Appariement de points spatio-temporels par hyper-graphes et optimisation discrète exacte

Oya Celiktutan[a] Christian Wolf[b] Bülent Sankur[a]

[a]Electrical and Electronics Engineering
Bogaziçi University, Istanbul, Turkey
{oya.celiktutan,bulent.sankur}@boun.edu.tr

[b]Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR CNRS 5205, F-69621, France
christian.wolf@liris.cnrs.fr

Résumé

Les graphes et les hyper-graphes sont souvent utilisés pour la reconnaissance de modèles complexes et non-rigides en vision par ordinateur, soit par appariement de graphes ou par appariement de nuages de points par graphes. La plupart des formulations recourent à la minimisation d'une fonction d'énergie difficile contenant des termes géométriques ou structurels, souvent couplés avec des termes d'attache aux données comportant des informations liées à l'apparence locale. Les méthodes traditionnelles tente une résolution approximative du problème de minimisation, par exemple avec des techniques spectrales. Dans cet article nous traitons des données embarquées dans l'« espace-temps », comme cela est typiquement le cas pour les applications de reconnaissance d'actions. Nous montrons que, dans ce contexte, nous pouvons profiter des propriétés particulières du domaine temporel, notamment la causalité et l'ordre stricte imposé par cette dimension. Nous montrons que la complexité du problème est inférieure à la complexité de la problématique générale et nous dérivons un algorithme calculant la solution exacte. Comme une seconde contribution, nous proposons une nouvelle structure graphique allongée dans le temps. Nous soutenons que, au lieu résoudre le problème d'origine de manière approximative, une meilleure solution peut être obtenue par en résolvant, de manière exacte, un problème approché. Un algorithme de minimisation exacte est dérivé de cette structure et appliqué avec succès à la reconnaissance d'actions dans les vidéos.

Mots clefs

Appariement d'hyper-graphes, appariement de nuages de points, optimisation discrète, reconnaissance d'actions

1 Introduction

Dans ce papier nous traitons la reconnaissance automatique de motifs visuels complexes à l'aide d'un exemple concret, à savoir la reconnaissance d'actions dans les vidéos. La littérature sur ce problème est devenue vaste, nous concentrons donc notre bibliographie sur les modèles structurés et semi-structurés. Pour le reste, nous referons le lecteur intéressé à un survey publié récemment [1].

Dans ce contexte, grâce à leur robustesse vis à vis d'occultations, les représentations par points parcimonieux (ou points d'intérêts) ont eu un grand succès dans la communauté. Cette description est structurelle dans la manière où elle consiste en un ensemble de points dont les relations spatiales ou spatio-temporelles sont souvent aussi importantes que les caractéristiques d'apparence associées. La plupart de classifiers nécessitant une description embarquée dans un espace vectoriel, leur usage direct est difficile. Pour palier à ce problème, les modèles *Sacs de mots* (BoW) ont été introduits [2, 3]. Il s'agit d'une modélisation par un histogramme décrivant les fréquences des mots visuels obtenus par un clustering, sans utiliser les relations spatiales ou spatio-temporelles.

D'autre part, les graphes et les hyper-graphes [1] sont une description naturelle pour un ensemble de points d'intérêt, puisque les relations spatio-temporelles sont pris en compte implicitement. Dans ce travail, nous nous concentrons sur l'appariement de hyper-graphes et sur l'appariement de nuages de points par hyper-graphes. La spécificité de l'approche réside dans le fait que les sommets d'un graphe correspondent à des points d'intérêt espace-temps. L'appariement tente à trouver les points d'un modèle parmi les points d'une scène, typiquement beaucoup plus nombreux. Trois articles ont récemment été publiés sur l'appariement de graphes de type espace-temps : dans [5], des hyper-graphes sont construits sur un ensemble de points d'intérêt ; dans [6], des graphes sont construits à partir de tubes spatio-temporelles issue d'une sur-segmentation de la vidéo ; dans [7], des chaînes sont formées, où chaque sommet correspond à un graphe construit à partir d'une petite sous-séquence. Les séquences sont appariées avec une méthode de programmation dynamique impliquant une méthode spectrale pour l'appariement des graphes.

