\Omega_A^{f-1}}(\Omega_A^{f-1}) + 1 - \gamma_{\mathcal{T}}$ for the first case, and $\hat{m}_{\mathcal{F}}^{\Omega_A^f}(F_A^f) = \gamma_{\mathcal{F}} \cdot m^{\Omega_A^{f-1}}(F_A^{f-1})$, $\hat{m}_{\mathcal{F}}^{\Omega_A^f}(\Omega_A^f) = \gamma_{\mathcal{F}} \cdot m^{\Omega_A^{f-1}}(\Omega_A^{f-1}) + 1 - \gamma_{\mathcal{F}}$ for the second case.

A method has been proposed in [31] in order to estimate $\gamma_{\mathcal{M}}$ that we can not describe in this paper due to limited space. In this paper we have always set these parameters to 0.9.

2-Fusion of prediction and measure: $m^{\Omega_A^f} = \hat{m}_{\mathcal{M}}^{\Omega_A^f} \textcircled{\cap} \tilde{m}^{\Omega_A^f}$ combines the available information, where the operator $\textcircled{\cap}$ is the conjunctive rule of combination defined in equation 3.

3-Conflict: $\epsilon_f = m^{\Omega_A^f}(\emptyset_A^f)$ quantifies the contradiction between model of belief evolution and data. The higher the conflict, the higher the necessity to change the current model.

4-Cusum: $\mathbf{CS}(f) = \lambda \times \mathbf{CS}(t-1) + \epsilon_f$ builds the cumulative sum of conflict along time, and $\lambda \in [0, 1]$ is a fader coefficient to cope with low/high variation of conflict (smoothing).

5-Decision on model change: when the cumulative sum is too high, i.e. if $\mathbf{CS}(f) > \mathcal{T}_s$ (stop threshold) at frame f_s, the model is changed. The new model is applied from f_s.

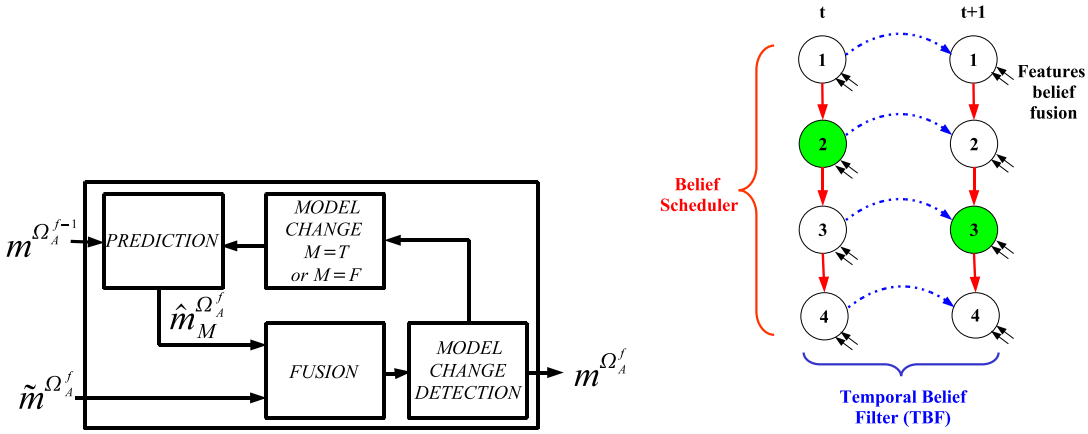
6-TCF output: if current conflict ϵ_f is low, then the output is the fusion result of prediction and observations. It conflict is too high, then we keep the prediction (cautious approach). Formally: $m^{\Omega_A^f} = \hat{m}_{\mathcal{M}}^{\Omega_A^f} \textcircled{\cap} \tilde{m}^{\Omega_A^f}$ if $\epsilon_f \leq \delta_\emptyset$ and $\hat{m}_{\mathcal{M}}^{\Omega_A^f}$ otherwise where $\delta_\emptyset$ is a threshold reflecting a tolerance to the conflict adaptively computed using the mean of conflict over a window (size $N = 5$) of a few frames: $\delta_\emptyset = 1/N \cdot \sum_{f_i=(f-N-1)}^{f} \epsilon_{f_i}$.

