

L'ordonnanceur crédibiliste pour la reconnaissance de séquences d'états dans le cadre du Modèle des Croyances Transférables

The belief scheduler for state sequence recognition in the Transferable Belief Model framework

Emmanuel Ramasso

Michèle Rombaut

Denis Pellerin

GIPSA-lab, Images and Signal Department, 46, avenue Félix Viallet, 38031 Grenoble, France

e_ramasso@yahoo.fr, {rombaut, pellerin}@lis.inpg.fr

Résumé :

Nous présentons une méthode permettant de reconnaître des séquences d'états décrits par des fonctions de croyance temporelles dans le cadre du Modèle des Croyances Transférables (MCT). L'ordonnanceur crédibiliste sur lequel est basée la méthode repose sur le filtre temporel de fonctions de croyances présenté dans LFA 2006 [9]. Il assure une cohérence temporelle des états tout en lissant les croyances sur chacun des états. Une de ses caractéristiques vient de l'exploitation originale de l'information de conflit, dont une valeur est disponible par le MCT, pour piloter un mécanisme de rupture de modèle. Un critère est proposé pour l'inférence dans un contexte de séquences concurrentes. Ce critère est aussi utilisé pour créer une classe de rejet. L'expérimentation concerne la reconnaissance de sauts d'athlétisme dans des vidéos de sports réelles où nous comparons les performances de l'approche proposée à celles des Modèles de Markov Cachés.

Mots-clés :

Analyse de séquence, Reconnaissance d'activité, Modèle des Croyances Transférables

Abstract:

A method is proposed for state sequence recognition when states are described by belief functions as understood in the Transferable Belief Model (TBM). The method, called Belief Scheduler, relies on the Temporal Belief Filter proposed in a previous LFA's communication [9]. The scheduler ensures temporal coherence of belief on states and is able to separate each state. The scheduler originally exploits conflict information, emphasized in the TBM, which controls a model change process. A criteria is proposed for inference in the context of concurrent state sequences. This criteria is also exploited for a rejection mechanism. The proposed method is applied to athletics jumps recognition in real videos and is compared to Hidden Markov Models.

Keywords:

State sequence analysis, Activity recognition, Transferable Belief Model, Rejection class

1 Introduction

L'analyse de l'activité humaine est un domaine de recherche important et complexe [14] avec de nombreuses applications : indexation de

vidéos, surveillance automatique des personnes ou encore développement d'interfaces Homme-Machine. Le problème principal est de lier le monde réel "continu" à l'interprétation humaine "symbolique" de ce monde réel.

Une *activité* humaine est généralement interprétée comme une séquence d'*actions* primitives [14]. L'objectif est alors de détecter ces actions dans les images d'une vidéo puis d'en déduire les séquences correspondantes. La détection des actions est souvent réalisée par une méthode de type reconnaissance des formes. Puis les séquences sont classiquement reconnues par une méthode basée sur le formalisme (probabiliste) Bayésien [4] telle que les modèles de Markov cachés (HMM) ou les réseaux bayésiens dynamiques (DBN) [6].

Bien que les fonctions de croyance aient la particularité d'être plus générales que les probabilités dans le sens où la mesure de croyances n'est pas additive [5], elles ont été très peu exploitées pour la reconnaissance de séquences. Dans ce formalisme, le doute entre des hypothèses décrivant symboliquement le monde réel peut être explicité. C'est une des différences fondamentales avec les probabilités. Parmi les méthodes reposant sur les fonctions de croyance, nous pouvons citer les réseaux de croyance [12] utilisés pour le diagnostic [15, 2], mais ne traitant pas l'aspect temporel. Celui-ci est en revanche explicite dans le réseau de Pétri crédibiliste [11] et le système IDRES [7] développés pour l'assistance à la conduite.

Dans cette communication, nous présentons une méthode pour reconnaître des séquences d'états dans le cadre du Modèle des Croyances Transférables (MCT). A chaque état est associé une fonction de croyance définie sur un cadre d'hypothèses binaire : l'état est vrai (T : True) ou faux (F : False). La méthode proposée est dénommée *ordonnanceur crédibiliste* et s'inspire de plusieurs mécanismes utilisés lors de l'ordonnement [10]. Par l'intermédiaire du filtre temporel crédibiliste présenté dans LFA 2006 [9], l'ordonnanceur permet de lisser les fonctions de croyance sur chacune des deux hypothèses T ou F en utilisant deux modèles d'évolution (connexions horizontales de la Fig. 1). Lors d'une séquence, il est nécessaire de prendre en compte les informations sur les autres états. En posant comme hypothèse qu'un seul état peut être vrai à tout instant, les autres étant contraints à être faux, l'ordonnanceur crédibiliste permet aussi de gérer la commutation de ces modèles pour tous les états de la séquence (connexions verticales de la Fig. 1). Nous proposons aussi un critère basé sur le conflit pour choisir une séquence parmi plusieurs et pour créer une classe de rejet.

L'ordonnanceur est comparé au HMM pour la reconnaissance de sauts d'athlétisme dans des vidéos réelles acquises par une caméra mobile et angle de vue variable.