Hors contexte de données spatio-temporelles, le problème d'appariement de graphes a été étudié de manière intensive en vision par ordinateur. Alors que le problème d'isomorphisme de graphes peut être résolu en temps polynomial, l'appariement de sous-graphes est NP-complet [8], tout comme l'isomorphisme de sous-graphes [9]. La mé-

1. Un hyper-graphe est une généralisation d'un graphe, où une hyper-arête peut connecter n'importe quel nombre de sommets, généralement plus de deux [4].

thode dans [4] procède par optimisation convexe d'un problème relâché dans un cadre probabiliste. Récemment, [10] ont généralisé la méthode spectrale de [11] aux l'hyper-graphes en utilisant un algorithme basé sur les tenseurs. Dans [12], une approche de programmation convexe-concave est employée sur un problème de moindres carrés sur les matrices de permutation. Plusieurs méthodes décomposent le problème initial en sous-problèmes qui sont résolus avec des outils d'optimisation discrète comme des *graph cuts* [8, 13]. Dans [14], un algorithme de type *graph cuts* est étendu aux multi-labels et aux 2D en alternant entre les coordonnées x et y. Dans [15], une structure de graphe candidate est créé et le problème est formulé comme un problème de coloration de graphes organisé en plusieurs couches. Une solution pour le problème résultant de programmation d'entier quadratique est proposée dans [16] ; dans [17], le problème est étendu aux relations d'ordre général (> 3) et résolu avec des marches aléatoires.

Dans ce travail nous proposons de profiter de certaines propriétés de l'espace 3D dans lequel les données sont embarquées pour concevoir un algorithme calculant la solution exacte du problème d'appariement.

Le reste de cet article est organisé comme suite. Section 2 formule le problème d'appariement dans notre contexte. Section 3 introduit les propriétés de l'espace concerné et dérive un algorithme d'appariement calculant la solution exacte d'un problème d'optimisation discrète. Section 4 présente une approximation de la structure graphique permettant de calculer l'appariement en complexité linéaire. Section 5 présente les expériences et les résultats et section 6 donne la conclusion.

2 Formulation du problème

Nous formulons le problème comme un cas particulier du problème de la correspondance générale entre deux ensembles de points. L'objectif est d'attribuer, à chaque point d'un ensemble de points d'un modèle, un point de l'ensemble de la scène, tel que certaines invariances géométriques soient satisfaites. Nous résoudrons ce problème par la minimisation d'une fonction d'énergie définie sur un hyper-graphe construit à partir de l'ensemble de points du modèle. Les M points du modèle sont donc organisés comme un hyper-graphe $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, où $\mathcal{V}$ est l'ensemble de sommets (correspondant aux points) et $\mathcal{E}$ est l'ensemble des arêtes. A partir de maintenant nous allons abusivement appeler « graphes »les hyper-graphes et « arêtes », ou « triangles »les hyper-arêtes. Nos arêtes connectent des ensembles de trois sommets, donc des triangles.

Bien que notre méthode suppose que les données de la vidéo du modèle soient structurées en un graphe, cela n'est pas nécessairement demandé pour les données de la vidéo de la scène. Des informations structurelles sur les données de scène *peuvent* être facilement intégrées dans notre formulation, ce qui donne le problème classique d'appariement de graphes. Notre formulation est donc plus générale, mais peut aussi traiter des problèmes d'appariement de graphes.

A chaque sommet i du graphe de modèle est associé une variables discrète x_i, i=$1 \dots M$ qui peut prendre des valeurs de l'ensemble $\{1 \dots S, \epsilon\}$, où S et le nombre de points. La valeur $x_i = j$ signifie que le point i du modèle est apparié au point j de la scène. La valeur $x_i = \epsilon$ signifie que le point i du modèle n'est pas apparié, une possibilité admis pour gérer les occultations. L'ensemble complet de variables x_i est aussi noté x.

A chaque point i des deux ensembles (modèle et scène) et également associé une position spatio-temporelle $p_i = [\ p_i^{<x>}\ p_i^{<y>}\ p_i^{\langle t \rangle}\ ]^T$ et un vecteur de caractéristiques d'apparence f_i. Lorsque nécessaire, nous ferons une distinction entre le modèle et la scène par les exposants : $p_i^{\langle m \rangle}, f_i^{\langle m \rangle}, p_i^{\langle s \rangle}, f_i^{\langle s \rangle}$ etc. Notons que les symboles dans les exposants entourés de chevrons $\langle . \rangle$ ne sont pas des indices numériques ; il s'agit de symboles indiquant une catégorie.

L'appariement est contrôlé par une fonction d'énergie $E(x)$ qui sera petite pour des appariements qui correspondent à une transformation réaliste du modèle à la scène. Traditionnellement, ces fonctions contiennent des termes par paires de sommets reliés par arêtes, vérifiant la similarité des longueurs d'une arête et l'arête appariée. Ces contraintes n'étant pas invariantes à l'échelle, le formalisme a été étendu aux hyper-graphes permettant la vérification des contraintes exprimées sur des triplets de sommets, par exemple en se basent sur des angles. Initialement proposé pour la reconnaissance d'objets [11], nous avons appliqué un raisonnement similaire pour la reconnaissance d'actions [5]. Ici nous proposons une fonction d'énergie similaire :

$$E(x) = \lambda_1 \sum_i U(x_i) + \lambda_2 \sum_{(i,j,k) \in \mathcal{E}} D(x_i, x_j, x_k) \quad (1)$$

où U est un term d'attache aux données tenant compte des distances des caractéristiques d'apparance, D est un terme mesurant la distorsion entre deux triangles et les λ_i sont des poids. Pour rendre plus facile la lecture, nous avons omis toutes les dépendances vers des valeurs sur lesquelles nous ne ferons pas de minimisation.

