# Reconnaissance d'actions 3D par codage parcimonieux sur l'espace de Kendall *

Amor Ben Tanfous Hassen Drira Boulbaba Ben Amor
IMT Lille Douai, Université de Lille, CNRS, UMR 9189 – CRIStAL –
Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

omar.bentanfous@imt-lille-douai.fr

## Résumé

*Les représentations de formes ainsi que leur évolution temporelle, appelées trajectoires, reposent souvent sur des variétés non-linéaires. Ceci engendre une contrainte supplémentaire pour l'usage des techniques d'apprentissage automatique conventionnelles. Cet article adapte la méthode classique de codage parcimonieux et apprentissage de dictionnaire à l'espace de formes de Kendall et illustre son utilisation sur la reconnaissance d'actions 3D à partir de données squelettiques. En se basant sur la géométrie Riemannienne de l'espace des formes, une formulation intrinsèque du codage parcimonieux est d'abord proposée pour des formes statiques afin de contourner le problème de non-linéarité de l'espace des formes. Les trajectoires sont ainsi représentées dans un espace vectoriel par des séries temporelles parcimonieuses. Enfin, nous adoptons deux techniques de classification : un LSTM bidirectionnel appliqué directement sur ces séries temporelles et un SVM linéaire précédé par l'application d'une pyramide temporelle de Fourier sur celles-ci. Les expérimentations effectuées sur trois bases de données montrent la supériorité de l'approche proposée par rapport aux approches Riemanniennes de la littérature, et sa compétitivité vis-à-vis des autres approches récentes.*

## Mots Clef

Espace de Kendall, analyse de formes, codage parcimonieux et apprentissage de dictionnaire, reconnaissance d'actions 3D, apprentissage automatique.

# 1 Introduction

La démocratisation des capteurs de profondeurs et d'algorithmes d'estimation de marqueurs (position des articulations) du corps humain [28] a suscité l'intérêt des chercheurs en vision par ordinateur pour étudier les formes squelettiques ainsi que leurs dynamiques dans le temps. En particulier, le problème de modélisation et de reconnaissance d'actions 3D a reçu une attention particulière au cours des dernières années, notamment avec la publication de bases de données d'actions et l'importance des applications comme les jeux vidéos, l'interface homme machine/robot, l'étude de données cinématiques afin d'évaluer les performances physiques, etc. Cependant, les actions humaines observées à partir de ces capteurs sont sujettes à des variations dues au facteur d'échelle, la translation et l'orientation du corps. Une solution évidente serait d'analyser les mouvements du squelette 3D en prenant en compte des propriétés d'invariances géométriques, donnant lieu à des représentations de formes qui vivent souvent dans des espaces non linéaires [4, 20]. Par exemple, Kendall [20] définit la forme comme étant l'information géométrique qui persiste lorsque la position, le facteur d'échelle et les effets de rotation sont filtrés d'un objet. En conséquence, il serait intéressant de représenter les séquences squelettiques 3D par des trajectoires de formes, tel qu'il a été proposé dans [2]. Toutefois, inférer une telle représentation reste un problème difficile vu la non-linéarité de l'espace des formes. En effet, l'application des techniques de codage standards (ACP, codage parcimonieux, etc.) et des techniques d'apprentissage automatique (SVM, apprentissage de dictionnaires, apprentissage profond, etc.) n'est pas évidente. Le problème est encore plus aigu avec l'introduction de la dimension temporelle, c'est-à-dire, l'analyse des trajectoires dans l'espace des formes de Kendall. Dans la littérature, deux alternatives ont été proposées pour surmonter ces problèmes pour différentes variétés Riemanniennes : elles sont soit *Extrinsèques* [14, 17, 22], soit *Intrinsèques* [5, 18, 19]. Alors que la première famille est basée sur la projection dans des espaces de Hilbert (vectoriels) de dimension élevée, la seconde opte pour la projection des points de la variété dans un espace tangent fixe, attaché à la variété en un point de référence. Dans la deuxième famille, le problème principal consiste en l'introduction de distorsions lorsque les points ne sont pas proches du point de référence et lors du calcul des distances entre les différents points sur l'espace tangent [1, 2, 30]. Dans ce travail, nous proposons une solution intrinsèque au problème de non-linéarité de l'espace des formes tout en évitant de projeter les points dans un espace tangent commun.