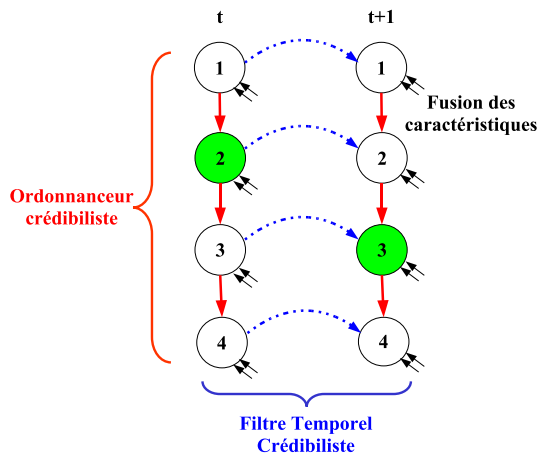


Figure 1 – Méthode de reconnaissance de séquences d'états : l'ordonnanceur inclut le filtre présenté dans [9].

La section 2 permet de rappeler le principe

du filtre, l'ordonnanceur est présenté à la section 3, enfin nous décrivons les résultats expérimentaux à la section 4 avant de conclure.

2 Le filtre temporel crédibiliste

Nous décrivons le filtre dans le cadre de la reconnaissance d'actions humaines au cours du temps. L'action A correspond donc à un état de la séquence. A l'instant f , elle peut être vraie ou fausse donc son cadre de discernement (FoD : frame of discernement), rassemblant les hypothèses sur A , est $\Omega_A^f = \{T_A^f, F_A^f\}$.

La croyance d'une action A est représentée par une distribution de masses (BBA : basic belief assignment) $m^{\Omega_A^f}$ définie sur $2^{\Omega_A^f}$ dans $[0, 1]$ respectant $\sum_{E \subseteq \Omega_A^f} m^{\Omega_A^f}(E^f) = 1$. Ici, $2^{\Omega_A^f} = \{\{T_A^f\}, \{F_A^f\}, \{T_A^f F_A^f\}, \emptyset\}$. Dans la suite, on interprétera les sous ensembles d'hypothèses comme des propositions logiques, et donc $\{T_A^f F_A^f\} \equiv T_A^f \cup F_A^f$ représentera le doute explicite entre les deux hypothèses. Deux BBAs distinctes $m_1^{\Omega_A^f}$ et $m_2^{\Omega_A^f}$ définies sur Ω_A^f peuvent être combinées par [13] :

$$m_1^{\Omega_A^f} \triangle m_2^{\Omega_A^f}(E^f) = \sum_{C^f \triangle D^f = E^f} m_1^{\Omega_A^f}(C^f) \cdot m_2^{\Omega_A^f}(D^f) \quad (1)$$

avec $\triangle = \cap$ (resp. \cup) pour la combinaison conjonctive (resp. disjonctive).

Le filtre utilise un modèle d'évolution \mathcal{M} tel que $\mathcal{M} \in \{\mathcal{T}, \mathcal{F}\}$ (\mathcal{T} pour T_A et \mathcal{F} pour F_A). \mathcal{M} décrit la façon dont évolue la BBA entre deux instants $f-1$ et f . Un modèle et un seul est appliqué à chaque instant f et permet de prédire la BBA $\hat{m}^{\Omega_A^f}$ associée à l'action A à partir de la BBA à l'instant précédent $m^{\Omega_A^{f-1}}$. Le modèle est exprimé sur l'espace joint entre deux instants $\Omega_A^f \times \Omega_A^{f-1}$ par :

$$\text{Modèle } \mathcal{T} \begin{cases} m_{\mathcal{T}}^{\Omega_A^f \times \Omega_A^{f-1}}(E_{\mathcal{T}}) = \gamma_{\mathcal{T}} \\ m_{\mathcal{T}}^{\Omega_A^f \times \Omega_A^{f-1}}(\Omega_A^f \times \Omega_A^{f-1}) = 1 - \gamma_{\mathcal{T}} \end{cases} \quad (2)$$

$$\text{Modèle } \mathcal{F} \begin{cases} m_{\mathcal{F}}^{\Omega_A^f \times \Omega_A^{f-1}}(E_{\mathcal{F}}) = \gamma_{\mathcal{F}} \\ m_{\mathcal{F}}^{\Omega_A^f \times \Omega_A^{f-1}}(\Omega_A^f \times \Omega_A^{f-1}) = 1 - \gamma_{\mathcal{F}} \end{cases}$$

où $E_{\mathcal{T}} = (T_A^f \cap T_A^{f-1}) \cup (F_A^f \cap F_A^{f-1}) \cup (T_A^f \cap F_A^{f-1})$, $E_{\mathcal{F}} = (F_A^f \cap F_A^{f-1}) \cup (T_A^f \cap T_A^{f-1}) \cup (F_A^f \cap T_A^{f-1})$ et $\Omega_A^f \times \Omega_A^{f-1}$ est un espace produit.