La distance U est définie comme la distance Euclidienne entre vecteurs de caractéristiques, en tenant compte d'une punition W^P si l'appariement n'a pas lieu :

$$U(x_i) = \begin{cases} W^p & \text{if } x_i = \epsilon \\ ||{f_i}^{\langle m \rangle} - f_{x_i}^{\langle s \rangle}|| & \text{else} \end{cases} \quad (2)$$

La distorsion géométrique est traitée de manière séparée pour l'espace et le temps :

$$D(x_i, x_j, x_k) = D^t(x_i, x_j, x_k) + \lambda_3 D^g(x_i, x_j, x_k) \quad (3)$$

où la distorsion temporelle D^t est définie comme une différence tronquée sur deux paires de sommets d'un triangle :

$$D^t(x_i, x_j, x_k) = \\ = \begin{cases} W^t & \text{if } \Delta(i,j) > T^t \ \vee \Delta(j,k) > T^t \\ \Delta(i,j) + \Delta(j,k) & \text{else} \end{cases} \quad (4)$$

Ici, $\Delta(a,b)$ est la distorsion temporelle d'une paire (a,b) :

$$\Delta(a,b) = |(p_a^{\langle m\rangle\langle t\rangle} - p_b^{\langle m\rangle\langle t\rangle}) - (p_{x_a}^{\langle s\rangle\langle t\rangle} - p_{x_b}^{\langle s\rangle\langle t\rangle})| \quad (5)$$

Le terme D^g est défini sur des différences d'angles :

$$D^g(x_i,x_j,x_k) = \left|\left| \begin{array}{c} \phi^{\langle m\rangle}(i,j,k) - \phi^{\langle s\rangle}(x_i,x_j,x_k) \\ \phi^{\langle m\rangle}(j,i,k) - \phi^{\langle s\rangle}(x_j,x_i,x_k) \end{array} \right|\right| \quad (6)$$

Ici, $\phi^{\langle m\rangle}(a,b,c)$ et $\phi^{\langle s\rangle}(a,b,c)$ sont des angles sur le point b pour, respectivement, les triangles du modèle de la scène indexés par (a,b,c).

3 Appariement

Nous supposons les propriétés suivantes de l'espace-temps, pour dériver un algorithme efficace :

Hypothesis 1 : Causalité — Les dimensions spatiales (x,y) et temporelle (t) ne doivent pas être traitées de la même manière. Pour un appariement correct, l'ordre temporel des points doit rester intact, ce qui peut être formalisé comme suit :

$$\forall\, i,j : p_i^{\langle m\rangle\langle t\rangle} \leq p_j^{\langle m\rangle\langle t\rangle} \Leftrightarrow p_{x_i}^{\langle s\rangle\langle t\rangle} \leq p_{x_j}^{\langle s\rangle\langle t\rangle} \quad (7)$$

Hypothesis 2 : Proximité temporelle — Une autre hypothèse raisonnable limite la quantité de distorsion temporelle entre les deux séquences. En d'autres termes, deux points proches dans le temps dans le modèle doivent être appariés à deux points proches dans le temps dans la scène. En supposant que le graphe du modèle est construit à partir d'informations de proximité, cela peut être formalisé comme suit :

$$\forall\, i,j,k \in \mathcal{E} : |p_{x_i}^{\langle s\rangle\langle t\rangle} - p_{x_j}^{\langle s\rangle\langle t\rangle}| < T^t \vee |p_{x_j}^{\langle s\rangle\langle t\rangle} - p_{x_k}^{\langle s\rangle\langle t\rangle}| < T^t \quad (8)$$

Hypothesis 3 : Unicité des instants temporels — Nous supposons qu'un instant ne peut être divisé ou fusionné. Tous les points d'une frame unique du modèle doivent donc être appariés avec les points d'une seul frame unique également :

$$\forall i,j : \quad (p_i^{\langle m\rangle\langle t\rangle} = p_j^{\langle m\rangle\langle t\rangle}) \quad \Leftrightarrow \quad (p_{x_i}^{\langle s\rangle\langle t\rangle} = p_{x_j}^{\langle s\rangle\langle t\rangle}) \quad (9)$$

Selon l'hypothèse nr. 3, un appariement valide impliquera un appariement des frames du modèles aux frames de la scène. Pour cette raison nous reformulerons la fonction d'énergie (1) en divisant chaque variable x_i en deux variables z_i et $x_{i,l}$ qui seront intérprétées de manière suivante : la valeur de z_i est l'indice de la frame de la scène appariée avec la frame i du modèle. Le nombre de frames du modèle sera noté comme $\overline{M}$. Chaque frame i du modèle comprend un nombre $\overline{M}_i$ de variables $x_{i,1},\ldots,x_{i,M_i}$, où la valeur $x_{i,1}$ donnera l'indice du sommet auquel il sera apparié dans scène. Ces indices seront numérotés en commençant par 1 dans *chaque frame de la scène*.