Motivés par le succès des représentations parcimonieuses dans plusieurs contextes de reconnaissance [6, 10, 14, 18], nous proposons une formulation intrinsèque du codage parcimonieux et d'apprentissage de dictionnaire (SCDL) de formes squelettiques dans l'espace de Kendall. Plus préci-

*Ce travail est accepté pour publication dans la conférence IEEE CVPR'2018.

sément, une forme squelettique est codée sur l'espace tangent lui est attaché, ce qui nous évite la projection vers un espace tangent commun. Ainsi, pour chaque trajectoire, cette représentation donne lieu à une fonction de codes parcimonieux. En outre, nous proposons d'apprendre un dictionnaire en respectant la géométrie Riemannienne de l'espace des formes. Dans le contexte de la reconnaissance d'actions, notre approche apporte deux avantages principaux : (1) Le codage des formes squelettiques est effectué par rapport à un dictionnaire Riemannien. Par conséquent, les fonctions de codes résultantes devraient être plus discriminantes que les données elles-mêmes [18] ; (2) L'utilisation des séquences de codes comme descripteurs nous permet d'effectuer la classification dans un espace vectoriel, évitant la tâche plus difficile de classification sur la variété. Les **contributions de ce travail** sont, (i) Une nouvelle représentation d'actions basée sur du codage parcimonieux de trajectoires de formes sur l'espace de Kendall. Ceci permet de projeter les trajectoires de formes d'un espace non-linéaire en des séries temporelles dans un espace Euclidien ; (ii) Une classification des fonctions de codes avec deux schémas de classification différents. Des expériences sont menées sur trois bases de données communément utilisées pour montrer la compétitivité de l'approche proposée dans le contexte de la reconnaissance d'actions.

Le reste de l'article est organisé comme suit. Dans la section 2, nous décrivons les solutions existantes pour le SCDL sur des variétés non-linéaires, en plus des travaux récents sur la reconnaissance d'actions 3D, avec un accent particulier sur les méthodes Riemanniennes. La section 3 introduit la méthode proposée de SCDL de formes statiques sur l'espace de Kendall. Nous présentons dans la section 4 la modélisation temporelle des séquences d'actions ainsi que les deux schémas de classification adoptés. Les résultats expérimentaux et les discussions sont décrits dans la section 5, et la section 6 conclut l'article.

# 2 Etat de l'art

Dans cette section, nous nous intéressons, tout d'abord, aux méthodes de SCDL dans les variétés Riemanniennes. Ensuite, nous décrivons brièvement quelques approches récentes de reconnaissance d'actions 3D.

## 2.1 SCDL sur les variétés riemanniennes

Les approches basées sur les représentations parcimonieuses ont montré leur efficacité dans plusieurs applications de la vision par ordinateur [10]. Cela a motivé plusieurs recherches récentes dans ce domaine à s'intéresser à ce type de représentations [6, 14, 18]. Dans ce type d'approches, l'idée est de représenter chaque élément de données par un vecteur de codes parcimonieux, en l'exprimant comme étant une combinaison linéaire de quelques atomes d'un dictionnaire. Cela suppose que les données ainsi que les atomes sont définis dans un espace Euclidien (pour pouvoir appliquer une combinaison linéaire d'atomes). Cependant, les descripteurs d'images les plus pertinents vivent souvent dans des variétés non-linéaires [25]. Ainsi, pour coder parcimonieusement ces données tout en exploitant les propriétés d'invariance des représentations Riemanniennes, le problème de SCDL classique doit être étendu au cas non-linéaire. Certains travaux ont abordé ce problème [6, 14, 15, 18, 22, 39]. Une solution directe a été proposé dans [37] en projetant les éléments de la variété vers un espace tangent fixé, qui est attaché à la variété à un point de référence. Néanmoins, cette solution n'exploite pas entièrement la structure Riemannienne de la variété. En effet, dans cet espace tangent, seules les distances par rapport au point de référence correspondent aux vraies distances géodésiques. Afin de surmonter cette limitation, Ho et al. [18] ont proposé un cadre général pour le SCDL sur les variétés Riemanniennes en exploitant le fibré tangent à la variété. Chaque point est codé sur son espace tangent sur lequel les atomes sont projetés. Ainsi, les propriétés intrinsèques de la variété Riemannienne sont considérées. Leur méthode inclut une mise à jour itérative des atomes du dictionnaire en utilisant une descente de gradient le long de la géodésique. Cette solution repose principalement sur la projection dans des espaces tangents en utilisant l'opérateur de projection logarithmique. Toutefois, pour certaines variétés, cet opérateur n'admet pas d'expression analytique explicite ou il est difficile à calculer. Cela a motivé certaines recherches [14, 15, 17, 22] à définir des noyaux permettant de projeter la variété Riemannienne en question dans des espaces de Hilbert de plus grande dimension. Ces derniers sont des espaces linéaires où le SCDL devient possible. Récemment, Harandi et al. [14] ont proposé de projeter la variété de Grassmann dans l'espace des matrices symétriques. Ils ont aussi proposé des versions de l'algorithme de SCDL basées sur des noyaux pour remédier au problème de non-linéarité, comme il a été appliqué dans [16] pour les matrices symétriques définies positives.

