



HAL
open science

Belief scheduling for recognition of human action sequence

Emmanuel Ramasso, Denis Pellerin, Michèle Rombaut

► **To cite this version:**

Emmanuel Ramasso, Denis Pellerin, Michèle Rombaut. Belief scheduling for recognition of human action sequence. 9th International Conference on Information Fusion, ICIF'06, IEEE/ISIF, 2006, Florence, Italy. hal-00068009

HAL Id: hal-00068009

<https://hal.science/hal-00068009>

Submitted on 10 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Belief scheduling for recognition of human action sequence

E. Ramasso, D. Pellerin, M. Rombaut
Laboratoire des Images et des Signaux (LIS, UMR 5083)
46, avenue Félix Viallet, 38031 Grenoble, France.
first_name.family_name@lis.inpg.fr

Abstract - *In the context of human action sequence recognition in video sequences, a scheduler of actions is proposed. The Belief Scheduler is based on a Temporal Belief Filter ensuring a consistency in the temporal belief evolution as well as temporal constraints. The Belief Scheduler is inspired from System Science and is used to recognize actions in human activity defined as a sequence of actions. This new scheduler is proposed for applications in Computer Vision. The whole system of action and sequence recognition is based on the Transferable Belief Model proposed and developed by Ph. Smets which allows to explicitly model the doubt between actions. Furthermore, it emphasizes the conflict which is exploited for state change in action sequence. The experiments concern real video sequences with moving camera and several unknown view angles. The Temporal Belief Filter and the Belief Scheduler show their efficiency to cope with varying video quality and experiment conditions.*

Keywords: human motion analysis, sequence analysis, constrained action recognition, scheduling, Transferable Belief Model.

1 Introduction

Human motion analysis is an important topic of interest in Computer Vision and Video Processing [1]. Research in these domains is motivated by the diversity of applications such as video indexing and retrieval, automatic surveillance and human-computer interaction. The main problem is to link the real world, which has intrinsically an analogical nature, to the human world which is symbolic [2].

Many methods have been proposed for human action sequence recognition [1] generally based on the *Bayesian framework* [3] with *Hidden Markov Models* (HMM) and *Dynamic Bayesian Networks* (DBN) [4].

In [5], an architecture was proposed for human action recognition using the *Transferable Belief Model* (TBM) [6]. The TBM is well-suited for action recognition notably because (i) *doubtful* transitions between actions are explicitly modelled, (ii) *conflict* between parameters reflects the need to improve the fusion process and (iii) *reliability* of parameters depends on the context and can be included in the system. In [7],

a Temporal Belief Filter (TBF) was described for improving action recognition in the context of real video sequences with varying experiment conditions.

In this paper, a *Belief Scheduler* based on the TBM is proposed for human action sequence recognition. The *Belief Scheduler* exploits and embeds the TBF as well as scheduling principles [8]. Temporal sequences recognition is achieved by the scheduler by means of criteria based on conflict as defined in the TBM. In order to illustrate the proposed approach, athletics meeting videos acquired with a moving camera under an unknown view angle are analyzed to detect and recognize action sequences performed by one athlete.

The organization of the paper is as follows. An overview of the action recognition architecture based on the TBM is presented in Section 2. The main characteristics of the TBF are recalled in Section 3 to string with the *Belief Scheduler* described in Section 4. Section 5 deals with experimental results. Finally, we conclude and propose future work.

2 Basic belief on actions obtained by fusion

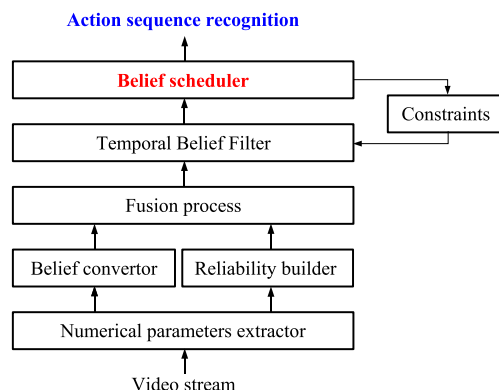


Figure 1: Architecture for human action sequence recognition in videos. The architecture is based on the Transferable Belief Model.

The whole system is presented in figure 1. The video stream is analyzed by means of image processing. At each frame of the video, a belief is computed for each numerical parameter extracted from the video. These

beliefs are then combined in the TBM framework. A detailed description of this part is given in [5] and the main points will be recalled in the sequel.

Some reliability factors are automatically determined at each frame of the video and inserted into the fusion process. The beliefs are then analyzed by the TBF described in [7] which provides smooth beliefs without conflict and with emphasized transitions. The TBF works on each action independently providing *natural state* of the actions. Causal relations and temporal constraints concerning actions are then added and managed in a *scheduler* of actions. As a result, belief on actions with *constrained states* are obtained. The *Belief Scheduler* is the main contribution of this paper.

2.1 Numerical parameters

Relevant numerical parameters are extracted at each frame from the video stream. Parameters are generally application dependant. In this paper, they are based on the following two main assumptions: first, *the human is tracked by the cameraman* because he is the center of interest and, second, *the trajectories of human's head, center of gravity and one end of leg give information on actions*.

