Synchronisation de vidéos

E. Dexter[1] P. Pérez[1] I. Laptev[1] I. Junejo[1]

[1]IRISA/INRIA - Centre de Recherche Rennes-Bretagne Atlantique

Campus Universitaire de Beaulieu
35042 RENNES Cedex, FRANCE

{emilie.dexter, patrick.perez, ivan.laptev, imran.junejo}@irisa.fr

Résumé

Un nombre croissant d'applications de vision par ordinateur utilise des séquences d'images issues de multiples caméras. Une étape critique de telles applications est la synchronisation des vidéos. Nous proposons une approche originale qui exploite les similarités et dissimilarités temporelles d'une séquence comme descripteur temporel de celle-ci. La synchronisation consiste à aligner ces descripteurs issus de séquences différentes par programmation dynamique. Notre approche est simple, ne requiert aucune correspondance de points alors qu'elle s'accommode d'importants changements de vues. La méthode a été validée sur des bases de données publiques avec des conditions de vues contrôlées ainsi que sur des vidéos naturelles.

Mots clefs

Synchronisation, alignement temporel, auto-similarité

1 Introduction

Ces dernières années, nous assistons à l'augmentation et la diversification des dispositifs d'enregistrement vidéos : camescopes, appareils photo numériques, téléphones mobiles, caméras de surveillance. Par conséquent, l'enregistrement simultané d'une même scène dynamique devient de plus en plus probable notamment pour des événements sportifs ou autres performances publiques. De telles vidéos diffèrent souvent par le point de vue et le mouvement de la caméra. De plus, la synchronisation se révèle indispensable à des applications comme la synthèse de nouvelles vues ou la reconstruction de scènes dynamiques.

Dans la littérature, la synchronisation est étudiée en considérant des hypothèses de caméras statiques et de transformation linéaire des axes temporels. La majorité des approches exploitent des correspondances spatiales entre les vues, soit pour estimer une matrice fondamentale [1], soit pour utiliser des contraintes de rang sur des matrices d'observations [2]. D'autres méthodes essaient d'extraire des caractéristiques temporelles sans correspondance, comme dans [3] où les auteurs explorent des descripteurs "images" pour la synchronisation. Finalement, peu d'articles traitent de la synchronisation automatique sans contrainte. Dans [4] par exemple, les auteurs choisissent manuellement 5 points mobiles indépendants afin de s'assurer que les points sont suivis correctement tout au long des séquences.

Nous nous intéressons dans cet article à la synchronisation automatique de vidéos sans recours à des correspondances ou à un a priori sur le type de synchronisation. Nous explorons un descripteur original, stable aux changements de vue et reposant sur des auto-similarités temporelles. La synchronisation s'effectue par alignement de ces descripteurs à l'aide d'un algorithme de programmation dynamique connu sous le nom de *Dynamic Time Warping* (DTW).

1.1 Travaux Antérieurs

Nos travaux sont liés par la notion d'auto-similarité aux méthodes proposées dans [5, 6, 7]. Cette notion est exploitée dans [7] afin de mettre en correspondance des images ou des vidéos et de détecter des actions dans des vidéos. Les auteurs calculent pour cela un descripteur local pour chaque pixel. Mettre en correspondance un modèle d'image ou d'action avec un autre revient à trouver un ensemble similaire de descripteurs. Cependant, la notion d'auto-similarité que nous exploitons ici est plus proche des travaux [5, 6] où les auteurs construisent une matrice de similarité où chaque coefficient est un score de corrélation en valeur absolue entre les silhouettes des objets en mouvement. Ce type de matrice est utilisée respectivement pour la détection de mouvement périodique et la reconnaissance de démarche.

De manière analogue à l'approche proposée dans [8], nous utilisons l'algorithme DTW qui permet d'estimer une transformation non-linéaire des axes temporels des séquences. Cependant, contrairement à ces travaux, nous n'utilisons pas de correspondances spatiales entre les séquences d'images qui, en pratique, sont difficiles à obtenir.

1.2 Approche Proposée

Dans cet article, nous proposons une approche originale de synchronisation automatique des vidéos d'un même évènement dynamique enregistré par des caméras correspondant à des points de vue différents.