La prédiction $\hat{m}^{\Omega_A^f}$ est combinée avec la mesure $\tilde{m}^{\Omega_A^f}$ par la règle conjonctive. De cette combinaison résulte une BBA renforcée $m^{\Omega_A^f}$. Cette combinaison permet de mettre en lumière le conflit entre mesure et prédiction. La décision de changer le modèle est intégrée dans un mécanisme de somme cumulée pondérée du conflit (CUSUM). Tant que celle-ci ne dépasse pas le seuil de stop, le modèle courant est contraint. Sinon, il est changé.

L'algorithme du filtre temporel se résume ainsi :

1. **Prédiction** : elle est donnée par

$$\hat{m}_{\mathcal{M}}^{\Omega_A^f} = (m_{\mathcal{M}}^{\Omega_A^f \times \Omega_A^{f-1}} \odot m^{\Omega_A^{f-1} \uparrow \Omega_A^f \times \Omega_A^{f-1}}) \downarrow \Omega_A^f \quad (3)$$

avec \uparrow l'extension vide [12] et \downarrow la marginalisation.

2. **Fusion de la prédiction avec la mesure** :

$$m^{\Omega_A^f} = \hat{m}_{\mathcal{M}}^{\Omega_A^f} \odot \tilde{m}^{\Omega_A^f} \quad (4)$$

combine les informations disponibles.

3. **Conflit** :

$$\epsilon_f = m^{\Omega_A^f}(\emptyset) \quad (5)$$

quantifie le degré de contradiction entre modèle et données à l'instant courant.

4. **Cusum** : construit la somme cumulée par

$$\text{CS}(f) = \lambda \times \text{CS}(f-1) + \epsilon_f \quad (6)$$

en intégrant un coefficient d'oubli $\lambda \in [0, 1]$.

5. **Décision** : si $\text{CS}(f) > \mathcal{T}_s$ (seuil de stop) à l'instant f_s alors le modèle est changé. Le nouveau modèle est appliqué à partir de f_s et la croyance sur $[f_s - \mathcal{W}, f_s]$ est vide.

6. **Sortie du filtre** : en fonction du conflit courant on a

$$m^{\Omega_A^f} = \begin{cases} \hat{m}_{\mathcal{M}}^{\Omega_A^f} \odot \tilde{m}^{\Omega_A^f} & \text{si } \epsilon_f \leq \delta_\emptyset \\ \hat{m}_{\mathcal{M}}^{\Omega_A^f} & \text{sinon} \end{cases} \quad (7)$$

Le seuil de tolérance au conflit δ_\emptyset est adapté automatiquement sur un intervalle de temps (N) par $\delta_\emptyset = \frac{1}{N} \sum_{f_i=(f-N-1)}^f \epsilon_{f_i}$.

7. **Critère de qualité locale** : nous quantifions la qualité du filtrage à partir du conflit. Si le modèle est contraint sans cesse alors cela signifie que soit les données sont très bruitées soit

le seuil de stop est trop important. Connaissant les intervalles de temps où chaque modèle est appliqué, nous proposons la qualité locale (à t) par :

$$LQ_{i,j}^{f_s:f}(h) = \left(1 - \frac{1}{f - f_s}\right) \cdot LQ_{i,j}^{f_s:(f-1)}(h) + \frac{\text{BetP}^{\Omega_A^f}(h)}{f - f_s} \cdot (1 - \epsilon_f) \quad (8)$$

pour chaque action A_i et $\text{BetP}^{\Omega_A^f}(h)$ est la probabilité pignistique [13] de h . La variable h vaut T_A si le modèle utilisé est \mathcal{T} et vaut F_A si le modèle est \mathcal{F} . Ce critère est calculé en ligne et représente une moyenne glissante intégrant la qualité passée et la probabilité pignistique de l'état en cours pondérée par l'inverse du conflit. L'indice j concerne la séquence S_j .

3 L'ordonnanceur crédibiliste

L'ordonnanceur a été proposé initialement dans [8] pour reconnaître des séquences d'états. Dans ce papier quelques améliorations sont apportées. On considère une séquence S_n (ex : une activité humaine) composé de K états (ex : des actions). Les croyances sur les actions K sont fournies par des données bruitées, imprécises et incertaines. Bien que traitées par le filtre présenté ci-dessus, on aura dans de nombreux cas soit deux actions consécutives qui seront vraies simultanément, soit que l'action amont deviendra fausse avant que l'action aval devienne vraie. Le rôle de l'ordonnanceur est donc de synchroniser les actions.

Le principe de l'ordonnanceur est basé sur l'hypothèse que deux actions ne peuvent être vraies en même temps et qu'une action au moins doit être vraie dès que la séquence a débuté. Les modèles d'évolution des actions \mathcal{T} et \mathcal{F} utilisés dans le filtre sont considérés comme des ressources auxquelles les actions peuvent accéder. Étant donné qu'une seule action peut être vraie à un instant f , un seul modèle du type "l'action est vraie" (\mathcal{T}) sera actif et les autres $K-1$ modèles actifs seront "l'action est fausse" (\mathcal{F}). La "circulation" du modèle \mathcal{T} ne se fait que dans le sens de la séquence, sauf exception que nous décrirons dans la section suivante.