Afin de rendre la lecture plus facile, nous simplifieront la notation en représentant une hyper-arête (ainsi que les indices de frame et les indices de sommet) par c et les variables correspondant comme z_c et x_c ; nous supprimerons également les poids λ_1 et λ_2 qui peuvent être absorbés dans les potentiels U et D. Cela donne l'équation reformulée :

$$E(z,x) = \sum_{(i,l)\in\overline{M}\times\overline{M}_i} U(z_i,x_{i,l}) + \sum_{c\in\mathcal{E}} D(z_c,x_c) \quad (10)$$

Nous introduisons une décomposition de l'ensemble des arêtes $\mathcal{E}$ en sous-ensembles disjoints $\mathcal{E}^i$, où $\mathcal{E}^i$ est l'ensemble d'arêtes c tel que c contient au moins un sommet avec une coordonnée temporelle égale à i et aucun sommet dans c a une coordonnée temporelle supérieure (donc, plus tard) à i. Il est facile de voir que l'ensemble des $\mathcal{E}^i$ donne un partitionnement complet de $\mathcal{E}$, ç.à.d. que $\mathcal{E} = \bigcup_i \mathcal{E}^i$. Nous pouvons maintenant échanger les sommes et le minima de notre problème selon ce partitionnement :

$$\begin{aligned} \min_{z,x} E(z,x) = \\ \min_{z_1;x_{1,1},\ldots,x_{1,\overline{M}_1}} & \left[\sum_{l=1}^{\overline{M}_1} U(z_1,x_{1,l}) + \sum_{c\in\mathcal{E}^1} D(z_c,x_c) + \right. \\ \min_{z_2;x_{2,1},\ldots,x_{2,\overline{M}_2}} & \left[\sum_{l=1}^{\overline{M}_2} U(z_2,x_{2,l}) + \sum_{c\in\mathcal{E}^2} D(z_c,x_c) + \right. \\ & \vdots \\ \min_{z_{\overline{M}};x_{\overline{M},1},\ldots,x_{\overline{M},\overline{M}_{\overline{M}}}} & \left[\sum_{l=1}^{\overline{M}_{\overline{M}}} U(z_{\overline{M}},x_{\overline{M},l}) + \sum_{c\in\mathcal{E}^{\overline{M}}} D(z_c,x_c)\right] \left.\vdots\right] \end{aligned} \quad (11)$$

Nous introduisons aussi le concept de la *portée* $\mathcal{R}^i$ de la frame i qui, intuitivement, est l'ensemble d'arêtes atteignant le passé de la frame i et qui viennent de i ou de son avenir ($> i$). Plus formellement,

$$\mathcal{R}^i = \left\{ c \in \mathcal{E} : [\min^{\langle t\rangle}(c) < i] \wedge [\max^{\langle t\rangle}(c) \geq i] \right\} \quad (12)$$

où $\min^{\langle t\rangle}(c)$ et $\max^{\langle t\rangle}(c)$ sont, respectivement, la coordonnée minimale et maximale des sommets de l'arête c. Notons que $\mathcal{E}^i \subseteq \mathcal{R}^i$. L'expression $\mathcal{X}^i$ dénotera l'ensemble de toutes les variables z_i et $x_{i,j}$ impliquées dans les arêtes de la portée $\mathcal{R}^i$:

$$\begin{array}{rl} \mathcal{X}^i = & \left\{ z_j : \exists k : (j,k) \in c \wedge c \in \mathcal{R}^i \right\} \cup \\ & \left\{ x_{j,k} : (j,k) \in c \wedge c \in \mathcal{R}^i \right\} \end{array} \quad (13)$$

Enfin, les variables $\mathcal{R}^i$ restreints aux variables des frames *avant* la frame i seront dénotés comme $\mathcal{X}^{i-}$:

$$\mathcal{X}^{i-} = \{ z_j, x_{j,k} \in \mathcal{X}^i : j < i \} \quad (14)$$

Le schéma de calcul récursive minimisant (11) peut maintenant être dérivé en définissant une variable récursive α_i qui minimise les variables d'une frame donnée étant donné les valeurs optimales pour les variables de sa portée :

$$\begin{aligned} \alpha_i(\mathcal{X}^{i-}) \quad = \quad & \min_{z_i;x_{i,1},\ldots,x_{i,\overline{M}_i}} \left[\sum_{l=1}^{N_i} U(z_i,x_{i,l}) + \right. \\ & + \quad \left. \sum_{c\in\mathcal{E}^i} D(z_c,x_c) + \alpha_{i+1}(\mathcal{X}^{(i+1)^-}) \right] \end{aligned} \quad (15)$$