## 2.2 Reconnaissance d'actions à partir de séquences de squelettes 3D

Certaines approches récentes utilisent les états pour la modélisation temporelle afin de classifier les séquences d'actions sans tenir compte de la non-linéarité des données. En considérant les actions comme des transitions entre les poses, G. Hernando et al. [11] ont proposé un classifieur basé sur les forêts d'arbres de classification pour discriminer la pose statique et les transitions temporelles entre les trames indépendantes. Dans [35], l'action est modélisée comme un ensemble d'éléments sémantiques dits *motionlets* obtenus en suivant et segmentant la trajectoire de chaque joint. Cette modélisation a été combinée à la corrélation spatio-temporelle pour obtenir un graphe complet, non orienté labellisé qui représente une vidéo. Ils ont enfin proposé un noyau adapté pour mesurer la similarité entre les graphes, et classifier ainsi les vidéos. Plus récemment, deux représentations tensorielles basées sur les noyaux appelées *Sequence Compatibility Kernel* (SCK) et *Dynamics Compatibility Kernel* (DCK) ont été introduites dans [21].

Elles sont capables de capturer une relation d'ordre élevé entre les joints. La première représentation permet de capturer la compatibilité spatio-temporelle des joints entre les deux séquences comparées tandis que la deuxième modélise une séquence dynamique par des co-occurences spatio-temporelles des joints. Les tenseurs sont ensuite formés à partir des noyaux pour apprendre un SVM. Par ailleurs, les réseaux de neurones récurrents (RNN) ont montré des performances prometteuses pour la reconnaissance d'actions 3D. Dans [9], les auteurs ont proposé un RNN bidirectionnel (HBRNN-L) en divisant le squelette en cinq segments constitués de joints voisins. Ensuite, chacun est passé à l'entrée d'un RNN bidirectionnel avant de fusionner leurs sorties pour former les parties supérieures et inférieures du corps. Plus récemment, le *spatio-temporal LSTM* (ST-LSTM) [24] a été proposé comme une version adaptée aux domaines spatio-temporels. Pour cela, l'analyse d'un joint d'un squelette 3D considère l'information spatiale issue des joints voisins et l'information temporelle issue des trames précédentes. En outre, une méthode basée sur la structure d'arbre permet de mieux décrire la propriété d'adjacence entre les joints. Finalement, un mécanisme de *gating* est utilisé pour gérer le bruit et les occultations.