The chosen parameters are (1) the camera motion parameters estimated from two successive frames and which are pan, tilt and zoom, and (2) the three human major points detected and tracked [5, 9]. Points coordinates are analytically combined to obtain more advanced parameters which are: the angle made by the human main axis and the horizon, the variation of the center of mass and the legs alternation.

2.2 Transferable Belief Model fusion

The numerical parameters values are converted into belief concerning the trueness of actions. A belief on actions is generated at each frame for each parameter. Belief of several parameters are then combined in the axiomatically well-founded Transferable Belief Model framework proposed by Smets and Kennes [6] to obtain a belief which takes all parameters into account.

2.2.1 From numerical parameters to belief on actions

An action A is described by two hypotheses 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”). In the sequel, an hypothesis concerning an action is called a state, e.g. “the current state of A is R_A ”.

In this part, the modelling is based on the assumption that *actions are not exclusive*, i.e. they can take place at the same time. They are also *independent*, i.e. the belief on one action does not imply anything about the belief of the others. Links between actions will be added by the *Belief Scheduler* described in Section 4.

The goal of the fusion process is to obtain the belief of each action A according to several numerical

parameters. A basic belief assignment (BBA) on an action A according to a parameter P is defined on the set of propositions $2^{\Omega_A} = \{\emptyset, R_A, F_A, R_A \cup F_A\}$ (where $R_A \cup F_A$ is the explicit doubt between hypotheses R_A and 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 $\sum_{X \subseteq \Omega_A} m_P^{\Omega_A}(X) = 1$. A value $m_P^{\Omega_A}(X)$ is a basic belief mass which expresses a confidence in proposition $X \subseteq \Omega_A$ according to parameter P but does not imply any additional claims regarding subsets of X . It is the fundamental difference with probability theory. A fuzzy-set inspired method is used to convert each numerical parameter into sources of belief.

2.2.2 Fusion process

Rules of combination are then applied to obtain a belief which takes the belief of all parameters into account. The fusion process is performed frame by frame for each action independently. Given two distinct BBAs $m_{P_1}^{\Omega_A}$ and $m_{P_2}^{\Omega_A}$ defined on the same FoD Ω_A then their combination is defined as:

$$m_{P_1}^{\Omega_A} \otimes m_{P_2}^{\Omega_A}(E) = \sum_{C \Delta D = E} m_{P_1}^{\Omega_A}(C) \cdot m_{P_2}^{\Omega_A}(D) \quad (1)$$

with $\Delta = \cap$ (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 parameters states. These logical rules are then translated into belief combinations where the logical AND is replaced by the \otimes -rule and the logical OR by the \oplus -rule assuming the same FoD [10].

2.2.3 Integrating reliability

The TBM offers a discounting process that allows to weight the belief of a parameter according to the reliability of the corresponding source. The reliability is an important tool for action recognition in video because it gives a penalty on belief provided by sources that work in non-optimal conditions. Given $\alpha_P \in [0, 1]$ the reliability factor of a parameter P then $1 - \alpha_P$ is called the discount rate and the belief provided by the parameter is revised as:

$$\begin{aligned} m_P^{\Omega_P, \alpha_P}(A) &= \alpha_P \times m_P^{\Omega_P}, & \forall A \subsetneq \Omega_P \\ m_P^{\Omega_P, \alpha_P}(\Omega_P) &= (1 - \alpha_P) + \alpha_P \times m_P^{\Omega_P} \end{aligned} \quad (2)$$

Statistics can be used to compute the reliability factor [11]. In this work, 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. Two reliability factors are considered. The first one is based on the quantity of pixels belonging to the dominant motion in an image and gives an opinion concerning the quality of the camera motion estimation. The second one provides an opinion for the quality of the tracking algorithm and is computed as the normalized ratio between the distance “leg-center

of gravity” and ”head-center of gravity” which, physically, has to be close to 1.

3 Natural state estimation

The Temporal Belief Filter (TBF) was proposed in a previous work described in [7]. The TBF works on each action independently taking as input the BBA obtained after parameters fusion and provides a BBA with the following properties:

- *Consistency between parameters*: the conjunctive rule of combination used in the TBM fusion process may emphasize a conflict between parameters which has to be solved,
- *Temporal consistency*: the belief on action can not vary abruptly between two successive frames,
- *Exclusivity between action states*: only one hypothesis concerning action (either R_A or F_A) gets a no null belief at each frame.

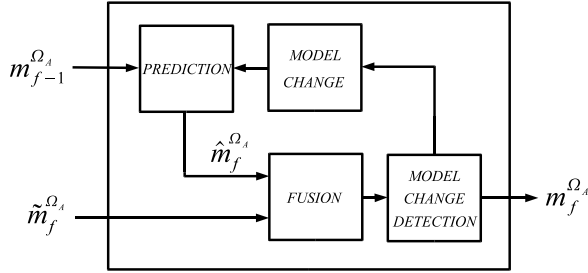


Figure 2: The Temporal Belief Filter (TBF) principle where $\hat{m}_f^{\Omega_A}$ is the prediction, $m_f^{\Omega_A}$ is the output value of the TBF at frame f given the natural state of actions and $\tilde{m}_f^{\Omega_A}$ is the measure provided by the fusion of the parameters at frame f .