In order to remain coherent with the model of evolution that is used, we modify the belief mass as follows: if the model used is $\mathcal{T}$ then the belief on the emptyset ($m^{\Omega_A^f}(\emptyset_A^f)$) and the belief on F_A^f ($m^{\Omega_A^f}(F_A^f)$) are transfered

onto T_A^f and Ω_A^f respectively, i.e. when the model is "$\mathcal{T}$: the state is true", $m^{\Omega_A^f}(T_A^f) \leftarrow m^{\Omega_A^f}(T_A^f) + m^{\Omega_A^f}(\emptyset_A^f)$, $m^{\Omega_A^f}(\Omega_A^f) \leftarrow m^{\Omega_A^f}(\Omega_A^f) + m^{\Omega_A^f}(F_A^f)$, $m^{\Omega_A^f}(\emptyset_A^f) = m^{\Omega_A^f}(F_A^f) = 0$. When the model is "$\mathcal{F}$: the state is false", $m^{\Omega_A^f}(F_A^f) \leftarrow m^{\Omega_A^f}(F_A^f) + m^{\Omega_A^f}(\emptyset_A^f)$, $m^{\Omega_A^f}(\Omega_A^f) \leftarrow m^{\Omega_A^f}(\Omega_A^f) + m^{\Omega_A^f}(T_A^f)$, $m^{\Omega_A^f}(\emptyset_A^f) = m^{\Omega_A^f}(T_A^f) = 0$. These redistributions allow to decrease conflict between successive frames.

7-Local Quality criterion: given a model of evolution ($\mathcal{M}$), we compute the quantity: $LQ_{i,j}^{f_s:f}[\mathcal{M}](T_A^f) = (1 - \frac{1}{f-f_s}) \times LQ_{i,j}^{f_s:(f-1)}[\mathcal{M}](T_A^f) + (1 - \epsilon_f) \cdot m^{\Omega_A^f}(T_A^f)/(f - f_s)$ for each action A_i within each activity S_j. This criterion is computed on-line and embeds past events and innovation. It uses conflict to weigh the current belief on T_A^f. This criterion is said "local" because concerns only one action within a sequence. It reflects how we can be confident in an action.

8-Transition and false alarm detection: when the stop threshold is reached for an action A_i in sequence S_j, we compare its Local Quality criterion $LQ_{i,j}^{f_s:f}[\mathcal{M}](T_A^f)$ with a threshold δ_{FA} which is the minimal quality value required to validate a model change. If the criterion is lower, we declare the model change. When a false alarm occurs with the model $\mathcal{T}$ on a given interval of frames, then the TCF is run again on this interval but the model is compelled to be false $\mathcal{F}$ (it does not take into account the stop CUSUM threshold on this interval).



(a) TCF principle.

(b) Action filtering using TCF and sequencing using Belief Scheduler.

Figure 9: Sequential data recognition using the proposed TBM based approach.

When the TCF is applied on several actions separately, we obtain smooth belief on actions. In this case, action state is called natural. In order to recognize activity, we need to link actions and for that we use the notion of constrained state: the state of on action depends on the state of the other actions. That is the purpose of the next section.

3.4 Belief Scheduler for action sequencing

We have proposed a state machine, called belief scheduler of states [23], based on TBM and relying on the TCF. The scheduler allows to recognize a sequence of actions in the TBM context (Fig. 9(b)) and ensures that, at each frame of the video, one and only one action is in the *right state* while the others are in the *false state*.

We assume a sequence $S_n = \{A_1^n \rightarrow A_2^n \rightarrow \cdots \rightarrow A_k^n \rightarrow \cdots \rightarrow A_K^n\}$ made of K actions. The sequences evolve from an action A_k^n to the following one A_{k+1}^n if the TCF indicates that A_k^n becomes *false* or if A_{k+1}^n becomes *right*. As presented in the previous section, this information is provided by the CUSUM:

– A_k^n and A_{k+1}^n are false: if the CUSUM of A_k^n is greater than the stop threshold and if the quality LQ of

this action is high, then the following action becomes true. The model of A_k^n becomes naturally false while the model of A_{k+1}^n is compelled to be true. We call this process a forcing [23].