D'autres approches ont exploité les outils de la géométrie Riemannienne pour analyser les séquences de squelettes. Dans [30], les auteurs ont proposé de représenter les mouvements des squelettes par des trajectoires dans le groupe de Lie $SE(3)^n$. Ces trajectoires ont été ensuite projetées dans l'algèbre de Lie $\mathfrak{se}(3)^n$ (l'espace tangent à la variété à l'identité) qui est un espace vectoriel, où elles ont été alignées temporellement puis classifiées. Cette représentation a été adaptée par Anirudh et al. [1] qui ont utilisé la représentation TSRVF [29] pour encoder les trajectoires qui vivent dans les groupes de Lie. Ils ont étendu des méthodes de codage existantes, telles que l'analyse en composante principale (ACP) et KSVD, à ces trajectoires Riemanniennes. Une différente approche [2] a étendu la théorie de l'espace des formes de Kendall au cas des trajectoires. Ici, une métrique élastique basée sur la représentation TSRVF a été définie pour aligner et comparer les trajectoires. Cette étape est effectuée en transportant toutes les trajectoires sur l'espace tangent à la variété en un point de référence. La limitation commune de ces approches Riemannienne consiste en la *projection* des trajectoires dans un espace tangent en un point fixé. Une amélioration a été proposée dans [31] en projetant les trajectoires du groupe de Lie différemment, en combinant l'opérateur de projection logarithmique usuel avec une application de *rolling* qui permet une meilleure représentation des trajectoires sur l'espace tangent. Dans ce papier, nous proposons, dans un premier temps, de représenter les actions par des trajectoires dans l'espace de Kendall comme proposé dans [2]. Afin de remédier au problème de non-linéarité de cette variété, nous proposons une solution intrinsèque pour coder ces trajectoires tout en évitant de les projeter dans un espace tangent de référence, contrairement à [1, 2, 30].

# 3 Codage de formes squelettiques dans l'espace de Kendall

Nous proposons d'adapter une formulation générale du SCDL au cas particulier de l'espace des formes de Kendall. Ceci nous permet de représenter chaque forme squelettique vivant sur la variété de Kendall comme étant un vecteur de codes parcimonieux calculé par rapport à un dictionnaire de formes. Dans ce qui suit, nous commençons par rappeler certaines propriétés géométriques de l'espace de Kendall avant de détailler la méthode de SCDL.

## 3.1 Géométrie de l'espace de Kendall

Un squelette est représenté par un nombre fini de points saillants qu'on appelle des marqueurs (points dans $\mathbb{R}^3$). Pour quantifier les formes, Kendall [20] a proposé d'établir des équivalences par rapport aux transformations qui préservent la forme qui sont *les translations*, *les rotations* et *l'échelle globale* des configurations. Suivant [2], soit $Z \in \mathbb{R}^{n\times 3}$ une forme squelettique, i.e., une configuration de $n$ marqueurs dans $\mathbb{R}^3$. Pour éliminer les variations en translation, nous suivons [8] qui introduit la notion de sous-matrice de Helmert, une sous-matrice de taille $(n-1)\times n$ d'une matrice de Helmert couramment utilisée pour centrer les configurations. Pour tout $Z \in \mathbb{R}^{n\times 3}$, le produit $HZ \in \mathbb{R}^{(n-1)\times 3}$ représente les coordonnées Euclidiennes de la configuration centrée. Soit $\mathcal{C}_0$ l'ensemble de toutes les configurations centrées de $n$ marqueurs dans $\mathbb{R}^3$, i.e., $\mathcal{C}_0 = \{HZ \in \mathbb{R}^{(n-1)\times 3} | Z \in \mathbb{R}^{n\times 3}\}$. $\mathcal{C}_0$ est un espace vectoriel de dimension $3(n-1)$ qui peut être identifié par $\mathbb{R}^{3(n-1)}$. Pour éliminer la variabilité de l'échelle, on définit l'espace de pré-formes par : $\mathcal{C} = \{Z \in \mathcal{C}_0 | \|Z\|_F = 1\}$; $\mathcal{C}$ est la sphère unité dans $\mathbb{R}^{3(n-1)}$ qui est donc de dimension $(3n-4)$. L'espace tangent en toute pré-forme $Z$ est donné par : $T_Z(\mathcal{C}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0\}$. Pour éliminer la variabilité des rotations, pour tout $Z \in \mathcal{C}$, on définit la classe d'équivalence : $\bar{Z} = \{ZO | O \in SO(3)\}$ qui représente toutes les rotations d'une configuration $Z$. L'ensemble de toutes ces classes d'équivalence, $\mathcal{S} = \{\bar{Z} | Z \in \mathcal{C}\} = \mathcal{C}/SO(3)$ est appelé *l'espace des formes*. L'espace tangent en toute forme $\bar{Z}$ est $T_{\bar{Z}}(\mathcal{S}) = \{V \in \mathcal{C}_0 | \text{trace}(V^T Z) = 0, \ \text{trace}(V^T ZU) = 0\}$ , où $U$ est toute matrice antisymétrique de taille $3\times 3$. La première condition fait que $V$ soit tangent à $\mathcal{C}$ et la deuxième implique que $V$ soit perpendiculaire à l'orbite de rotation. Ensemble, elles forcent $V$ à être tangent à l'espace des formes $\mathcal{S}$. En considérant la structure sphérique de $\mathcal{C}$, des expressions analytiques des projections exponentielle et logarithmique sont définies [8, 20] et peuvent être facilement adaptées à $\mathcal{S}$. En résumé, nous disposons d'expression analytiques pour calculer les géodésiques, la projection exponentielle et la projection logarithmique (voir [2] pour plus de détails). En considérant la métrique Riemannienne standard dans $\mathcal{S}$, la géodésique entre deux points $\bar{Z}_1, \bar{Z}_2 \in \mathcal{S}$