Nous exploitons pour cela les matrices d'auto-similarité (*Self-Similarity Matrix* - SSM) comme descripteur tempo-

rel de séquences comme dans [9] pour de la reconnaissance d'actions. Bien que ces matrices ne soient pas strictement invariantes aux changements de vue, elles sont néanmoins relativement stables, ceci est illustré Fig. 1(b,d). Ces matrices, calculées pour différentes vues d'un même swing de golf en utilisant les distances entre les points sur la trajectoire de la main, ont des structures similaires. Cette similitude s'explique par le fait que des points proches sur la trajectoire, A et B, restent proches dans les deux vues alors que des points éloignés, A et C, restent éloignés dans ces mêmes vues ce qui implique pour les points correspondant dans les SSMs de faibles valeurs et de fortes valeurs respectivement. Ceci conduit pour un même évènement dynamique à des matrices d'auto-similarité dont les structures sont similaires. Nous développons ensuite un descripteur temporel afin de caractériser ces structures. L'alignement de ces descripteurs par DTW permet enfin d'obtenir des correspondances temporelles exhaustives entre les séquences d'images.

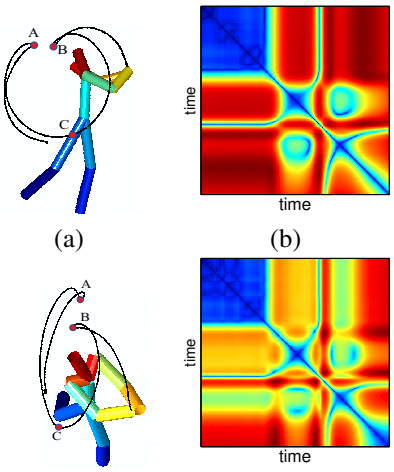


Figure 1 – *(a) et (c) montrent un swing de golf vu sous deux angles différents. (b) et (d) représentent les matrices d'auto-similarité calculées à partir de trajectoires 2D.*

La suite de l'article s'organise comme suit. La section 2 introduit les descripteurs pour les vidéos. La section 3 décrit l'alignement des descripteurs par l'algorithme DTW. Enfin la section 4 est consacrée aux résultats de synchronisation aussi bien sur des bases de données publiques que sur nos propres séquences.

2 Descripteurs de Vidéos

Dans cette section, nous introduisons la description temporelle des vidéos. Nous décrivons tout d'abord le calcul et les propriétés des matrices d'auto-similarité. Ensuite, un descripteur local pour SSM est proposé.

2.1 Matrices d'Auto-Similarité (SSM)

Notre principale hypothèse consiste à affirmer que les similarités et dissimilarités temporelles de la séquence sont préservées sous des changements de vue. En conséquence, le même évènement dynamique enregistré sous des vues différentes produit des matrices avec des structures similaires permettant la synchronisation.

Pour une séquence d'images notée $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \dots \mathcal{I}_T\}$, définie dans un espace discret (x, y, t), la matrice d'auto-similarité est une matrice carrée symétrique $\mathcal{D}(\mathcal{I})$ définie dans $\mathbb{R}^{T\times T}$ comme une table exhaustive de distances entre les caractéristiques des images prises par paires dans l'ensemble $\mathcal{I}$:

$$\mathcal{D}(\mathcal{I}) = [d_{ij}] = \begin{bmatrix} 0 & d_{12} & \dots & d_{1T} \\ d_{21} & 0 & \dots & d_{2T} \\ \vdots & \vdots & & \vdots \\ d_{T1} & d_{T2} & \dots & 0 \end{bmatrix} \quad (1)$$

où d_{ij} représente la distance entre les caractéristiques extraites des images $\mathcal{I}_i$ et $\mathcal{I}_j$ respectivement. Chaque élément de la diagonale, qui correspond à comparer une image avec elle-même, est par conséquent nul. Les motifs de la matrice $\mathcal{D}(\mathcal{I})$ sont déterminés par la distance et par les caractéristiques utilisées pour calculer ses coefficients. Dans ces travaux, nous utilisons la distance euclidienne ainsi que deux types de caractéristiques pour calculer les coefficients d_{ij} de la matrice $\mathcal{D}(\mathcal{I})$.