– A_k^n and A_{k+1}^n are true: if the CUSUM of A_{k+1}^n is greater than the stop threshold and if the action quality LQ is high, then action A_{k+1}^n becomes true. The model of A_{k+1}^n is naturally true while the model of A_k^n is compelled to be false. This is the preemption process [23].

Some illustrations are depicted in Figures 10 and 11 for a high jump sequence. Figure 10 demonstrates the capability of the both TCF and Belief Scheduler to smooth belief on actions. Figure 11 depicts conflict and CUSUM in a high jump sequence. Conflict is generally high during transitions.

3.5 Action and activity inference

In order to infer which action and which activity are true at a given time, we use the Local Quality recognition performance for action (LQ) and we compute a Global one (GQ) for activity.

When the sequence S_n evolves from action A_k^n to action A_{k+1}^n, a criterion ($\mathbf{LQ}_k^n$) is computed for A_k^n without reference (see previous Section). When a sequence is covered totally, K values of $\mathbf{LQ}_k^n$ are available. A criterion $\mathbf{GQ}^n$ is computed by aggregating the local ones: $\mathbf{GQ}^n = \sum_{k=1}^{K} \mathbf{LQ}_k^n / K$. The sequence S_n better corresponds to the data than S_p if $\mathbf{GQ}^n > \mathbf{GQ}^p$ and if $\mathbf{GQ}^n$ is greater than a given required value (e.g. 50%). In a recognition process with four activities (high jump, pole vaults, triple jumps and long jumps for instance), we just need to compute the Global Quality for each activity given a set of observations (features) and then to choose the one that maximizes the global criterion.

The proposed criterion has the same role as likelihood in the probability context but this criterion has the strong advantage to be understandable. It can be thresholded in order to create a new class of action or a new class of activity (class of rejects) since it is bounded. Class of rejects can not really be obtained using usual log-likelihood in probability context except by using log-likelihoods ratio but this is not well justified in case of several online and competing actions and activities recognizers.

The proposed inference method is illustrated in Figure 11(b). This figure shows evolution of Local Quality recognition performance for the two possible action states: action is right with $LQ_{i,hj}^{fs:f}[\mathcal{T}](R_A)$ (left-side) and action is false with $LQ_{i,hj}^{fs:f}[\mathcal{F}](F_A)$ (right-side). For instance, actions quality given the model is true (left-side) are around 100%, 80%, 70% and 95% for running, jumping, falling and standing up respectively. So the Global quality is of about 86% (using the mean). This value reflects the confidence of the system in activity high jump.

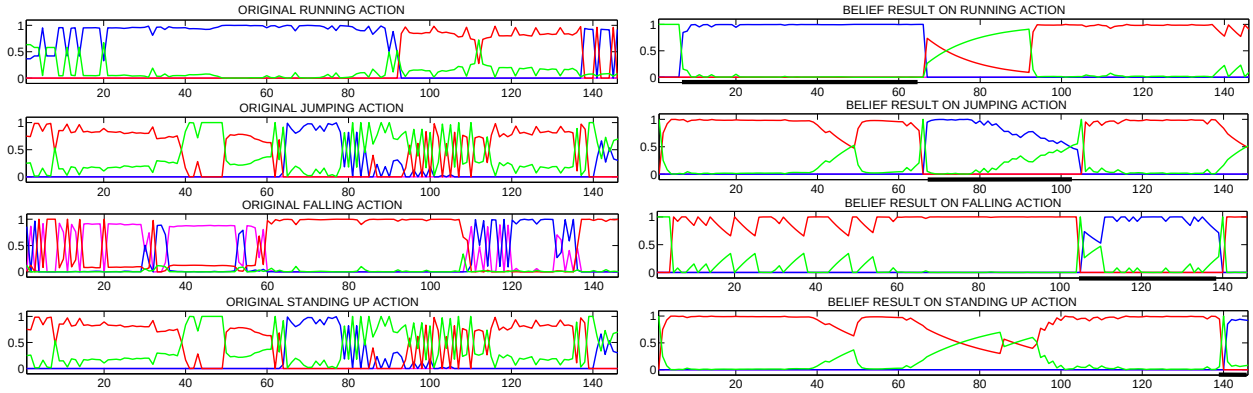
3.6 Coarse to fine approach and feedback