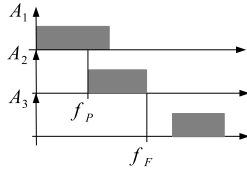


Figure 2 – Actions superposées (A_1 et A_2) et disjointes (A_2 et A_3).

Deux actions consécutives A_k (action en cours, avec le modèle \mathcal{T}) et A_{k+1} (avec le modèle \mathcal{F}) peuvent être (Figure 2) :

- Superposées : à l’instant f_P , appelé instant de PRÉEMPTION, l’action A_{k+1} devient vraie (le filtre de cette action veut prendre le modèle \mathcal{T}) alors que A_k est toujours vraie (le filtre sur cette action ne veut pas changer de modèle et veut garder \mathcal{T}). Donc, deux actions consécutives ont un état *naturel* vrai.
- Disjointes : à l’instant f_F , appelé instant de FORÇAGE, l’action A_k devient fautive (le filtre de cette action veut prendre le modèle \mathcal{F}) alors que A_{k+1} est fautive (le filtre veut garder \mathcal{F}). Donc deux actions ont un état *naturel* faux.

Ces deux cas sont gérés par l’ordonnanceur. Pour cela, nous reprenons les notations du filtre mais en ajoutant l’indice k de l’action. Par exemple la CUSUM de A_k à f est $CS^k(f)$ et son seuil de stop est \mathcal{T}_s^k .

Initialement, l’ordonnanceur attend que la première action soit vraie pour débiter une séquence.

3.1 Gestion des actions superposées : PRÉEMPTION

On considère que l’action en cours est A_k . Une PRÉEMPTION est indiquée par le filtre à un instant f_P lorsque $CS^{k+1}(f) > \mathcal{T}_s^{k+1}$ et que $CS^k(f) < \mathcal{T}_s^k$. Dans ce cas, l’action A_{k+1} prend le modèle vrai (\mathcal{T}) tandis que l’action A_k est contrainte à prendre le modèle faux (\mathcal{F}).

Pour éviter les fausses alarmes, le contexte est sauvegardé avant la commutation des modèles. Donc l’action A_{k+1} est temporairement vraie. Nous verrons dans la section suivante comment nous validons ou invalidons une PRÉEMPTION.

Dans le cas de PRÉEMPTIONS consécutives multiples, la sauvegarde du contexte peut être coûteuse en mémoire, c’est pourquoi nous limitons leur nombre à 2 (au delà elles sont validées).

3.2 Gestion des actions disjointes : FORÇAGE, et détection de fausses alarmes

On considère que l’action en cours est A_k . Un FORÇAGE est indiqué par le filtre à un instant f_F lorsque $CS^k(f) > \mathcal{T}_s^k$ et que $CS^{k+1}(f) < \mathcal{T}_s^{k+1}$. Pour résoudre ce problème, nous proposons tout d’abord d’introduire un paramètre Δ_F qui va permettre, en fonction de la valeur imposée par un utilisateur, trois possibilités :

- si $\Delta_F = 0$ alors l’action suivante est forcée à prendre le modèle vrai,
- si $\Delta_F = \infty$ alors l’action en cours est contrainte à garder le modèle vrai tant que l’action suivante n’est pas vraie,
- si $0 < \Delta_F < \infty$ alors l’action en cours attend une durée finie pendant laquelle l’action suivante peut devenir vraie. Au delà de cette durée, l’action suivante est forcée à prendre le modèle vrai à moins que $CS^{k+1}(f) < \mathcal{T}_s^{k+1}$ i.e. l’action qui demandait à changer de modèle ne le veut plus.

Le paramètre Δ_F permet en quelque sorte de combler l’espace entre deux actions fausses en même temps.

Dans le cas où l’action suivante est forcée à être vraie, il y a deux cas de figure :

- 1) L’action qui exécute un FORÇAGE (A_k) avait préalablement exécuté une PRÉEMPTION (sur A_{k-1}). Dans ce cas, la validation de la PRÉEMPTION intervient (cf section précédente) : une PRÉEMPTION est validée si la qualité de l’action effectuant le FORÇAGE (A_k) est suffisante. Cette qualité est quantifiée par le critère local $LQ_{k+1,n}^{f_P:f_F+\Delta_F}[\mathcal{T}](T_A)$:
 - a) Si la qualité n’est pas suffisante ($< \delta_{FA}$) alors le contexte de A_{k-1} est restitué : l’action A_{k-1} redevient vraie et est forcée à conserver le modèle vrai sur l’intervalle de

fausse alarme donné par $[f_P, f_F + \Delta_F]$ tandis que A_k redevient faux et est contraint à garder le modèle faux sur cet intervalle.

b) Si la qualité est suffisante ($> \delta_{FA}$) alors l'action A_{k+1} est contrainte à prendre le modèle vrai (\mathcal{T}) tandis que l'action A_k prend naturellement le modèle faux (\mathcal{F}).