One important characteristic of the TBF is that it provides a BBA with only two focal sets: either R_A and $R_A \cup F_A$, or F_A and $R_A \cup F_A$. In the former case, the action is said to be in the *right state* while *false state* in the latter case. Both states are said to be *natural* because they are computed by the TBF for each action *independently* according to the BBA obtained after parameters fusion. These states will be called *constrained* if temporal constraints between actions are taken into account for the computation of an action state. In the sequel, the TBF is briefly described. Then, in Section 4, a method is proposed to add temporal constraints between actions.

The general principle of the TBF is depicted in Figure 2. The core of the TBF is based on *implication rules* which are well managed in the TBM framework. Details on their formalization as well as an application to target identification are described by Ristic and Smets in [10]. An implication rule is generally used to specialize a BBA. We have interpreted implication rules \mathcal{R} and \mathcal{F} as models of evolution denoted $\mathcal{M} \in \{\mathcal{R}, \mathcal{F}\}$. Each one focuses on one hypothesis of

the FoD of an action A which is either R_A or F_A states. At each frame f , the TBF works in two steps: (i) prediction, (ii) fusion and if necessary state change.

3.1 Prediction

The prediction step relies on the following assumption: if an action state is R_A (resp. F_A) at frame $(f - 1)$ then it would be partially R_A (resp. F_A) at frame f . This model of evolution \mathcal{R} (resp. \mathcal{F}) is weighted by a confidence value of $\gamma_{\mathcal{R}} \in [0, 1]$ (resp. $\gamma_{\mathcal{F}} \in [0, 1]$):

Model \mathcal{R} :

If R_A at $(f - 1)$ then R_A at f with belief of $\gamma_{\mathcal{R}}$

Model \mathcal{F} :

If F_A at $(f - 1)$ then F_A at f with belief of $\gamma_{\mathcal{F}}$

(3)

In the sequel, the following vector notation of a BBA defined on a FoD Ω_A is used:

$$m^{\Omega_A} = [m^{\Omega_A}(\emptyset) \quad m^{\Omega_A}(R_A) \quad m^{\Omega_A}(F_A) \quad m^{\Omega_A}(\Omega_A)]^T$$

A model of evolution can be interpreted as a BBA. For instance, for the model \mathcal{R} :

$$m_{\mathcal{R}}^{\Omega_A} = [0 \quad \gamma_{\mathcal{R}} \quad 0 \quad 1 - \gamma_{\mathcal{R}}]^T \quad (4)$$

The disjunctive rule of combination (Eq. 1) is then used to compute the prediction from the previous BBA and the model of evolution:

$$\hat{m}_{f,\mathcal{M}}^{\Omega_A} = m_{\mathcal{M}}^{\Omega_A} \odot m_{f-1}^{\Omega_A} \quad (5)$$

The \odot -rule never assigns more belief to an hypothesis than does the previous BBA. As a clue, the prediction w.r.t. \mathcal{R} (Eq. 6) is given by:

$$\hat{m}_{f,\mathcal{R}}^{\Omega_A} = \begin{bmatrix} 0 \\ \gamma_{\mathcal{R}} \times m_{f-1}^{\Omega_A}(R_A) \\ 0 \\ (1 - \gamma_{\mathcal{R}}) \times m_{f-1}^{\Omega_A}(R_A) + m_{f-1}^{\Omega_A}(\Omega_A) \end{bmatrix} \quad (6)$$

When $\gamma_{\mathcal{M}} = 1$, the prediction equals the previous BBA reflecting a total confidence in the current state of action A . When $\gamma_{\mathcal{M}} = 0$, the model expresses a total ignorance about the prediction of the current BBA on the action.

3.2 Natural state change

Prediction $\hat{m}_{f,\mathcal{M}}^{\Omega_A}$ and measure $\tilde{m}_f^{\Omega_A}$ represent two distinct pieces of information concerning the natural state of action A . The conjunctive combination (Eq. 1) of their associated BBA leads to a new BBA. If the sources are discordant, then a conflict is emphasized indicating a potential natural state change, i.e. the model might be changed. Thus, the conflict value ϵ_f (Eq. 7) seems relevant for model change requirement:

$$\epsilon_f = (\hat{m}_{f,\mathcal{M}}^{\Omega_A} \odot \tilde{m}_f^{\Omega_A})(\emptyset) \quad (7)$$

The conflict analysis is thus required to know whether the current model is no longer valid. The CUSUM process of the conflict is well adapted for solving

problems concerning *abrupt and short changes* or *gradual and long changes* in the conflict value because it allows to sum up conflict during time. Generally, the CUSUM is the cumulative sum during time of the error between a prediction and a measure. In the case concerned, the error is the conflict value.