Caractéristiques "trajectoires". Nous considérons tout d'abord les similarités basées sur les trajectoires de points appartenant à un objet en mouvement. Les coefficients d_{ij} sont exprimés comme la distance euclidienne entre les positions des points suivis pour une paire d'images de la séquence. La mesure de similarité entre les points suivis dans les images $\mathcal{I}_i$ et $\mathcal{I}_j$ est calculée par :

$$d_{ij} = \sum_k \|x_i^k - x_j^k\|_2 \quad (2)$$

où k indique le point suivi, et i et j indiquent les indices des images dans la séquence $\mathcal{I}$. Ces caractéristiques "trajectoires" sont utilisées dans nos expériences sur des données de *Motion Capture* 3D (MOCAP) présentées au paragraphe 4.1 où les points "suivis" correspondent aux articulations du corps humain. Nous notons cette matrice SSM-pos.

Caractéristiques "images". Nous proposons également d'utiliser des caractéristiques "images" de deux types différents : les vecteurs du flot optique et les histogrammes orientés du gradient (HoG) [10]. En pratique, le flot optique est calculé en utilisant la méthode proposée par Lucas et Kanade [11] soit dans une boîte englobante centrée sur l'objet soit sur l'image entière. Le vecteur global du flot optique est obtenu en concaténant les deux directions du flot. Alors que le flot optique exprime les mouvements, les vecteurs HoG, à l'origine utilisés pour de la détection, caractérisent la forme locale au travers des structures du gradient. Notre implémentation utilise des histogrammes de 4 classes pour chaque bloc 5 x 7 défini soit sur une boîte englobante soit sur l'image entière. Pour ces deux types de caractéristiques, d_{ij} est la distance euclidienne entre deux vecteurs correspondants aux images $\mathcal{I}_i$ et $\mathcal{I}_j$. Les matrices calculées pour ces caractéristiques sont notées respectivement SSM-of et SSM-hog.

2.2 Descripteur

La matrice d'auto-similarité est symétrique semi-définie positive avec une diagonale nulle et sa structure est stable aux changements de vue. Cette structure est primordiale pour synchroniser. Il faut donc construire des descripteurs appropriés afin de la caractériser. Comme, les structures globales des SSMs peuvent être influencées par des délais et/ou des déformations temporels, nous optons pour une représentation locale pour les descripteurs. De plus, l'incertitude des valeurs augmente avec la distance par rapport à la diagonale à cause de la difficulté grandissante à mesurer des auto-similarités sur de longs intervalles de temps.

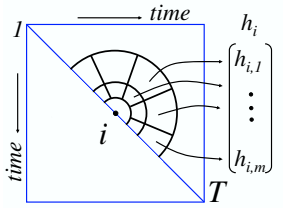


Figure 2 – *Les descripteurs locaux d'une SSM sont centrés sur chaque point de la diagonale $i = 1...T$ et reposent sur une structure log-polaire. Les histogrammes de directions du gradient sont calculés séparément pour chaque bloc et concaténés en un descripteur h_i.*

Comme illustré Fig. 2, nous calculons pour chaque point diagonal un descripteur sur une structure log-polaire. Nous construisons un histogramme à 8 classes suivant la direction du gradient pour chacun des 11 blocs et concaténons ces histogrammes normalisés en un vecteur descripteur h_i correspondant à l'image i. Finalement, la séquence d'images est représentée par une séquence de descripteurs $H = (h_1, ..., h_T)$.

3 Alignement des Descripteurs

Nous souhaitons aligner les descripteurs temporels extraits des SSMs par un algorithme classique de DTW. Une telle approche, qui a été introduite pour aligner deux signaux temporels notamment pour de la reconnaissance de la parole [12], est particulièrement bien adaptée à ce problème. Étant données deux séquences d'images I^1 et I^2 représentant le même évènement dynamique de différents points de vue, nous calculons les SSMs et les descripteurs correspondants, $H^1 = (h_1^1, ...h_i^1, ..., h_N^1)$ et $H^2 = (h_1^2, ...h_j^2, ..., h_M^2)$.

L'algorithme de DTW a pour but d'estimer une fonction d'alignement w entre les axes temporels des deux séquences. L'alignement entre les images i et j des deux séquences est exprimée par $j = w(i)$.