2) L'action qui exécute un FORÇAGE (A_k) avait préalablement subi un FORÇAGE (par A_{k-1}). Dans ce cas, l'action A_{k+1} est contrainte à prendre le modèle vrai (\mathcal{T}) tandis que l'action A_k prend naturellement le modèle faux (\mathcal{F}).

Le principe de gestion des fausses alarmes est illustré sur la Figure 3.

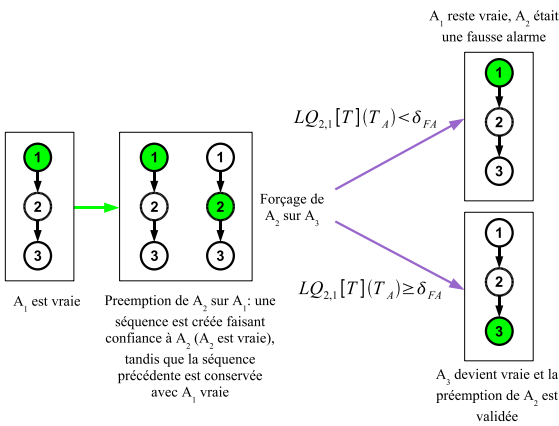


Figure 3 – Détection des fausses alarmes avec utilisation du critère de qualité locale. Le symbole $LQ_{2,1}[\mathcal{T}](T_A)$ donne la qualité de la proposition “action A_2 est vraie” sachant le modèle vrai dans la séquence S_1 .

3.3 Inférence et classe de rejet

Dans le cas où plusieurs séquences concurrentes sont en cours (en surveillance par exemple), il est intéressant de connaître la séquence la plus avancée à un instant donné. C’est donc un problème d’inférence. Notons que l’ordonnancier est un système qui analyse les séquences au fil de l’eau, il en est de même pour l’inférence. Dans les HMM par exemple, l’inférence au fil de l’eau est réalisée par l’algorithme *forward* [6].

Nous proposons un critère nommé *critère global de reconnaissance* (GQ) qui exploite le critère local LQ précédemment défini. Le principe est illustré sur la Figure 4. Pour une séquence S_j composée de K_j actions,

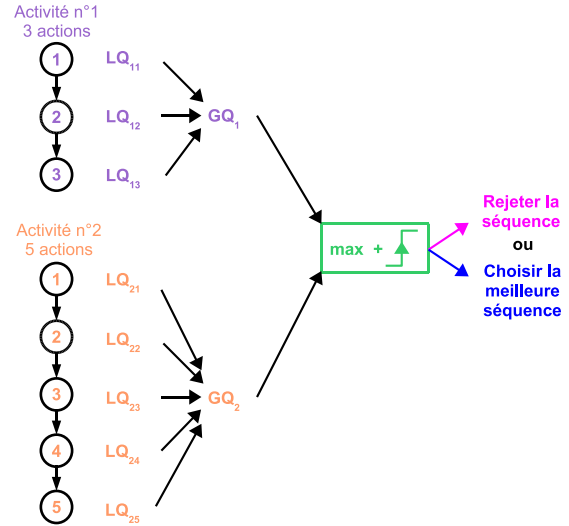


Figure 4 – Inférence : elle est exécutée en ligne par l’agrégation des qualités locales.

nous proposons d’agréger les critères locaux $LQ_{i,j}^{f_s:f}[\mathcal{T}](T_A)$ (les critères sur l’état vrai sachant que le modèle est vrai) pour toutes les actions A_i dans S_j . Le résultat est noté GQ_j^f qui représente la qualité de la reconnaissance de la séquence S_j de f_s (instant de départ) à f (l’instant courant). La fonction d’agrégation utilisée ici est simplement la moyenne :

$$GQ_j^f = \frac{1}{K_j} \sum_{i \in \{1..K_j\}} LQ_{i,j}^{f_s:f}[\mathcal{T}](T_A) \quad (9)$$

Ainsi, la séquence la plus avancée S_*^f à l’instant courant f est celle qui maximise le critère (Figure 4) :

$$S_*^f = S_j^f \quad \text{si} \quad GQ_j^f = \max_k (GQ_k^f > \theta) \quad (10)$$

où θ est un degré de sévérité sur la qualité. Ce paramètre permet de créer une classe de rejet si aucune des séquences ne vérifie le critère. Pour éviter de multiplier les coefficients, la valeur de θ peut être la même que celle de δ_{FA} . La classe de rejet est importante en classification. Elle permet de diminuer les erreurs mais aussi d’attirer l’attention sur des cas “inconnus” et donc peut permettre une éventuelle adaptation. Créer une séquence de rejet avec l’ordonnancier est possible du fait que GQ est borné entre 0% and 100%.

4 Illustration

Nous cherchons à reconnaître des actions et des activités humaines dans les vidéos. La base de tests est composée de 69 vidéos acquises en caméra mobile avec des angles de vue variables. Il y a 26 sauts à la perche (pv), 15 sauts en hauteur (hj), 12 triple sauts (tj) et 16 sauts en longueur (lj), soit environ 13500 images. La base est hétérogène (Figure 5) avec un panel d'angles de vue, des athlètes différents et un environnement très variable.