The initial CUSUM process works as follows: when the CUSUM value becomes greater than a warning threshold \mathcal{T}_w then the frame is stored as f_w and the model is *kept as valid*. As soon as the CUSUM value becomes greater than a stop threshold \mathcal{T}_s (at frame f_s) then the model is *changed* and the new model is applied from f_s . When a conflict appears between prediction and measure, as it could be the case in interval $[f_w, f_s]$, it was chosen to *trust the model of evolution*. Thus, the prediction is kept instead of an erroneous measurement and it avoids propagating conflict which is absorptive by the \odot -rule:

$$m_f^{\Omega_A} = \begin{cases} \hat{m}_{f,\mathcal{M}}^{\Omega_A} \odot \tilde{m}_f^{\Omega_A} & \text{if } \epsilon_f = 0 \\ \hat{m}_{f,\mathcal{M}}^{\Omega_A} & \text{otherwise} \end{cases} \quad (8)$$

Equation (8) accounts for the fact that the BBA $m_{f-1}^{\Omega_A}$ can have only two focal sets (Eq. 6) depending on the current model \mathcal{M} . Furthermore, the output of the TBF is a BBA without conflict and with only one hypothesis whose belief is not null. The interest of the \odot -rule is emphasized when there is often conflict because it allows to obtain $m_{f \rightarrow \infty}^{\Omega_A}(\Omega_A) = 1$ which reflects total ignorance of the system.

To cope with low conflict during a long time, a *fading memory* process has been embedded which allows to forget gradually past event. The fading memory process requires a coefficient nicknamed *fader*, and denoted as λ , which works on the current CUSUM $\mathbf{CS}(f)$ as follows:

$$\mathbf{CS}(f) \leftarrow \mathbf{CS}(f-1) \times \lambda + \epsilon_f \quad (9)$$

The fader is here chosen as a constant and is applied at each frame.

The two models (\mathcal{R} and \mathcal{F}) are tuned once and one model is applied while it is valid. Otherwise, it is changed by the other. A natural model change is required for an action A_k when the stop threshold \mathcal{T}_w^k is reached by its CUSUM. If the model change is accepted and performed, then the interval of frames $\mathbf{I}_{\mathbf{T}} = [f_w, \min(f_s, f_w + \mathcal{W})]$ can be interpreted as an interval of transition between the two action states. The parameter \mathcal{W} limits the size of the transition. The *vacuous* BBA is assigned to the frames belonging to $\mathbf{I}_{\mathbf{T}}$ to well represent ignorance: $m_{\mathbf{I}_{\mathbf{T}}}^{\Omega_A}(\Omega_A) = 1$. After a model change, the new model is applied from the upper bound of the interval of transition $\mathbf{I}_{\mathbf{T}}$ and the CUSUM is reset.

Initialization – The TBF is an online process. In order to initialize the system, the model which

better fits the first data is determined. For that, the CUSUM process is applied on an interval of frames for all models and the chosen one minimizes the CUSUM.

3.3 Suggestions for parameters setting

First, it is necessary to set the parameters in a relevant order: the fader and the models together, then the stop threshold, the warning threshold and at last the window. Second, for a given fader, the value of \mathcal{T}_s can be estimated by $\hat{\mathcal{T}}_s$ if references concerning ground truth of actions are available i.e. the start frame f_{sref} and the end frame f_{eref} of these actions are known. For that, the filter has to be applied with a model of type \mathcal{F} (*false state*) with a stop threshold \mathcal{T}_s unreachable (close to infinity). Then, $\hat{\mathcal{T}}_s$ is estimated by $\hat{\mathcal{T}}_s = \mathbf{CS}(f_{sref})$. If the data do not contain too much conflict, the estimation should be optimal otherwise, the fader has to be increased.

4 Constrained state estimation

In the previous section, action filtering was performed *independently* for each action. In this section, temporal constraints between actions are considered by assuming that the real state of an action is constrained by the activity defined as an ordered sequence of K actions. The proposed algorithm for constrained state estimation relies on the TBF and is called *Belief Scheduler*.

4.1 Sequence analysis problem

In the context of temporal sequence analysis, the Directed Acyclic Graph (DAG) is one of the most common and used tool in Computer Vision for human action sequence recognition such as Dynamic Bayesian Networks [4]. Within the TBM framework, at least two solutions have been proposed in the literature. In [12], a *belief Petri net* was proposed but it is not enough robust for human action sequence recognition in case of noise data, with false alarms and delayed action. The Evidential Network (ENC) initially proposed by Smets and deeply studied by Xu and Smets in [13] represents another solution but conditional beliefs must be provided. ENCs are discussed and generalized in [14].

These tools have not been applied for human motion analysis yet. Recently, the Credal Expectation-Maximization algorithm (CrEM) [15] has been proposed and might be a powerful tool when coupled with a ENC.

The method proposed in this paper is inspired from the work of System Science community. More precisely, the main idea is based on *scheduling* as defined in the *divisible load* paradigm [8]. Scheduling is performed by an algorithm nicknamed *scheduler* which generally aims at *executing tasks as fast as possible* while *respecting specifications* given by the designer of the scheduler like the ordering of the tasks, their priority or the access control to resources. The scheduler

is the core of an operating system such as Linux [16] and allows to optimize the programs execution while ensuring good performance for the user.