est définie par :

$$\alpha(t) = \frac{1}{\sin(\theta)}(\sin((1-t)\theta)Z_1 + \sin(t\theta)Z_2O^*), \quad (1)$$

où $\theta = \cos^{-1}(\langle Z_1, Z_2O^* \rangle)$ et $O^*$ est la rotation optimale qui aligne $Z_2$ avec $Z_1$ : $O^* = argmin_{O \in SO(3)} \|Z_1 - Z_2O\|_F^2$. Ce $\theta$ est aussi la distance géodésique entre $\bar{Z}_1$ et $\bar{Z}_2$ dans l'espace des formes $\mathcal{S}$, et qui représente la déformation optimale de $\bar{Z}_1$ pour arriver à $\bar{Z}_2$. Pour $t = 0$, $\alpha(0) = \bar{Z}_1$ et pour $t = 1$, nous avons $\alpha(1) = \bar{Z}_2$.
Noter que l'espace des formes de Kendall est une variété Riemannienne complète tel que $\log_{\bar{Z}}$ est défini pour tout $\bar{Z} \in \mathcal{S}$. Par conséquent, la distance géodésique entre deux configurations $\bar{Z}_1$ et $\bar{Z}_2$ peut être calculée par $\mathbf{d}_{\mathcal{S}}(\bar{Z}_1, \bar{Z}_2) = \| \log_{\bar{Z}_1}(\bar{Z}_2) \|_{\bar{Z}_1}$, où $\|.\|_{\bar{Z}_1}$ dénote la norme induite par la métrique Riemannienne dans $T_{\bar{Z}_1}(\mathcal{S})$.

## 3.2 Codage de formes

Avant d'étudier la méthode de codage parcimonieux dans l'espace des formes, nous commençons par rappeler la formulation Euclidienne du problème.
Dans le cas d'un espace Euclidien, soit $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ un ensemble de vecteurs dans $\mathbb{R}^k$ qui dénote un dictionnaire de $N$ éléments ou atomes, et $z \in \mathbb{R}^k$ un point requête. Le problème de codage parcimonieux de $z$ par rapport à $\mathcal{D}$ peut être exprimé par,

$$l_E(z, \mathcal{D}) = \min_w \|z - \sum_{i=1}^{N} [w]_i \, d_i\|_2^2 + \lambda f(w), \quad (2)$$

où $w \in \mathbb{R}^N$ dénote le vecteur de codes constitué de $\{[w]_i\}_{i=1}^N$, $f : \mathbb{R}^N \to \mathbb{R}$ est la fonction génératrice de parcimonie définie par la norme $\ell_1$ et $\lambda$ le paramètre de régularisation de parcimonie. L'Eq. (2) tend à approximer $z$ de manière optimale (par $\hat{z}$) comme étant une combinaison linéaire d'atomes, i.e., $\hat{z} = \sum_{i=1}^N [w]_i \, d_i$, en prenant en compte d'une contrainte de parcimonie particulière sur les codes, $f(w) = \|w\|_1$. Cette fonction permet de représenter $z$ en utilisant uniquement un petit nombre d'atomes.
Dans le cas de l'espace des formes, $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ est maintenant un dictionnaire dans $\mathcal{S}$ et de même, la requête $\bar{Z}$ est un point dans $\mathcal{S}$. Par conséquent, le problème de codage parcimonieux inclus la distance géodésique définie dans $\mathcal{S}$ et devient ainsi