La reconnaissance des quatre types de sauts passe d'abord par une phase de détection d'actions. Les trois premiers sauts sont décomposés en quatre actions : "course", "saut", "chute" et "relève". Les actions sont détectées à l'aide d'un estimateur de mouvement de caméra qui fournit les paramètres de mouvement de la caméra (supposée suivre l'athlète), et d'un algorithme de suivi de points situés sur l'athlète (tête, centre de gravité et un pied).

Nous comparerons ici les résultats de l'ordonnanceur crédibiliste à ceux des HMM. Afin d'avoir les mêmes sources d'informations pour les deux approches, la transformation numérique-symbolique des caractéristiques de mouvements repose sur un algorithme EM (Expectation-Maximization) [3]. L'EM permet d'estimer des modèles d'action assimilés à des mélanges de Gaussiennes. Ensuite, les vraisemblances (obtenues à chaque image en fonction des caractéristiques) sont transformées en croyances en utilisant une méthode proposée dans [1]. Les vraisemblances sont analysées par les HMM tandis que les croyances sont traitées par l'ordonnanceur.

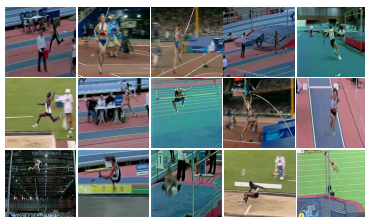


Figure 5 – Quelques images de la base de test. Quelques vidéos et résultats sont disponibles sur www.lis.inpg.fr/pages_perso/ramasso/.

Les paramètres du filtre et de l'ordonnanceur sont fixés en utilisant quelques astuces proposées dans [9, 8] : $\lambda = 0.9$, $\gamma_{\mathcal{T}} = \gamma_{\mathcal{F}} = 0.9$, $\mathcal{T}_s = 3$, $\mathcal{T}_w = 1$, $\mathcal{W} = 5$, $\delta_{FA} = 50\%$ et $\Delta_F = 20$. Les actions étant plus courtes pour le triple saut, nous avons diminué Δ_F et \mathcal{T}_s .

Nous construisons quatre HMM et quatre ordonnanceurs, un pour chaque type de saut. Tous possèdent quatre états, un pour chaque action, excepté le HMM et l'ordonnanceur du triple saut qui possèdent huit états (course, saut, chute, saut, chute, saut, chute et relève). Les paramètres des HMM (a priori, matrice de transition et modèles d'observations) sont estimés avec la boîte à outils BNT [6]. Chaque état est modélisé par une mélange de gaussiennes dont le nombre de composantes est estimé avec un critère de type MDL [3] et varie entre 3 et 6.

L'évaluation de la reconnaissance des HMM est réalisée avec un algorithme de Viterbi qui fournit une log-vraisemblance (critère) de la meilleure séquence d'état estimée. Ainsi, pour une vidéo inconnue, quatre log-vraisemblances sont calculées. La valeur maximum donne le type de saut correspondant à la vidéo. Pour l'ordonnanceur, on obtient un critère global GQ pour chaque saut et on maximise là aussi le critère pour la classification finale de la vidéo. Quelques exemples de critères obtenus sont donnés sur les Figures 8(b) et 9 pour des sauts à la perche.

Les résultats de la classification sont donnés par la matrice de confusion du tableau 1. On montre ici que les résultats de l'ordonnanceur sont bien meilleurs. Le taux de classification global est de 71% sans rejet et de 93% avec rejet pour l'ordonnanceur contre 54% pour le HMM. Les mauvais résultats des HMM s'expliquent en partie par leur sensibilité à la durée des actions. Ceci est d'autant plus vrai que la classification des triples sauts a été réalisée manuellement (un triple saut est reconnu si deux sauts et deux chutes sont reconnus) car les actions sont très courtes exceptée la course. Sans la classification manuelle, le taux de bonne classification était proche de 0%. Notons que les HMM

utilisés ne sont pas les plus complexes existants [6] : il existe de nombreuses autres formes permettant peut être d'améliorer les résultats au détriment de l'interprétation du système de reconnaissance (boîte noire) et du temps/difficulté d'apprentissage.

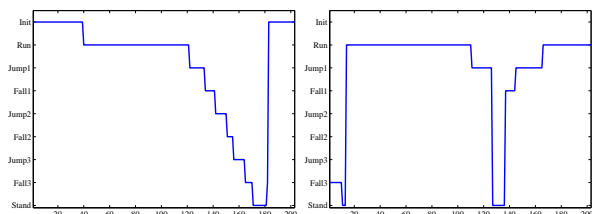


Figure 7 – Décision sur la détection des actions dans le triple saut de la figure 6 : par l'ordonnanceur (gauche) et par l'algorithme de Viterbi (droite).