4.2 Activity constraints

An activity is defined as an ordered sequence of K actions, $A_1, A_2, \dots, A_k, \dots, A_K$. Ideally, the actions are synchronized but in real videos, they can be either *overlapped* or *unconnected* such as represented in Figure 3. Three main reasons can be identified:

- it is physically impossible to determine crisp boundary between two consecutive actions,
- the thresholds setting are seldom perfect in the numeric-to-symbolic conversion or in the TBF.
- there is always *noise* on the numerical parameters leading to either overlapped or unconnected actions.

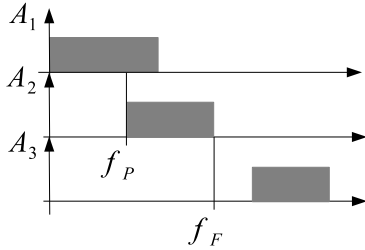


Figure 3: Representation of natural right state of actions using intervals (overlap between A_1 and A_2 ; gap between A_2 and A_3).

Overlapping and unconnection phenomenons are described as follows (where f is the current frame):

- overlapped: at frame f_P read as "the frame of preemption", two consecutive actions A_k and A_{k+1} have a natural right state at the same time.
- unconnected: at frame f_F read as "the frame of forcing", an action A_k becomes false while the next A_{k+1} is not yet right. Thus, two actions have a natural false state at the same time.

As it has been seen in [7] and recalled Section 3, a model \mathcal{R} or \mathcal{F} is associated to each action according to its state. Thus, the models $\{\mathcal{R}, \mathcal{F}\}$ can be interpreted as *resources* while actions represent *threads*¹ which try to access the resources. There are one resource of type $\{\mathcal{R}\}$ and $(K - 1)$ resources of type $\{\mathcal{F}\}$. The access control is managed by the *Belief Scheduler* according to the information provided by the TBF concerning the *model change* process.

The *Belief Scheduler* relies on the following assumption: at each frame of the video, one and only one action is in the *right state* while the others are in the *false state* despite of the overlapping and gap problems. Therefore, the *Belief Scheduler* modifies the state of each action according to the state of its neighboring actions.

¹Concept introduced in [3] for human motion analysis.

4.3 Belief Scheduler description

The *Belief Scheduler* has in charge to deal with the state change according to the data and the sequence. Its goal is to re-synchronize the actions by defining the constrained state for each of them. The proposed algorithm is based on the TBF and the diagram of the *Belief Scheduler* is depicted in Figure 4. One important characteristic of the *Belief Scheduler* is that it allows only one action to be true at a given time.

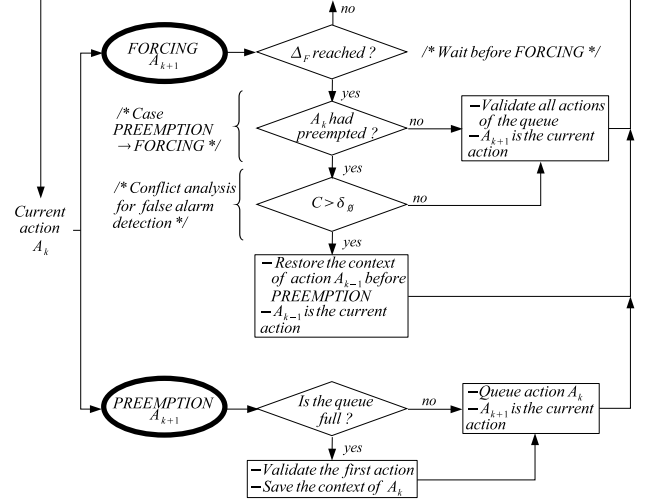


Figure 4: The *Belief Scheduler* of actions.

The algorithm considers two consecutive actions: the current one A_k and its following A_{k+1} . Thus, the order in the sequence is taken into account. All actions are compelled to be in the *false state* with the model \mathcal{F} except the current action A_k which is in the *right state* with the model \mathcal{R} . If the CUSUM of A_k or of A_{k+1} exceeds their respective stop threshold, then the *Belief Scheduler* is informed that the sequence is in a transition step. The transition between two actions in the sequence is achieved by two processes: the *PREEMPTION* and the *FORCING*, for managing *overlapped* and *unconnected* actions respectively.

4.3.1 Preemption process

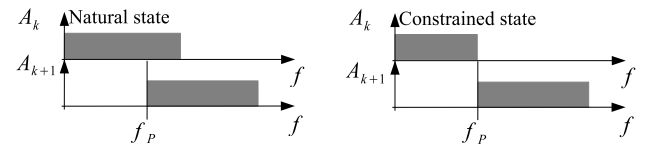


Figure 5: Preemption process.

This process concerns overlapped actions. At $(f_P - 1)$, natural state of A_k is right and natural state of A_{k+1} is false. At f_P , A_k is still right while A_{k+1} becomes right thus, two actions are right at the same time. In this case, constrained state of A_{k+1} is temporarily right (*right state*) from frame f_P and the constrained state of A_k is temporarily false (*false state*) until *validation* (see Figure 5). The validation is enabled when the quality of the recognition of the ac-

tion which asks for preemption, e.g. A_{k+1} , is good (section 4.3.3). Information at f_P concerning actions (cusum, belief,...), also called the *context*, is stored in case the PREEMPTION would not be enabled.