Étant donné une mesure de dissimilarité S, une petite valeur de $S(h_i^1, h_j^2)$ indiquant une grande similarité entre h_i^1 et h_j^2, nous définissons une matrice de coût $\mathcal{C}$ comme

$$\mathcal{C} = [c_{ij}] = \left[S(h_i^1, h_j^2)\right]. \quad (3)$$

Chaque coefficient de cette matrice mesure le coût d'alignement entre les descripteurs de chacune des séquences considérés respectivement aux instants i et j. Le meilleur alignement est l'ensemble des paires $\{(i, j)\}$ qui contribuent au coût global minimum de similarité. En conséquence, la fonction optimale de mise en correspondance temporelle, w, doit minimiser le coût cumulé C_T :

$$C_T = \min_w \sum_{i=1}^{N} S(h_i^1, h_{w(i)}^2) \quad (4)$$

Pour résoudre cette équation en utilisant la programmation dynamique, nous devons construire la matrice de coût cumulé C_A à partir de la matrice de coût $\mathcal{C}$. En considérant 3 déplacements possibles (horizontal, vertical et diagonal) dans $\mathcal{C}$, nous pouvons calculer récursivement pour chaque paire d'instants (i, j), $C_A(h_i^1, h_j^2)$ par

$$\begin{aligned} C_A(h_i^1, h_j^2) &= c_{ij} + \min[C_A(h_{i-1}^1, h_j^2), \\ &\quad C_A(h_{i-1}^1, h_{j-1}^2), C_A(h_i^1, h_{j-1}^2)] \end{aligned} \quad (5)$$

Les déplacements verticaux et horizontaux correspondent à l'association d'une image dans une séquence à deux images consécutives dans la seconde alors qu'un déplacement diagonal revient à associer deux paires d'images consécutives. La solution finale C_T de (4) est par définition $C_T = C_A(h_N^1, h_M^2)$. La fonction d'alignement, w, est obtenue en retraçant le chemin optimal dans la matrice de coût cumulé C_A en partant de la paire (N, M). L'algorithme DTW requiert une mesure de distance $S(\cdot, \cdot)$ afin d'évaluer de coût d'alignement. Nous avons opté pour la distance euclidienne.

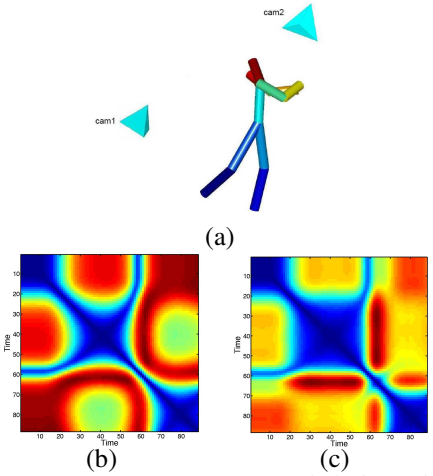
4 Résultats de Synchronisation

Les paragraphes 4.1 et 4.2 présentent la validation de notre méthode dans des conditions multi-vues contrôlées en utilisant des données de *Motion Capture* (MOCAP) et des séquences d'images issues de la base IXMAS [13]. Ensuite, nous testons la méthode sur des séquences d'images naturelles au paragraphe 4.3. Finalement, nous comparons nos résultats d'alignement avec la méthode proposée par Wolf et Zomet dans [2] au paragraphe 4.4.

4.1 Données CMU MOCAP

Nous avons utilisé des données MOCAP 3D provenant de la base CMU (*mocap.cs.cmu.edu*) afin de simuler des conditions de vues multiples et contrôlées d'une même action dynamique. Les trajectoires de points situés sur le corps humain sont projetées sur deux caméras avec des orientations prédéfinies, comme illustré Fig. 3(a). Les effets de translation et d'échelle doivent être éliminés de telle sorte que les points soient centrés sur zéro. Les points sont donc normalisés par $\mathbf{x}_i = \frac{\mathbf{x}'_i}{\|\mathbf{x}'_i\|}$, où $\mathbf{x}'_i$ correspond aux points suivis dans l'image i et $\mathbf{x}_i$ correspond à leurs coordonnées normalisées. Des exemples de SSMs, calculées pour ces deux projections, sont montrées en Fig. 3(b,c).