Vérité terrain					Vérité terrain				
	pv	lj	tj	hj		pv	lj	tj	hj
pv	19	1	0	0	pv	14	5	1	0
lj	0	9	0	0	lj	11	9	0	9
tj	0	0	9	0	tj	0	1	9	0
hj	2	2	0	12	hj	1	1	2	6
rej	5	4	3	3					

Tableau 1 – Résultats de classification par matrice de confusion. Gauche : par l'ordonnanceur. Droite : par le HMM. "rej" signifie classe de rejet.

Les Figures 6-7 montre la reconnaissance des actions dans une séquence de triple sauts (8 états) bruitée avec en 6(a) l'évolution des croyances obtenues à partir des vraisemblances, 6(b) l'évolution des croyances après filtrage et ordonnancement avec le système proposé, 7(a) les états finalement détectés par l'ordonnanceur et 7(b) le décodage de Viterbi dans le cas des HMM. L'action située en haut des figures est la "course". La figure 7(a) montre que l'ordonnanceur est bien meilleur pour la reconnaissance des différentes actions que l'algorithme de Viterbi (6(b)).

La Figure 8 présente l'évolution de la log-vraisemblance des HMM et du critère global GQ pour 26 sauts à la perche analysés par les quatre types de modèles (hj, lj, pv, tj). A partir de cette courbe on peut retrouver quelques résultats de la matrice de confusion précédente. La ligne en trait gras (bleue) représente la réponse (GQ) de l'ordonnanceur

associée au saut à la perche (le vrai modèle). On voit sur cette figure que le critère proposé GQ (Figure 8(b)) est beaucoup plus fiable que le critère de log-vraisemblances (Figure 8(a)) dans le sens où d'une part les résultats sont meilleurs en terme de taux de bonne classification mais aussi que la différence entre les deux valeurs GQ les plus élevées est plus grande donc la discrimination semble meilleure et la décision est plus fiable. Les pointillés sur la Figure 8(b) représentent le seuil de rejet (50%). Les vidéos 9, 13, 19, 22 et 24 sont rejetées (points bleus) en partie parce que ces vidéos sont acquises sous un angle de vue difficile (de face). Les points rouges représentent des cas de mauvaise classification. La Figure 9 illustre l'évolution du critère GQ pour un saut à la perche (critère calculé en ligne). Cette figure est intéressante pour le monitoring par exemple. On voit que le système décide que la vidéo est un saut à la perche avec une qualité d'environ 75%.

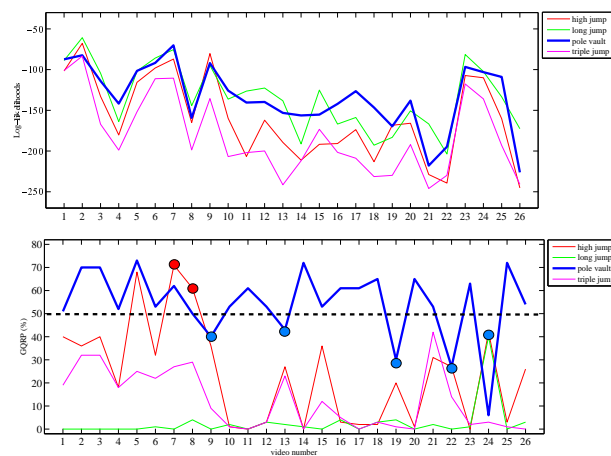


Figure 8 – Evolution des critères de classification pour les quatre types de modèles et appliqués sur 26 sauts à la perche. (Haut) Log-vraisemblances dans les HMM. (Bas) Critère GQ dans l'ordonnanceur. Point rouges : erreurs. Points bleus : rejets.

5 Conclusion

L'ordonnanceur crédibiliste proposé pour la reconnaissance de séquences d'états dans le cadre du Modèle des Croyances Transférables (MCT) permet de lisser les fonctions de croyances sur un cadre de discernement binaire et de séparer les hypothèses de chaque état. Il re-