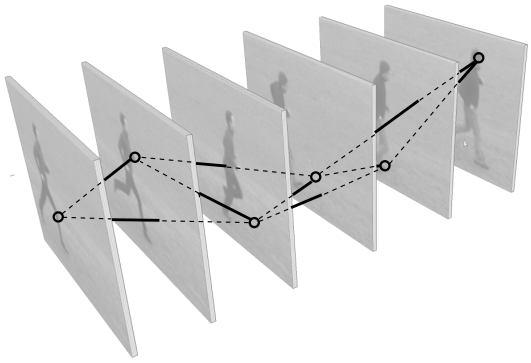

Figure 1 – *Une structure graphique pour le modèle, construit pour une faible complexité de calcul. Aucune structure est imposée sur les points de la scène.*

Cela est possible grâce à la relation suivante : $\mathcal{X}^{(i+1)^-} \subseteq (\mathcal{X}^{i-} \cup z_i; x_{i,1}, \ldots, x_{i,\overline{M}_i})$.
Le calcul est initié pour la dernière frame $i = \overline{M}$ avant d'itérer en calculant α_i à partir de α_{i+1}. La complexité de calcul dépend du nombre de variables dans la portée $\mathcal{R}^i$ et des tailles des domaines de ces variables :

$$O\left(\max_i\left[\prod_{v\in\mathcal{V}^i}|\text{domain}(v)|\right]\right) \approx O\left(\max_i\left[\overline{S}^{|\mathcal{X}_z^i|}\langle\langle s\rangle\rangle^{|\mathcal{X}_x^i|}\right]\right) \tag{16}$$

où $\overline{S}$ est le nombre de frames de la scène, $\langle\langle s\rangle\rangle$ est le nombre moyen de sommets par frame dans le modèle et $|\mathcal{X}_z^i|$ est le nombre de variables de l'ensemble z dans $\mathcal{X}^i$. La complexité est donc très inférieure à la complexité de l'approche directe, donnée par $O(S^M M|\mathcal{E}|)$. Notons que S est le nombre total de sommets dans la scène et M est le nombre total de sommets du modèle ; surtout, $S \gg \overline{S}$ et $S \gg \langle\langle s\rangle\rangle$. Aussi, $|\mathcal{X}_z^i|$ et $|\mathcal{X}_x^i|$ sont bornés et faible si le graphe est construit à partir d'informations de proximité. Par contre, en pratique la complexité reste élevée. Dans la section suivante nous introduirons une structure graphique spécifique pour rendre l'algorithme encore plus efficace.

4 Approximations

La plupart des formulations du problème d'appariement de graphes ou d'appariement à partir d'un graphe sont NP-complets. Les méthodes classiques résolvent ce problème en calculant une solution approchée. Dans ce travail, nous préconisons d'approximer la structure graphique et de résoudre le nouveau problème de manière exacte. Cette stratégie est particulièrement attrayante dans le cas de problèmes d'appariement où la structure du graphe est moins liée à la description de l'objet, mais plutôt aux contraintes du processus d'appariement. Nous rappelons que la structure graphique est obtenue à partir d'informations d'adjacence ou de proximité. Elle est donc déduite des attributs des sommets (les positions), la changer ne nuira pas sensiblement à la description de l'objet spatio-temporel. Une philosophie similaire a été mis en avant par [18] dans le contexte de la reconnaissance d'objets, où l'objet spatial est structuré en un k-arbre, ce qui rends peu complexe l'algorithme *junction tree* utilisé pour la minimisation de la fonction d'énergie.
Nous proposons de structurer les points du modèle comme suit :

– Nous gardons un point unique pour chaque frame du modèle en choisissant le point le plus saillant, d'après le détecteur de point d'intérêt. Aucune restriction est appliquée pour la scène, chaque frame de la scène peut contenir autant points que souhaité.
– Chaque point i du modèle est connecté à ces prédécesseurs immédiats $i-1$ et $i-2$ ainsi que à ses successeurs immédiats $i+1$ et $i+2$.