4.3.2 Forcing process

At $(f_F - 1)$, the natural state of A_k is right and the natural state of A_{k+1} is false. At f_F , A_k is false as well as A_{k+1} . Thus, the two actions are false at the same time. If the two successive actions are a little unconnected, i.e. with a small gap less than Δ_F , constrained state of A_k is forced to the right state until A_{k+1} becomes right. However, sometimes, the gap between successive actions can be large, i.e. with a size greater than a value Δ_F . The action requiring a FORCING, e.g. constrained state of A_k , keeps on being right until the frame $(f_F + \Delta_F)$. In this case, constrained state of A_{k+1} is forced to be right and constrained state of A_k becomes false (see Figure 6).

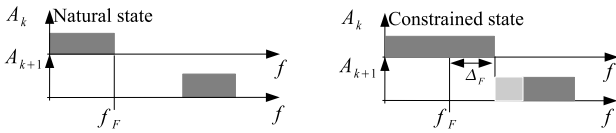


Figure 6: Forcing process.

In the case of FORCING, there is conflict between the natural and constrained states of actions A_k and A_{k+1} . Therefore, according to Equations (6) and (8) embedded in the TBF, the belief tends to ignorance. If the gap is too large, i.e. more than Δ_F , the constrained state of A_{k+1} is always in conflict with its natural state. Thus, the TBF naturally changes it into the false state.

4.3.3 False alarm detection

A false alarm appears when an action becomes true instead of staying false. It corresponds to the case that an action A_{k+1} tries to perform a FORCING on A_{k+2} while, beforehand, A_{k+1} had made a PREEMPTION on A_k . If actions A_{k+1} and A_{k+2} are too much unconnected, then A_{k+1} can be interpreted as a false alarm (see Figure 7). This false alarm procedure is applied to *validate* a PREEMPTION.

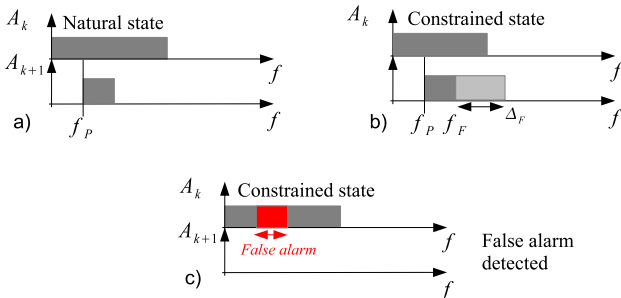


Figure 7: False alarm processing.

In order to decide whether action A_{k+1} is or not a false alarm, it is required to assess the *recognition performance* of this action. Thus, there is a need to find

a *quality criterion* in order to assess this performance. One difficulty is that the criterion has to be computed *without references*.

The chosen criterion is the *mean of conflict*, denoted \overline{C}_θ , computed along the interval $[f_P, f_F + \Delta_F]$. We denote $N_f = (f_F + \Delta_F - f_P + 1)$ the number of frames of the interval concerning the action A_{k+1} which is potentially a false detection and which has made a PREEMPTION after a FORCING (Fig. 7,b). In this interval, the constrained state of the action A_{k+1} is often in conflict with its natural state. The mean of conflict \overline{C}_θ is:

$$\overline{C}_\theta = \frac{1}{N_f} \left(\underbrace{\sum_{f=f_P}^{f_F} \epsilon_f(A_{k+1})}_{\text{natural conflict}} + \underbrace{\sum_{f=f_F+1}^{f_F+\Delta_F} \epsilon_f(A_{k+1})}_{\text{artificial conflict}} \right) \quad (10)$$

where $\epsilon_f(A_{k+1})$ is the conflict between measurement and model for action A_{k+1} and given by Equation (7). If $\overline{C}_\theta > \delta_\theta$, where δ_θ is a crisp threshold, the detection of the action is considered as a false detection. In consequence, the context of the actions at frame f_P , as for instance the value of the CUSUM, is restored and the previous action (before preemption), e.g. A_k , becomes right again. If $\overline{C}_\theta < \delta_\theta$ then A_{k+2} becomes right and A_{k+1} and A_{k+2} are both validated.

4.3.4 Elucidating the parameter Δ_F

To give a hand in the setting of the parameter Δ_F , some details are given concerning its interpretation. Its value can be bounded if it is possible to estimate the higher value D_M of the disjunction space between two states estimated or given by expert knowledge. In this case, $\Delta_F \geq D_M/2$. In addition, the value of Δ_F should influence the value of \overline{C}_θ by increasing artificially the conflict. Given δ_θ , let d_m be the minimal duration of an action estimated or given by expert knowledge. In this case, $\Delta_F/(\Delta_F + d_m) \leq \delta_\theta$. Therefore, if D_M and d_m are known then Δ_F can be bounded:

$$\frac{D_M}{2} \leq \Delta_F \leq \frac{\delta_\theta}{1 - \delta_\theta} \times d_m \quad (11)$$

Learning D_M and d_m is quite straightforward if references are given. Concerning δ_θ , its setting depends on the setting of the TBF. Its value was set empirically to 0.2 for all tests.