Pour ces données, les SSMs peuvent être calculées et synchronisées en présence d'un désalignement temporel simulé. Nous choisissons d'appliquer la plus simple transformation temporelle possible : le délai temporel. Nous tron-



(a) (b) (c)

Figure 3 – *(a) Un personnage animé de la base de données CMU et deux caméras virtuelles utilisées afin de simuler des projections. (b) SSM pour cam1. (c) SSM pour cam2.*

quons simplement les SSMs pour le simuler comme illustré Fig. 4(a,b). Nous alignons temporellement les descripteurs des SSMs tronquées en estimant le chemin optimal dans la matrice de coût (Fig. 4(c)). Ce chemin, illustré en rouge Fig. 4(c,d), retrouve presque parfaitement la vérité terrain représentée par la courbe bleue en Fig. 4(d). Ces expériences avec des conditions de vues contrôlées valident notre méthode pour une déformation de type délai temporel.

4.2 Données IXMAS

Nous avons également testé notre méthode sur les séquences d'images de la base IXMAS [13]. Cette base regroupe les séquences de 10 acteurs réalisant 11 classes d'action sous 5 vues différentes. Les acteurs ont choisi arbitrairement leur position et leur orientation par rapport aux caméras. Pour ces séquences, nous calculons des SSM-of. Le flot optique est calculé à l'intérieur de boîtes englobantes centrées sur les acteurs, obtenues à partir des silhouettes disponibles pour cette base.

Comme les séquences de cette base sont synchronisées, nous simulons comme précédemment un désalignement temporel. Considérant deux vues d'une même succession d'actions, nous avons appliqué, en plus du délai temporel, la transformation non linéaire $t' = a\cos(bt)$. Une illustration est proposée en Fig. 5. Nous pouvons observer que la transformation estimée, illustrée par la courbe rouge de la Fig. 5(d), coïncide presque parfaitement avec la vérité terrain (courbe bleue) malgré la différence de point de vues entre les deux séquences comme le montre la Fig. 5(a).

Notons que le début et la fin de l'estimation ne correspondent pas exactement avec la vérité terrain. Ceci est dû au fait que l'algorithme DTW suppose, à tort, que les chemins admissibles se terminent en (N, M). Malgré les fausses correspondances que cette contrainte cause, l'algorithme est capable de retrouver quasiment la totalité de la vérité terrain. Ces résultats démontrent que notre approche permet l'estimation de transformations aussi bien linéaires que non linéaires, même pour des vues extrêmement différentes.

4.3 Séquences d'Images Naturelles

Nous testons maintenant la méthode sur des séquences d'images d'objets en mouvement ou d'activités humaines. Pour celles-ci, nous calculons le flot optique entre deux images consécutives puis les SSM-of et les descripteurs correspondants.

Séquences avec des Objets Mobiles. Les séquences de cet exemple représentent deux balles rebondissant sur une table, selon deux points de vue différents : une vue de dessus et une vue de côté. Une illustration de la configuration de cette scène est proposée Fig. 6, où les deux courbes en couleur représentent les trajectoires des balles. Les résultats de synchronisation sont présentés en Fig. 7(c). La transformation originale entre les séquences est partiellement retrouvée. En effet, au début et à la fin de chacune des séquences il n'y a pas de mouvement ce qui conduit à un mauvais alignement lié à un manque d'information temporelle.

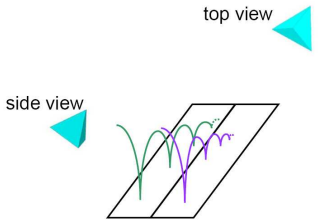


Figure 6 – *Configuration des séquences avec deux balles rebondissant sur une table.*