Cela donne un graphe planaire avec une structure triangulaire comme illustré dans la figure 1. Le cas général de l'énergie (1) peut être simplifié pour tenir compte de cette structure. La division des variables en paires (z_i, x_i), introduite dans la section 3, n'est plus nécessaire. Le système de voisinage peut être décrit de manière très simple en se basant sur les indices des variables x_i :

$$E(x) = \sum_{i=1}^{M} U(x_i) + \sum_{i=3}^{M} D(x_i, x_{i-1}, x_{i-2}) \tag{17}$$

La portée de cette structure est constante, elle consiste de deux arêtes : $\mathcal{R}^i = \{(x_{i-2}, x_{i-1}, x_i), (x_{i-1}, x_i, x_{i+1})\}$; L'ensemble des variables de la portée est également constant : $\mathcal{X}^i = \{x_{i-2}, x_{i-1}, x_i\}$. La récursion peut être donnée par l'équation suivante :

$$\alpha_i(x_{i-1}, x_{i-2}) = \min_{x_i}\Big[\; U(x_i) + D(x_{i-2}, x_{i-1}, x_i) + \alpha_{i+1}(x_i, x_{i-1})\;\Big] \tag{18}$$

Durant le calcul du treillis, les arguments des minima de l'équation (18) sont retenus dans la table $\beta_i(x_{i-1}, x_{i-2})$. Une fois le treillis complété, l'appariement optimal peut être trouvé par un *backtracking* classique (en partant d'une recherche initiale pour les valeurs de x_1 et x_2) :

$$\widehat{x_i} = \beta_i(x(i-1), x(i-2)), \tag{19}$$

L'algorithme sa la forme donnée ci-dessus est d'une complexité de calcul de $O(M{\cdot}S^3)$. En profitant des différentes hypothèses introduits dans la section 2, la complexité peut encore être diminuée :

Ad) Hypothesis 1 — pour une valeur donnée de x_i, les valeurs des prédécesseurs x_{i-1} et x_{i-2} doivent être *avant* x_i, ç.à.d. inférieures.

Ad) Hypothesis 2 — de manière similaire, les valeurs de x_{i-1}, x_{i-2} sont supposées être proches, ç.à.d. la distance doit être inférieure à T^t.

Cela réduit la complexité à $O(M{\cdot}S{\cdot}{T^t}^2)$, où T^t est une constante (autour de 15) ; la complexité est donc linéaire en fonction du nombre de frames de la scène : $O(M{\cdot}S)$.

	B	HC	HW	J	R	W
B	**100**	0	0	0	0	0
HC	0	**100**	0	0	0	0
HW	3	26	**71**	0	0	0
J	0	0	0	**69**	31	0
R	0	0	0	25	**75**	0
W	0	0	0	3	3	**94**

(a)

	B	HC	HW	J	R	W
B	**100**	0	0	0	0	0
H	3	**97**	0	0	0	0
H	6	15	**79**	0	0	0
J	0	0	0	**72**	28	0
R	0	0	0	8	**89**	3
W	0	0	0	6	0	**100**

(b)

Tableau 1 – *La matrice de confusion avec (a) et sans (b) sélection de modèles. Les taux de reconnaissance respectifs :* 84.8%*,* 89.3%*.*

Method	B	HC	HW	J	R	W	Tot.
Laptev *et al.* [19]	97	95	91	89	80	99	91.8
Schuldt *et al.* [20]	98	60	74	60	55	84	71.8
Li *et al.* [21]	97	94	86	**100**	83	97	**92.8**
Niebles *et al.* [22]	99	**97**	**100**	78	80	94	91.3
Our method	**100**	**97**	79	72	**88**	**100**	89.3

Tableau 2 – *Comparaison avec des méthodes existantes utilisant le même protocole et la même base de vidéos (KTH).*

5 Expériences

Nous avons testé la méthode proposée sur la base publique KTH [20] qui comprend 25 personnes effectuant 6 actions (*walking*, *jogging*, *running*, *handwaving*, *handclapping* et *boxing*) enregistrées dans quatre différentes scénarios. Nous utilisons le même protocole que dans le document original [20]. Tout d'abord, nous construisons un dictionnaire de modèles composé de séquences extraites des ensembles d'apprentissage et de validation. Nous utilisons 383 séquences pour l'apprentissage et 216 pour le test. Un totale de 1429 graphes de longueurs de 20 à 30 sommets sont créés pour servir comme modèles.

Les points d'intérêts spatio-temporelle sont extraits avec le détecteur Harris 3D [19]. Comme descripteurs d'apparence et de mouvement f_i nous avons choisi les HoG/HoF (*Histograms of oriented gradients*, *Histograms of oriented flow*), largement utilisée dans la littérature [19]. Comme mentionné dans la section 4, nous avons sélectionné un seul point par image modèle, tous les points ont été conservés pour les vidéos de test.