The action sequence method consists of two steps: a coarse detection and a fine detection of actions. The coarse step involves the camera motion features and the center of mass. In the fine step, sequencing is based on other "specialized" features such as Θ_1 in order to discriminate all actions.

3.6.1 Coarse step

The sequences to be recognized represent four types of jump: high jump (S_{hj}), pole vault (S_{pv}), triple jump (S_{tj}) and long jump (S_{lj}). Sequences $S_n, \forall n \in \{hj, pv, lj\}$ are firstly described by a *coarse* action sequence: $S_n = \{R_n \rightarrow J_n \rightarrow F_n \rightarrow U_n\}$, where R_n is the running action, J_n is jumping, F_n is falling and U_n is standing up in sequence S_n. For triple jump, the coarse sequence is: $S_{tj} = \{R_{tj} \rightarrow J_{tj} \rightarrow F_{tj} \rightarrow J_{tj} \rightarrow F_{tj} \rightarrow J_{tj} \rightarrow F_{tj} \rightarrow U_{tj}\}$. There is no subsequence for triple jump because the coarse one is characteristic and can not be confused with the other types of jump.

All actions $\{R_n, J_n, F_n, U_n\}, \forall n \in \{hj, pv, lj, tj\}$ are detected by a fusion process performed at each frame of the video as described in Section 3.2.



(a) Before filtering and scheduling.

(b) After filtering and scheduling.

Figure 10: Variation of belief in each action of a *high jump* sequence before (a) and after (b) filtering and scheduling. The activity *high jump* is found using only the belief on *true* (blue). Legend: states true (T_A, in blue), false (F_A, false), ignorance ($T_A \cup F_A$, in green) and conflict ($\emptyset$, in magenta).

The coarse definition of a sequence provides the intervals of frame where an action is potentially true but does not allow to distinguish the type of sequence. In order to differentiate the sequences, a fine analysis is required.

3.6.2 Fine step

When a jumping action is coarsely detected (using vertical variation) by a coarse sequence, the analysis of the angle is performed within the interval of frames where the jumping was detected. We can call it a subsequence. This process allows for instance to discriminate between a jumping action in a pole vault or in a high jump.

The fine analysis is thus performed in the intervals of frames detected by the coarse process by exploiting feature Θ_1. The numerical-to-symbolic conversion [17] of Θ_1 is performed by dividing the interval of possible values $[-180^o, 180^o]$ into 4 main positions $\{N, S, W, E\}$ (North, South, West, East) and 4 intermediate positions $\{NW, SW, SE, NE\}$. The conversion is depicted in Figure 12 and shows the explicit modelling of the doubt between two positions, for instance $SW \cup W$. The fuzzy description of the angle value allows to take imprecision and uncertainty of this feature into account. Notably, each position is modelled by a trapezoidal fuzzy set with a size support of 40^o.

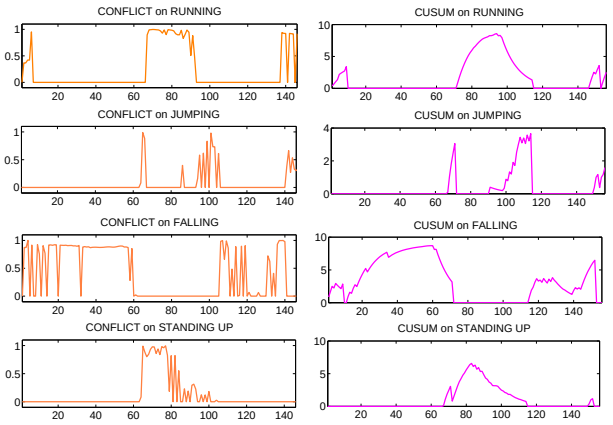
The sequencing of the angle value is performed according to each action sequence. One set of sequences is necessary for both right-to-left and left-to-right translations of the camera. In Table 2, only the right-to-left case is described. In Figure 13(b), the high jump action sequence is pictorially described.