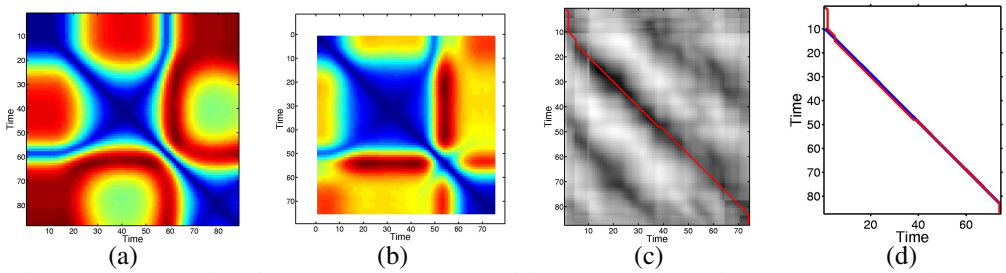
De plus, notre approche se heurte à quelques difficultés en présence de mouvements périodiques. En effet, ces mouvements induisent des structures périodiques au sein des SSMs et causent des ambiguïtés pour l'algorithme DTW. Quand le mouvement est presque périodique ce qui est le cas dans cet exemple, les performances de notre approche peuvent dépendre de l'amplitude du délai. Dans cet exemple, les ambiguïtés sont levées pour deux raisons : le délai est court et deux balles sont présentes.

Séquences d'Activités Humaines. Ces séquences, présentées Fig. 8(a), montrent quatre joueurs de basket-ball qui sortent du champ des caméras à certains instants. L'estimation de la transformation temporelle, illustrée par la courbe rouge en Fig. 8(c), est correcte (courbe bleue). Il est important de noter que les apparitions et les disparitions des joueurs ne perturbent pas l'estimation.

Il est important de noter que la méthode ne peut pas fournir de correspondances temporelles à la sous-image près ce qui dans le cas de la synchronization de séquences acquises à des fréquences temporelles différentes devrait se traduire par des palliers horizontaux ou verticaux au niveau de la transformation estimée ce qui signifie qu'une image d'une séquence est associée à plusieurs images dans la seconde.

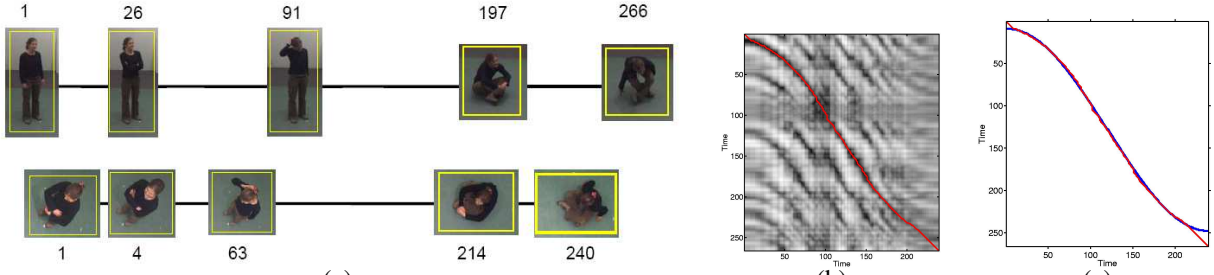
4.4 Comparaison

Nous souhaitons ici comparer notre méthode avec l'approche proposée par Wolf et Zomet (WZ) [2]. Cette approche peut être utilisée pour des transformations de type



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

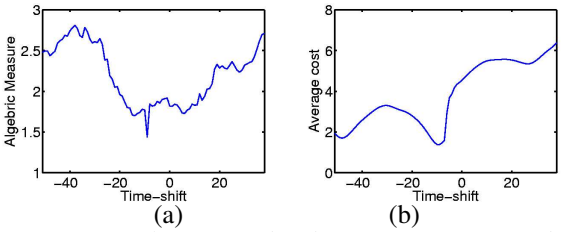
Figure 4 – *Synchronisation pour les données CMU pour un délai temporel simulé. (a) SSM originale pour cam1. (b) SSM tronquée pour cam2. (c) Représentation de la matrice de coût avec l'estimation de la transformation (courbe rouge). (d) L'estimation retrouve la transformation originale (courbe bleue).*



(a) (b) (c)

Figure 5 – *Synchronisation pour une déformation temporelle non linéaire. (a) Séquences représentées par quelques images. (b) Matrice de coût avec l'estimation de la transformation (courbe rouge). (c) Cette estimation recouvre presque complètement la transformation originale (courbe bleue).*