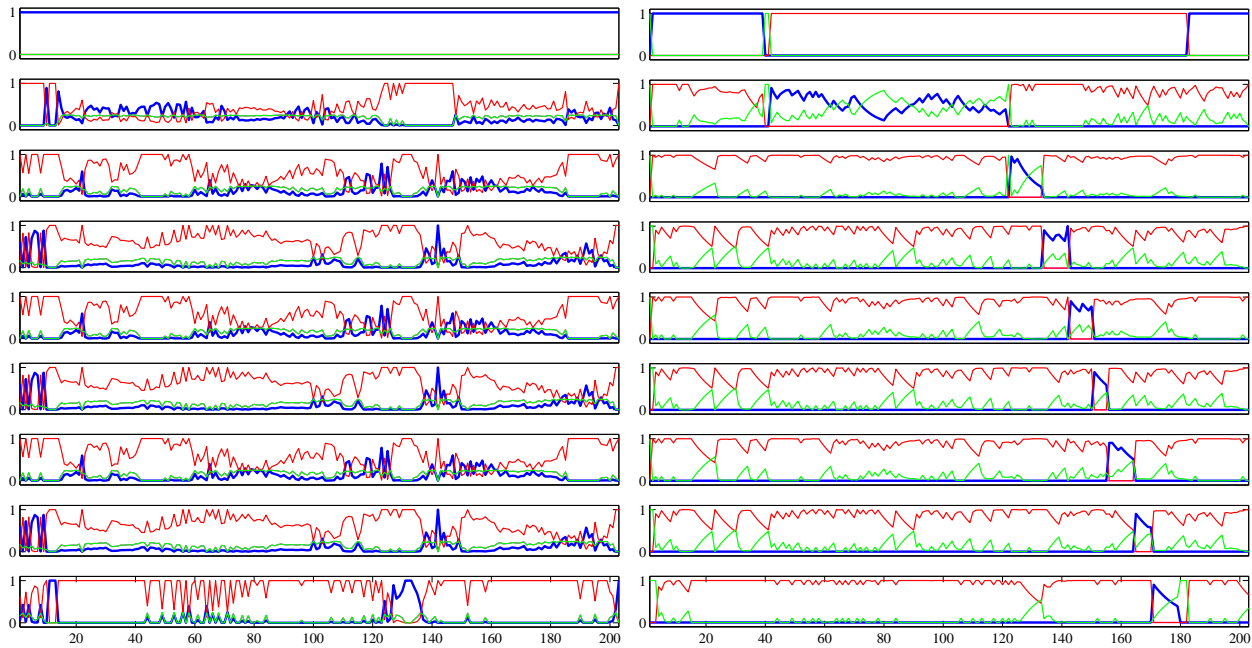


Figure 6 – Filtrage et séquençage d'un triple saut dans une vidéo de basse qualité. Croyances sur $[0, 1]$ avant (gauche) et après (droite) traitement par l'ordonnancement (BBA sur : T_A (bleu/gras), F_A (rouge), $F_A \cup T_A$ (vert), \emptyset (magenta)).

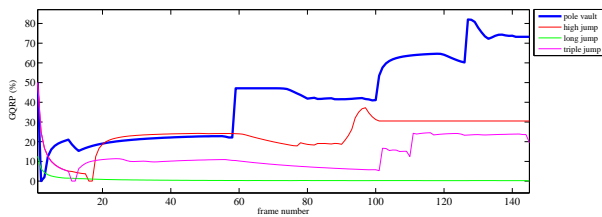


Figure 9 – Evolution du critère GQ pour un saut à la perche.

pose sur un critère construit à partir du conflit. Un autre critère est aussi proposé pour inférer des séquences concurrentes ainsi que pour créer une classe de rejet. Les expériences ont montré de très bonnes performances pour classer des vidéos en comparaison aux HMM habituellement utilisés. L'intérêt du critère pour l'inférence, la création d'une classe de rejet et le monitoring a aussi été souligné. Les travaux actuels se focalisent sur l'adaptation de l'ordonnancement (nombre d'états variable).

Références

- [1] F. Delmotte and Ph. Smets. Target identification based on the Transferable Belief Model interpretation of Dempster-Shafer model. *IEEE Trans. on Systems, Man and Cybernetics*, 34(4), 2004.
- [2] S. Démotier, W. Schön, and T. Denoeux. Risk assessment based on weak information using belief functions : A case study in water treatment. *IEEE Trans. on Systems, Man and Cybernetics*, 36(3) :382–396, 2006.
- [3] M.A.T Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24, 2002.
- [4] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96 :129–162, 2004.
- [5] G.J. Klir and M.J. Wierman. *Uncertainty-based information. Elements of generalized information theory, 2nd edition*. Studies in fuzzyness and soft computing. Physica-Verlag, 1999.
- [6] K. P. Murphy. *Dynamic Bayesian Networks : Representation, inference and learning*. PhD thesis, UC Berkeley (CSD), 2002.
- [7] J.M. Nigro and M. Rombaut. Idres : A rule-based system for driving situation recognition with uncertainty management. *Information Fusion*, 4 :309–317, 2003.
- [8] E. Ramasso, D. Pellerin, and M. Rombaut. Belief Scheduling for the recognition of human action sequence. In *Int. Conf. on Information Fusion*, Florence, Italia, 2006.
- [9] E. Ramasso, M. Rombaut, and D. Pellerin. Un filtre temporel crédibiliste pour la reconnaissance d'actions humaines dans les vidéos. In *LFA*, 2006.
- [10] T.G. Robertazzi. Ten reasons to use divisible load theory. *Computer*, 36 :63–68, 2003.
- [11] M. Rombaut, I. Jarkass, and T. Denoeux. State recognition in discrete dynamical systems using Petri nets and Evidence theory. In *Europ. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 352–361, 1999.
- [12] P. Smets. Beliefs functions : The Disjunctive Rule of Combination and the Generalized Bayesian Theorem. *Int. Jour. of Approximate Reasoning*, 9 :1–35, 1993.
- [13] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66(2) :191–234, 1994.
- [14] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3) :585–601, 2003.
- [15] H. Xu and P. Smets. Evidential reasoning with conditional belief functions. *Uncertainty in Artificial Intelligence*, 94 :598–606, 1994.