Les paramètres ont été fixés comme suit. La pénalité W_P devrait théoriquement être plus élevé que la moyenne d'énergie locale des triangles correctement appariés et inférieure à la moyenne d'énergie locale des triangles incorrectement appariés. Nous l'estimons en échantillonnant des énergies pour ces deux cas et en mettant $W^P = 8,4$ comme la valeur minimisant l'erreur de Bayes. Les paramètres λ_i ont été optimisées par recherche de grille sur l'ensemble de validation : $\lambda_1 = 0.6$, $\lambda_2 = 0.1$, $\lambda_3 = 10$, $T^t = 30$ et $W^t = 60$.

Les classes des vidéos de la base de test sont reconnues avec un classifier « plus proche voisin »avec comme distance l'énergie (1). Le taux de reconnaissance sans traitement supplémentaire est de 84.8%.

Apprentissage d'un dictionnaire — un dictionnaire optimal et équilibré a été obtenu avec *Sequential Floating Backward Search* (SFBS), une procédure qui supprime des modèles non pertinents de l'ensemble d'apprentissage avec une approche gloutonne [23]. Une moitié de l'ensemble d'apprentissage est utilisée comme ensemble de validation durant cet étape. La sélection d'un ensemble de 44 modèle permet d'améliorer la performance à 89.3%.

Dans la figure 2 quelques exemples d'appariement sont donnés : les premiers deux cas sont des exemples pour un appariement correct ; le 4^e cas est un exemple pour un appariement incorrect. La table 2 montre que la performance de notre méthode se compare avec les méthodes de l'état de l'art, tout en étant plus que compétitive en terme de calcul. Nous voudrons souligner que de nombreux résultats ont été publiés sur la base de données KTH. Par contre, les protocoles ne sont pas comparables pour la plupart d'entre eux — voir l'excellente comparaison des protocoles dans [24]. Dans table 2, nous avons choisi des résultats obtenus avec le même protocole (le protocole d'origine [20]).

La méthode a été implémentée en open-CL sur une carte graphique Nvidea GForce GTX560 avec 336 cores cuda. L'appariement de 44 modèles dans un bloc de vidéo de 55 frames nécessite $190ms$, soit $3,4ms$ par frame. Le calcul est donc bien plus rapide que le temps réel.

6 Conclusion

Dans cet article nous avons montré que la solution exacte du problème d'appariement à l'aide d'un hyper-graphes peut être calculée de manière efficace, lorsque les données sont embarquées dans l'espace-temps. Plus précisément, la complexité est exponentielle sur un petit nombre, qui est bornée lorsque le hyper-graphe est structuré avec des informations de proximité. Comme une seconde contribution, nous avons présenté une structure de graphe spécifique permettant de calculer l'appariement exact avec une complexité très faible, linéaire dans le nombre des sommets du modèle et le nombre de sommets de la scène. La méthode a été testée sur la base KTH où elle montre une très bonne performance. Les travaux futurs seront concentrés sur une implémentation GPGPU et sur la modélisation des activités plus complexes avec des modèles hiérarchiques.

Références

[1] J. K. Aggarwal et M. S. Ryoo. Human activity analysis : a review. *ACM Computing Surveys*, 2011.

[2] J. Sivic et A. Zisserman. Video google : a text retrieval approach to object matching in videos. Dans *ICCV*, volume 2, pages 1470–1477, 2003.

[3] I. Laptev et T. Lindeberg. Space-time interest points. Dans *ICCV*, pages 432–439, 2003.

[4] Ron Zass et Amnon Shashua. Probabilistic graph and hypergraph matching. Dans *CVPR*, 2008.

[5] A. P. Ta, C. Wolf, G. Lavoue, et A. Başkurt. Recognizing and localizing individual activities through graph matching. Dans *AVSS*, 2010.