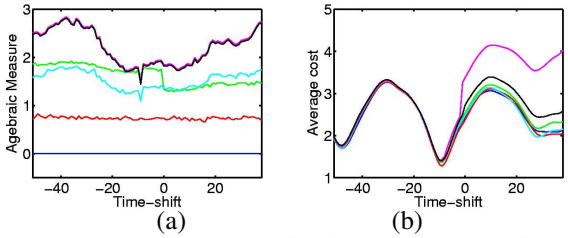
délai temporel. Pour chaque valeur de délai possible, les auteurs évaluent une mesure algébrique basée sur des contraintes de rang de matrices de trajectoires. Le délai estimé correspond à la valeur qui minimise cette mesure. Leurs résultats sont présentés sous la forme de courbes donnant cette mesure calculée en fonction du délai, comme illustré en Fig. 9(a). Afin d'obtenir une représentation analogue, nous calculons la valeur moyenne du coût pour le chemin correspondant à un délai temporel donné. Nous pouvons alors représenter cette valeur moyenne en fonction du délai, comme illustré en Fig. 9(b). Le minimum de cette courbe représente le délai estimé. La Fig. 9 présente les résultats pour les deux méthodes pour le même exemple que celui de la Fig. 4 mais en utilisant 20 trajectoires choisies aléatoirement pour chaque séquence.



(a) (b)

Figure 9 – *Résultats pour des données MOCAP non bruitées (a) Résultats pour WZ (b) Résultats par la méthode proposée.*

Afin de comparer la robustesse des deux approches, nous ajoutons du bruit aux trajectoires avec des variances différentes. Nous pouvons observer à la Fig. 10 que pour des bruits de faible variance (courbes noir, magenta et cyan) les deux méthodes retrouvent le délai. Cependant, pour des variances plus élevées, notre méthode retrouve le délai alors



(a) (b)

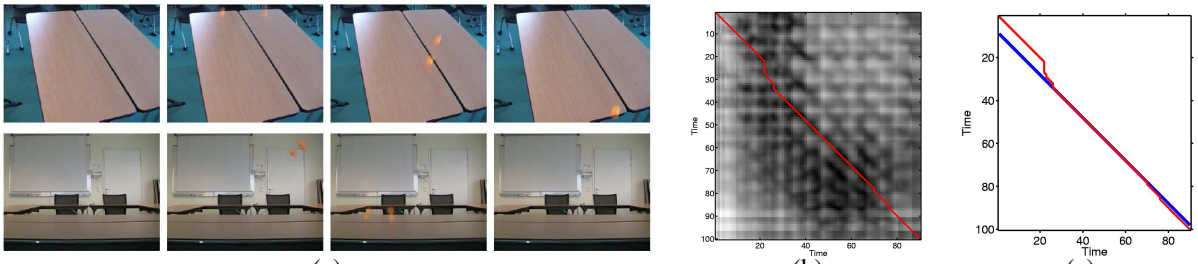
Figure 10 – *Résultats pour des données MOCAP bruitées (a) Résultats pour WZ (b) Résultats par la méthode proposée.*

que l'approche WZ montre des difficultés (courbes verte, rouge et bleue).

Notons donc que l'ajout de bruit sur les trajectoires ce qui pourrait être assimilé à des erreurs de suivi ne vient pas perturber l'estimation du délai temporel. Cependant, de graves erreurs de suivi sur un grand nombre de trajectoires pourraient effectivement dégrader les structures des SSMs et rendre notre méthode moins performante.