When a fine sequence is recognized, a criterion LQ is computed for all actions within the sequence. These criteria are then aggregated (as described in the previous Section) in order to compute the global quality criterion GQ of the whole sequence including subsequences (coarse and fine).

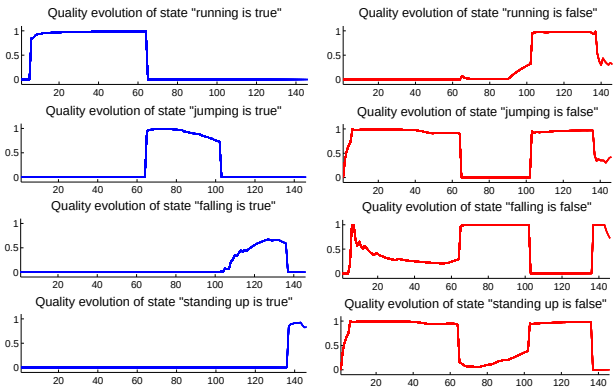
3.6.3 Correction of tracking using activity recognition

A feedback is a powerful means to adapt a processing chain to varying conditions. In this paper, we propose a solution to detect inversion of points in tracking that is based on *error sequence*: we assume that we know some sequences that correspond to inversion.

An example is provided in Figures 13-14 for a high jump where inversion often occurs at the end of the sequence. In these figures, the angle shows an inversion of the human points provided by the tracking due



(a) Conflict and CUSUM evolution corresponding to the case of Figures 10(a)-10(b).



(b) Local Quality recognition performance for each action in the sequence and for each action state (true and false).

Figure 11: Activity recognition: belief evolution, cumulative sum of conflict within TCF and Local Quality recognition performance for each action and in the sequence and given each model of evolution.

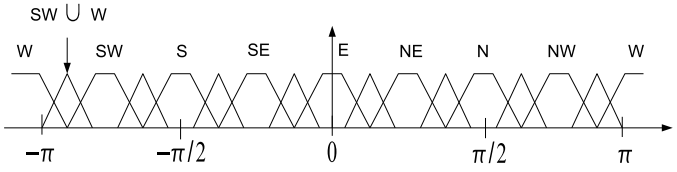


Figure 12: Numeric-to-symbolic conversion of Θ_1.

to very bad segmentation when the athlete falls on the air mattress (top foot, down head). This error can be detected by means of action sequencing (Figure 13(b)). For that, we denote I_{hj} the symbol (of activity) associated to the inversion in a high jump. Coarsely, the inversion (described previously) is searched after a falling action. Finely, the sequence used to detect this error is $I_{hj}^{\Theta_1} = \{S \rightarrow SE \rightarrow E \rightarrow SE \rightarrow E\}$. This sequence is depicted in Figure 13(a).

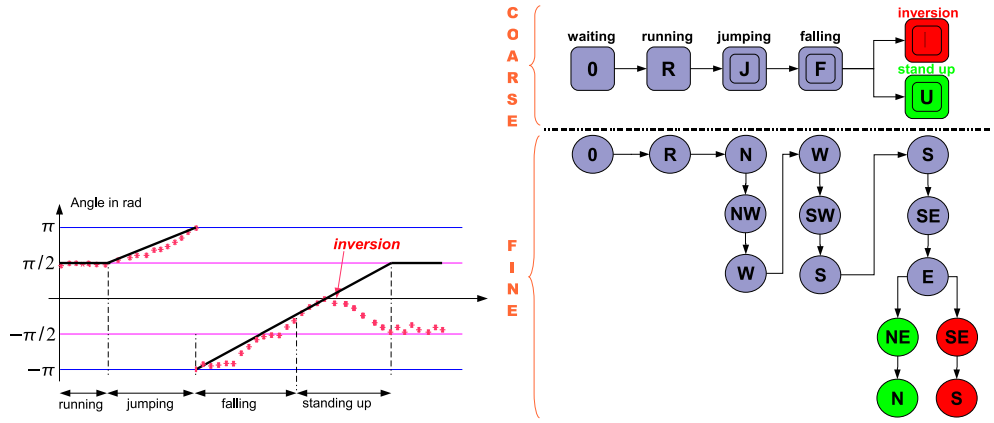
When the error sequence is of high quality, i.e. its quality $\mathbf{GQ}$ is high, then an error is assumed to be detected (the correct sequence, even "of error", is detected). In this case, a feedback process is performed onto the tracking algorithm in order to correct the inversion (Figure 13(b)).