4.3.5 Queuing

When several actions perform consecutive PREEMPTION, they must be validated to ensure they are not false alarms. One solution consists in storing them in a FIFO queue to wait for their validation. As emphasized in the previous section, the validation will appear after a FORCING process. The number of actions in the queue is limited and when the queue is full, the first queued-action is automatically validated.

Initialization – It can be noticed that, at the beginning of the scheduling, all actions are in the *false* state. An artificial initial right state action is added in

the sequence that allows the *Belief Scheduler* to wait for a PREEMPTION of the first action.

5 Experiments

Database description: The proposed system is used to recognize *high jump* and *pole vault* actions sequence. Their sequence are *running*, *jumping* and *falling*.



Figure 8: Heterogeneous database used for testing.

The database is composed of 40 videos acquired with a moving camera and several unknown view angles. There are 25 pole vaults and 15 high jumps equivalent to about 6500 frames. The database is characterized by its heterogeneity (Fig. 8) with a panel of view angles as well as environments and athletes (out/indoor, male, female, other moving people).

Settings: To show the generalization capability of the method, the TBF parameters as well as the the *Belief Scheduler* ones are the same for all actions and all videos. The parameters are tuned using the elucidations provided in the previous sections (§ 3.3 and 4.3.4) and expert knowledge. For information we set: $\lambda = 0.9$, $\gamma_{\mathcal{R}} = \gamma_{\mathcal{F}} = 0.9$, $\mathcal{T}_s = 3$, $\mathcal{T}_w = 0.5$, $\mathcal{W} = 5$, $\delta_{\emptyset} = 0.2$ and $\Delta_F = 8$.

Decision-taking based on belief: We use the pignistic probability (BetP) proposed by Ph. Smets [17]. We focus on the BetP of actions which are true (*right state*) therefore:

$$\text{BetP}(\mathbf{R}_A) = \frac{1}{(1 - m(\emptyset))} \left(m(\mathbf{R}_A) + \frac{m(\mathbf{R}_A \cup \mathbf{F}_A)}{2} \right) \quad (12)$$

If $\text{BetP}(\mathbf{R}_A) > 0$ then A is considered as true.

Evaluation criteria: The database is annotated. We assess, first, the actions within the sequences and then, the beginning and the end of the sequences.

Recall (\mathcal{R}) and precision (\mathcal{P}) indexes are often used for the evaluation of video processing. They are computed as $\mathcal{R} = \frac{C \cap R}{C}$ and $\mathcal{P} = \frac{C \cap R}{R}$, where C is the reference set obtained by expert annotations, R is the set of retrieved frames provided by the recognition module by using the BetP-based criteria, and $C \cap R$ is the number of correctly retrieved frames. In order to assess the method by only one criterion, the \mathcal{F}_1 -measure is used: $\mathcal{F}_1 = \frac{2 \times \mathcal{R} \times \mathcal{P}}{\mathcal{R} + \mathcal{P}}$. It allows to combine \mathcal{R} and \mathcal{P} (here, the same importance is given to both components). This measure tends to the minimum between \mathcal{R} and \mathcal{P} .

Illustration and analysis: Table 1 gathers the \mathcal{F}_1 -measures reflecting action recognition performance

for each action of each sequence. An important gain is obtained for all actions by using the *Belief Scheduler* because (1) it integrates past events while embedding a memory fading and (2) the model change is constrained according to a sequence, both allowing to remove false alarms.

Table 1: \mathcal{F}_1 -measure (in %) before and after filtering for actions in pole vault and high jump. The last column is the gain/loss on \mathcal{F}_1 after both filtering and scheduling.

High jump	before	after	gain
running	67.9	86.6	+18.7
jumping	57.9	78.4	+20.5
falling	70.1	81.0	+10.9
Sequence	—	91.2	—
Pole vault	before	after	gain
running	73.6	85.6	+12.0
jumping	48.6	80.9	+32.3
falling	64.9	78.5	+13.6
Sequence	—	88.5	—

The illustration depicted in Figure 9 (for a high jump) shows the efficiency of the approach. Belief on propositions R_A (A is right), F_A (A is false) and $R_A \cup F_A$ (A is right or false) are represented.

6 Conclusion

The *Belief Scheduler* of actions proposed for temporal sequence recognition is based on the Transferable Belief Model fusion process.

After the parameters fusion step, the Temporal Belief Filter is used to (i) smooth belief (ii) detect the transition between action using the conflict between model and observation. In order to improve and emphasize the transition between two actions, a cusum of the conflict is computed. Because the human activity can be described by a sequence of actions, we have developed an original *Belief Scheduler* to improve the state evaluation of each action that composes the concerned activity. The preemption and forcing processing, well known in the System Science community has been adapted in order to *re-synchronize* the actions belonging to the sequence. The originality of the proposed algorithm is that it is based on the conflict detection and evaluation: conflict between parameters and constraint state models, and conflict between exclusive actions states. An elucidation of the parameters tuning is also provided to facilitate the setting.

The method was successfully applied on 40 real video sequences acquired with a moving camera to recognize athlete jumps such as *pole vault* and *high jump* as well as actions in jumps such as *running*, *jumping* and *falling*.