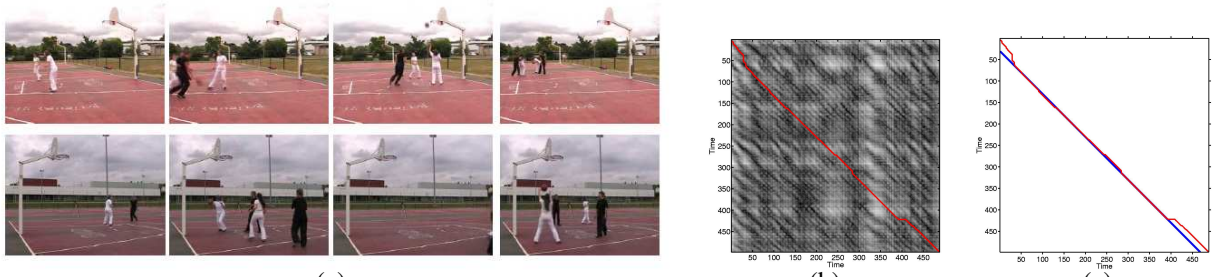
5 Conclusion

Nous avons proposé une approche originale pour synchroniser des vidéos à partir d'auto-similarités temporelles. Elle se caractérise par sa simplicité et sa flexibilité : elle ne repose pas sur des hypothèses restrictives telles que l'existence de correspondances de points entre les vues ou de modélisation de la transformation temporelle. De plus, les matrices d'auto-similarités, qui ne sont pas strictement invariantes aux changements de vue, fournissent des descripteurs robustes pour la synchronisation. La méthode a été testée sur des données multi-vues contrôlées et sur des séquences naturelles. Dans des travaux futurs, nous étudie-



(a) (b) (c)

Figure 7 – *Synchronisation pour des séquences avec des objets mobiles. (a) Deux balles rebondissent sur une table vues de côté et de dessus (b) Matrice de coût avec l'estimation de la transformation (courbe rouge). (c) Cette estimation recouvre partiellement la transformation originale (courbe bleue).*



(a) (b) (c)

Figure 8 – *Synchronisation de séquences de basket-ball avec quatre joueurs. (a) Les séquences représentent deux vues opposées au sein desquelles les joueurs peuvent apparaître et disparaître. (b) Matrice de coût avec l'estimation de la transformation (courbe rouge).(c) Cette estimation retrouve la transformation originale (courbe bleue).*

rons cette méthode dans le cas de caméras mobiles. De plus, comme les structures des matrices d'auto-similarité sont non seulement stables aux changements de vue mais également spécifiques aux actions, la méthode pourrait être utilisée pour synchroniser des actions, c'est-à-dire synchroniser des séquences représentant une même action réalisée par des personnes différentes et sous des points de vue différents.

Références

[1] Y. Caspi et M. Irani. Spatio-temporal alignment of sequences. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 24(11) :1409–1424, November 2002.

[2] L. Wolf et A. Zomet. Wide baseline matching between unsynchronized video sequences. *Int. J. of Computer Vision*, 68(1) :43–52, June 2006.

[3] M. Ushizaki, T. Okatani, et K. Deguchi. Video synchronization based on co-occurrence of appearance changes in video sequences. Dans *Int. Conf. on Pattern Recognition*, pages III : 71–74, 2006.

[4] T. Tuytelaars et L.J. Van Gool. Synchronizing video sequences. Dans *Proc. Conf. Comp. Vision Pattern Rec.*, volume 1, pages 762–768, 2004.

[5] R. Cutler et L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *PAMI*, 22(8) :781–796, 2000.

[6] C. Benabdelkader, R. G. Cutler, et L. S. Davis. Gait recognition using image self-similarity. *EURASIP J. Appl. Signal Process.*, 2004(1) :572–585, January 2004.

[7] E. Shechtman et M. Irani. Matching local self-similarities across images and videos. Dans *Proc. Conf. Comp. Vision Pattern Rec.*, June 2007.

[8] A. Rao, C.and Gritai, M. Shah, et T. F. Syeda Mahmood. View-invariant alignment and matching of video sequences. Dans *Proc. Int. Conf. on Image Processing*, pages 939–945, 2003.

[9] I.N. Junejo, E. Dexter, I. Laptev, et P. Pérez. Cross-view action recognition from temporal self-similarities. Dans *Proc. Eur. Conf. Computer Vision*, October 2008.

[10] N. Dalal et B. Triggs. Histograms of oriented gradients for human detection. Dans *Proc. Conf. Comp. Vision Pattern Rec*, volume 2, pages 886–893, 2005.

[11] B.D. Lucas et T. Kanade. An iterative image registration technique with an application to stereo vision. Dans *Image Understanding Workshop*, pages 121–130, 1981.

[12] L. Rabiner, A. Rosenberg, et S. Levinson. Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26(6) :575– 582, 1978.

[13] D. Weinland, E. Boyer, et R. Ronfard. Action recognition from arbitrary views using 3d exemplars. Dans *Proc. Int.Conf. on Computer Vision*, pages 1–7, 2007.