4 Experimental Results

In this section, we present experimental results on human detection, tracking and action/activity recognition. Algorithms have been implemented using C and Matlab.

We have developed a dataset of 68 videos, in order to test the proposed scheme. The database is characterized by its heterogeneity with a high variation of view angles as well as unconstrained indoor or outdoor environments (other moving people can appear), and athletes (male, female with different skills, skin colors). The most of the videos are in low quality (having resolution 352 x 288) captured from broadcast TV.

Some results of the proposed framework are available at the Web addresses: `www.lis.inpg.fr/pages_perso/ramasso/index.htm` and `www.csd.uoc.gr/~cpanag/DEMOS/actionActivityRecognition.`



(a) (b)

Figure 13: **(a)** Theoretical angle rough evolution (full line) and observed one (dotted-line). **(b)** Action sequence by a coarse to fine approach for high jump based on angle Θ_1.

(a) (b) (c)

Figure 14: **(a)** Original image from high jump sequence. **(b)** Result of points tracking before correction. **(c)** Result of points tracking after correction.

`htm`.

In many cases, the low quality silhouettes increase the errors of major human points computation (about $10-15\%$ of human height) mainly on leg points. These low tracking accuracy results suffice for action and activity recognition.

The database contains 68 videos with four types of jumps: high jump (hj), pole vault (pv), triple jump (tj) and long jump (lj). Each video is analyzed by the four sequences S_n, $\forall n \in \{hj, pv, lj, tj\}$ providing four criteria $\mathbf{GQ}^n$. A jump n^* is associated to the current video if $n^* = \max_n \mathbf{GQ}^n$ and if $\mathbf{GQ}^{n^*}$ is greater than 50%. One setting per type of jump is provided for the TCF. Then, the obtained results are compared with the manually annotated video to compute a precision index. Using the coarse sequencing, all actions are well detected. However, to discriminate actions, we use the refinement described in Section 3.6.2 and based on the angle.

The *classification rates* are:

$$\mathbf{C}_{hj} = 87\%\ (13/15)\ ; \mathbf{C}_{pv} = 85\%\ (22/26)\ \ \mathbf{C}_{tj} = 75\%\ (9/12)\ \ \mathbf{C}_{lj} = 74\%\ (11/15)$$

(for high jumps, pole vaults, triple jumps and long jumps respectively). For high jumps, two videos have been confused with pole vault due to errors in body rotation during the jumping and falling steps. For pole vaults,

Table 2: Sequences of the angle for each type of jump.

sequence name	symbol and action sequence expression
pole vault	$S_{pv} = \{R_{pv} \rightarrow J_{pv} \rightarrow F_{pv} \rightarrow U_{pv}\}$
running	$R_{pv} = \{N \cup (\varepsilon \ is \ high)\}$
jumping	$J_{pv} = \{N \rightarrow NE \rightarrow E \rightarrow SE \rightarrow S \rightarrow SE \rightarrow E\}$
falling	$F_{pv} = \{E \rightarrow NE \rightarrow N \rightarrow NW \rightarrow W\}$
standing up	$U_{pv} = \{W \rightarrow NW \rightarrow N\}$
high jump	$S_{hj} = \{R_{hj} \rightarrow J_{hj} \rightarrow F_{hj} \rightarrow U_{hj}\}$
running	$R_{hj} = \{N\}$
jumping	$J_{hj} = \{N \rightarrow NW \rightarrow W\}$
falling	$F_{hj} = \{W \rightarrow SW \rightarrow S\}$
standing up	$U_{hj} = \{S \rightarrow SE \rightarrow E \rightarrow NE \rightarrow N\}$
long jump	$S_{lj} = \{R_{lj} \rightarrow J_{lj} \rightarrow F_{lj} \rightarrow U_{lj}\}$
running	$R_{lj} = \{N\}$
jumping	$J_{lj} = \{N\}$
falling	$F_{lj} = \{N \rightarrow NE \rightarrow E\}$
standing up	$U_{lj} = \{E \rightarrow NE \rightarrow N\}$