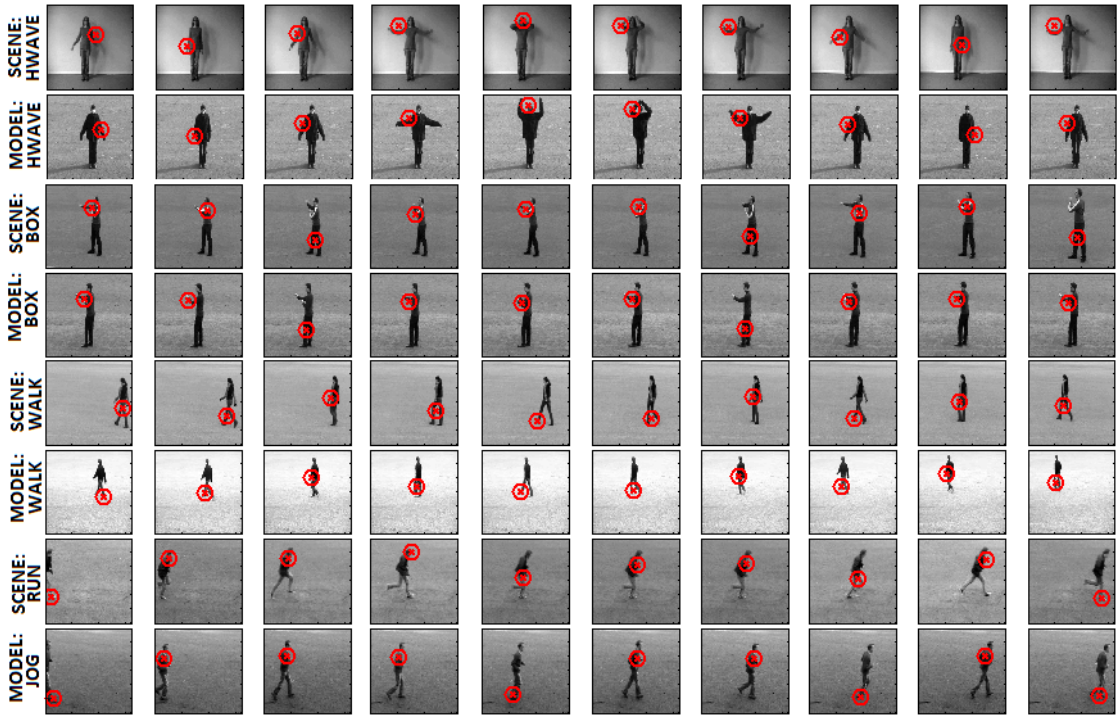

Figure 2 – *Exemples d'appariements : en haut trois appariements corrects ; le dernier appariement en bas est incorrect.*

[6] W. Brendel et S. Todorovic. Learning spatiotemporal graphs of human activities. Dans *ICPR*, 2011.

[7] U. Gaur, Y. Zhu, B. Song, et A. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. Dans *International Conference on Computer Vision*, 2011.

[8] Lorenzo Torresani, Vladimir Kolmogorov, et Carsten Rother. Feature correspondence via graph matching : Models and global optimization. Dans *ECCV*, pages 596–609, 2008.

[9] S. Zampelli1, Y. Deville, et C. Solnon. Solving subgraph isomorphism problems with constraint programming. *Constraints*, 2009.

[10] Olivier Duchenne, Francis R. Bach, In-So Kweon, et Jean Ponce. A tensor-based algorithm for high-order graph matching. Dans *CVPR*, pages 1980–1987, 2009.

[11] Marius Leordeanu et Martial Hebert. A spectral technique for correspondence problems using pairwise constraints. Dans *ICCV*, pages 1482–1489, Washington, DC, USA, 2005.

[12] M. Zaslavskiy, F. Bach, et J.P. Vert. A path following algorithm for the graph matching problem. *IEEE Tr. on PAMI*, 31(12) :2227–2242, 2009.

[13] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, et N. Paragios. Dense non-rigid surface registration using high-order graph matching. Dans *CVPR*, 2010.

[14] O. Duchenne, A. Joulin, et J. Ponce. A graph-matching kernel for object categorization. Dans *ICCV*, 2011.

[15] L. Lin, K. Zeng, X. Liu, et S.-C. Zhu. Layered graph matching by composite cluster sampling with collaborative and competitive interactions. *CVPR*, 0 :1351–1358, 2009.

[16] M. Leordeanu, A. Zanfir, et C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. Dans *ICCV 2011*, 2011.

[17] J. Lee, M. Cho, et K.M. Lee. Hyper-graph matching via reweighted random walks. Dans *CVPR X*, 2011.

[18] T.S.Caetano, T. Caelli, D. Schuurmans, et D.A.C. Barone. Graphical models and point pattern matching. *IEEE Tr. on PAMI*, 28(10) :1646–1663, 2006.

[19] I. Laptev, M. Marszalek, C. Schmid, et B. Rozenfeld. Learning realistic human actions from movies. Dans *CVPR*, pages 1–8, 2008.

[20] C. Schuldt, I. Laptev, et B. Caputo. Recognizing human actions : a local svm approach. Dans *ICPR*, pages 32–36, 2004.

[21] B. Li, M. Ayazoğlu, T. Mao, O. I. Camps, et M. Sznaier. Activity recognition using dynamic subspace angles. Dans *CVPR*, 2011.

[22] J. C. Niebles, C. W. Chen, et L. Fei-Fei. Modelling temporal sturcture of decomposable motion segments for activity classification. Dans *ECCV*, pages 1–14, 2010.

[23] P. Pudil, F. J. Ferri, J. Novovicov, et J. Kittler. Floating search methods for feature selection with non-monotonic criterion functions. Dans *ICPR*, pages 279–283, 1994.

[24] Z. Gao, M.Y. Chen, A. Hauptmann, et A.Cai. Comparing evaluation protocols on the kth dataset. Dans *Human Behavior Understanding*, volume LNCS 6219, pages 88–100, 2010.