The generalization capability was tested by applying successfully the *Belief Scheduler* with the same settings on both activities. The future work concern parameters learning and models adaptation based on gradual

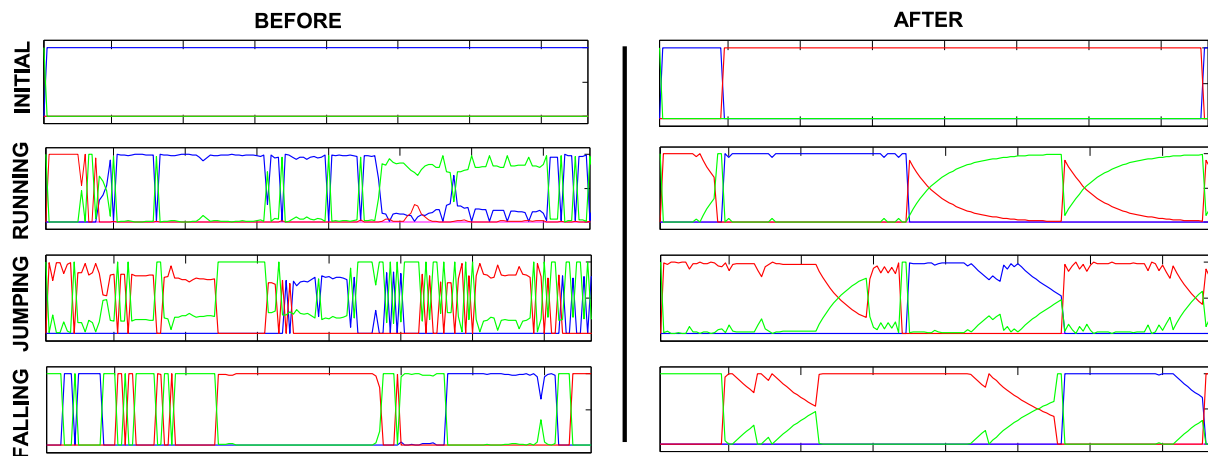


Figure 9: Result of the *Belief Scheduler* for a high jump sequence. The belief on R_A , F_A and $R_A \cup F_A$ are represented respectively in blue, red and green color.

rules. Furthermore, a comparison with ENC [13, 14] will be performed.

7 Acknowledgement

This research is partially supported by SIMILAR European excellence network. The authors would like to thank University of Crete for data exchange.

References

- [1] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *PR*, vol. 36, no. 3, pp. 585–601, 2003.
- [2] M. Lew, N. Sebe, and J. Eakins, "Challenges in image and video retrieval," *Lecture notes in Computer Science, ICIVR*, vol. 2383, pp. 1–6, 2002.
- [3] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition and probabilistic recognition methods," *CVIU*, vol. 96, pp. 129–162, 2004.
- [4] 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.
- [5] E. Ramasso, D. Pellerin, C. Panagiotakis, M. Rombaut, G. Tziritas, and W. Lim, "Spatio-temporal information fusion for human action recognition in videos," in *13th European Signal Processing Conf.*, Antalya, Turkey, Sept. 2005.
- [6] P. Smets and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, 1994.
- [7] E. Ramasso, M. Rombaut, and D. Pellerin, "A temporal belief filter improving human action recognition in videos," in *International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, 2006, to appear.
- [8] T.G. Robertazzi, "Ten reasons to use divisible load theory," *Computer*, vol. 36, pp. 63–68, 2003.
- [9] C. Panagiotakis and G. Tziritas, "Recognition and tracking of the members of a moving human body," in *Articulated Motion and Deformable Objects*, 2004, vol. 3179, pp. 86–98.
- [10] B. Ristic and P. Smets, "Target identification using belief functions and implication rules," *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, no. 3, pp. 1097–1102, July 2005.
- [11] Z. Elouedi, K. Mellouli, and Ph. Smets, "Assessing sensor reliability for multisensor data fusion within the Transferable Belief Model," *IEEE Trans. Systems, Man and Cybernetics*, vol. 34, no. 1, pp. 782–787, 2004.
- [12] M. Rombaut, I. Jarkass, and T. Denoeux, "State recognition in discrete dynamical systems using petri nets and evidence theory," in *ECSQARU*, June 1999, pp. 352–361.
- [13] H. Xu and P. Smets, "Evidential reasoning with conditionnal belief functions," *Uncertainty in Artificial Intelligence*, vol. 94, pp. 598–606, 1994.
- [14] B. Ben Yaghlane, P. Smets, and K. Mellouli, "Directed evidential networks with conditionnal belief functions," in *ECSQARU*, 2003, vol. 2711, pp. 291–305.
- [15] P. Vannoorenberghe and Ph. Smets, "Partially supervised learning by a Credal EM approach," in *ECSQARU*, 2005, vol. 3571, pp. 956–967.
- [16] J. Reuter and W.F. Tichy, "Logging kernel events on clusters," *Future Generation Computer Systems*, vol. 22, no. 3, pp. 313–323, 2006.
- [17] Ph. Smets, "Decision making in the TBM: the necessity of the pignistic transformation," *Int. J. of Approx. Reasoning*, vol. 38, pp. 133–147, 2005.