four videos have been confused with high jumps still due to errors in body rotation during the jumping and falling steps. For long jumps, confusions with pole vaults (2 cases) and high jumps (2 cases) have occured due to the athletes' arms movements that have disturbed the tracking and simulated rotation. Lastly, 3 triple jumps confusions with long jumps have occured due to lack of texture in the videos that has disturbed the camera motion estimation and has hidden the two first jumps (the third jump is the one with the highest amplitude).

Concerning inversion of the tracked points, we have tested high jumps and pole vaults: 8 videos with inversion were tested and the detection rate was $\mathbf{C}_{inv-hj} = 75\% \ (6/8)$.

Error rates in classification concerns the videos with 1) pure divergence (zoom) with athlete in front of the camera preventing from using the angle, 2) bad pole deletion, 3) video shot changes and 4) bad camera motion estimation in too low quality videos.

5 Conclusion

The proposed human motion analysis framework based on Transferable Belief Model (TBM) has demonstrated good performance on athletes actions and activities recognition. The TBM allows to represent doubt and conflict which can not represented in usual probability theory. These notions are fully exploited in this paper in both the Temporal Credal Filter which smooth belief on actions and in the Belief State Scheduler which recognized activities as sequence of understandable actions. The Belief State Scheduler has been exploited for hierarchical recognition of actions and activities in order to simplify their recognition. It has also been exploited for error sequence recognition in order to detect inversion of points during tracking therefore enabling one to perform feedback from high level to low level modules.

Algorithms for action and activities process features provided by robust extractors related to shape and motion of athletes. Videos are assumed to be acquired by moving camera. Results show good performance in the recognition of running, jumping, falling and standing up actions as well as athletics jumps that are pole vault, high jump, long jump and triple jump.

An extension of the proposed methodology include the addition of more sports and actions.

Acknowledgments

This work is partially supported by SIMILAR European Network of Excellence.

References

[1] J. Aggarwal and S. Park, "Human motion: Modeling and recognition of actions and interactions," in *3DPVT04*, 2004, pp. 640–647.

[2] C. Panagiotakis, E. Ramasso, G. Tziritas, M. Rombaut, and D. Pellerin, "Shape-motion based athlete tracking for multilevel action recognition," in *Proc. of AMDO*, 2006, pp. 385–394.

[3] K.M. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette across time: Part ii: Applications to human modeling and markerless motion tracking," *Int. Journal of Computer Vision*, vol. 63, no. 3, pp. 225–245, 2005.

[4] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density for visual surveillance," in *Proc. of the IEEE*, 2002, vol. 90, pp. 1151–1163.

[5] J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. of Vis. Comm. and Image R.*, vol. 6, no. 4, pp. 348–365, 1995.

[6] I. Grinias and G. Tziritas, "Robust pan, tilt and zoom estimation," *Int. Conf. on Digital Signal Processing*, 2002.

[7] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[8] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *PR*, vol. 36, no. 3, pp. 585–601, 2003.

[9] Pascual J. Figueroa, Neucimar J. Leite, and Ricardo M. L. Barros, "Tracking soccer players aiming their kinematical motion analysis," *Computer Vision and Image Understanding: CVIU*, vol. 101, no. 2, pp. 122–135, 2006.

[10] Fang-Hsuan Cheng and Yu-Liang Chen, "Real time multiple objects tracking and identification based on discrete wavelet transform," *Pattern Recognition*, vol. 39, no. 6, pp. 1126–1139, 2006.

[11] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition and probabilistic recognition methods," *CVIU*, vol. 96, pp. 129–162, 2004.

[12] Y. Luo, T.D. Wu, and J.N. Hwang, "Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks," *CVIU*, vol. 92, pp. 196–216, 2003.

[13] M. Rombaut, I. Jarkass, and T. Denoeux, "State recognition in discrete dynamical systems using petri nets and evidence theory," in *ECSQARU*, June 1999.

[14] P. Smets and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, 1994.

[15] G. Shafer, *A mathematical theory of evidence*, Princeton University Press, Princeton, NJ, 1976.

[16] P. Smets, *Advances in the Dempster-Shafer Theory of Evidence - What is Dempster-Shafer's model ?*, pp. 5–34, Wiley, r.r. yager and m. fedrizzi and j. kacprzyk edition, 1994.

[17] E. Ramasso, C. Panagiotakis, M. Rombaut, and D. Pellerin, "Human action recognition in videos based on the transferable belief model - application to athletics jumps,," *Pattern Analysis and Applications Journal*, 2007, Accepted, doi:10.1007/s10044-007-0073-y.

[18] V. Girondel, A. Caplier, L. Bonnaud, and M. Rombaut, "Belief theory-based classifiers comparison for static human body postures recognition in video," *Int. Jour. of Signal Processing*, vol. 2, no. 1, pp. 29–33, 2005.

[19] Z. Hammal, A. Caplier, and M. Rombaut, "Belief theory applied to facial expressions classification," *Int. Conf. on Advances in Pattern Recognition*, 2005.

[20] T.B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *CVIU*, vol. 81, pp. 231–268, 2001.

[21] T.B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, pp. 90–126, 2006.

[22] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *IEEE Computer Vision and Pattern Recognition*, Nice, France, 2001, vol. 2, pp. 123–130.

[23] E. Ramasso, D. Pellerin, and M. Rombaut, "Belief Scheduling for the recognition of human action sequence," in *Proc. of the 9th Int. Conf. on Information Fusion*, Florence, Italia, July 2006.

[24] C. Panagiotakis, E. Ramasso, G. Tziritas, M. Rombaut, and D. Pellerin, "Automatic people detection and counting for athletic videos classification," in *IEEE Int. Conf. on Advanced Video and Signal based Surveillance, special session on Vision-based gesture and human action recognition, by I. Patras and E. Hancock*, London, United Kingdom, 2007, Accepted.

[25] V. Girondel, A. Caplier, L. Bonnaud, and M. Rombaut, "Belief theory-based classifiers comparison for static human body postures recognition in video," *Int. Jour. of Signal Processing*, vol. 2, no. 1, pp. 29–33, 2005.

[26] Z. Hammal, A. Caplier, and M. Rombaut, "Belief theory applied to facial expressions classification," in *Int. Conf. on Advances in Pattern Recognition*, Bath, United Kingdom, 2005.

[27] P. Smets, "Beliefs functions: the disjunctive rule of combination and the generalized bayesian theorem," *IJAR*, vol. 9, pp. 1–35, 1993.

[28] E. Ramasso, M. Rombaut, and D. Pellerin, "State filtering and change detection using TBM conflict - application to human action recognition in athletics videos,," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 7, 2007, Accepted for publication.

[29] M. Rombaut and Y.M. Zhu, "Study of Dempster-Shafer theory for image segmentation applications," *Image and Vision Computing*, vol. 20, no. 1, pp. 15–23, 2002.

[30] Z. Elouedi, K. Mellouli, and Ph. Smets, "Assessing sensor reliability for multisensor data fusion within the transferable belief model," *IEEE Trans. SMC*, vol. 34, no. 1, pp. 782–787, 2004.

[31] E. Ramasso, *State sequence recognition using Transferable Belief Model and application to sports video analysis*, Ph.D. thesis, University of Joseph Fourier - Grenoble, 5 December 2007.