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_w (d_{\mathcal{S}}(\bar{Z}, C(\mathcal{D}, w))^2 + \lambda f(w)), \quad (3)$$

où $C : \mathcal{S}^N \times \mathbb{R}^N \to \mathcal{S}$ dénote une fonction de codage qui génère le point approximé $\hat{\bar{Z}}$ dans $\mathcal{S}$ en tant qu'une combinaison d'atomes et de codes. Noter que dans le cas particulier d'un espace Euclidien, $C(\mathcal{D}, w)$ serait une combinaison linéaire d'atomes. Cependant, dans la variété Riemannienne $\mathcal{S}$, la structure d'un espace vectoriel n'est plus valable ce qui fait que la combinaison linéaire d'atomes dans $\mathcal{S}$ n'est plus applicable, vu que le point approximé $\hat{\bar{Z}}$ peut se retrouver en dehors de la variété. Une alternative intéressante serait la formulation intrinsèque de l'Eq. (3) en considérant que $\mathcal{S}$ est une variété Riemannienne complète, ainsi la distance géodésique $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$ (comme expliqué dans la section 3.1). Par conséquent, la fonction-coût dans (3) peut être écrite comme

$$l_{\mathcal{S}}(\bar{Z}, \mathcal{D}) = \min_w \| \sum_{i=1}^{N} [w]_i \log_{\bar{Z}}(\bar{d}_i) \|_{\bar{Z}}^2 + \lambda f(w), \quad (4)$$

où $\log_{\bar{Z}}$ dénote l'opérateur de projection logarithmique qui projette chaque atome $\bar{d} \in \mathcal{S}$ dans l'espace tangent $T_{\bar{Z}}(\mathcal{S})$ au point $\bar{Z}$ à coder et $\|.\|_{\bar{Z}}$ est la norme induite par la métrique Riemannienne dans $T_{\bar{Z}}(\mathcal{S})$. Mathématiquement, ceci permet de compenser partiellement le manque de structure vectorielle dans $\mathcal{S}$, comme illustré dans la Fig. 1. Pour éviter la solution $w = 0$, nous avons imposé dans l'Eq. (4) une contrainte affine importante définie par $\sum_{i=1}^N [w]_i = 1$.

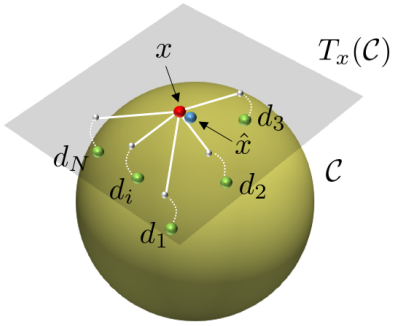


FIGURE 1 – Illustration de l'approche de codage parcimonieux dans l'espace de pré-formes $\mathcal{C}$.

## 3.3 Apprentissage de dictionnaires

Apprendre un dictionnaire discriminatif $\mathcal{D}$ implique une reconstruction précise des échantillons d'apprentissage et produit des codes discriminants avec la structure parcimonieuse souhaitée. Dans cette section, nous proposons d'apprendre $\mathcal{D}$ en respectant la géométrie de $\mathcal{S}$ et en se basant sur la méthode de codage parcimonieux décrite ci-dessus.
Tout d'abord, nous commençons par rappeler la formulation du problème dans le cas Euclidien. Étant donné un ensemble fini d'échantillons d'apprentissage $\{z_1, z_2, ..., z_m\}$ dans $\mathbb{R}^k$, apprendre un dictionnaire Euclidien est défini comme étant la minimisation de la fonction-coût du codage pour tous les choix d'atomes et de codes :

$$l_E(\mathcal{D}) = \min_{\mathcal{D}, w} \sum_{i=1}^{m} \left\| z_i - \sum_{j=1}^{N} [w_i]_j d_j \right\|_2^2 + \lambda f(w_i). \quad (5)$$

Pour résoudre ce problème non-convexe, une approche communément utilisée serait d'alterner entre les deux ensembles de variables $\mathcal{D}$ et $w$, tels que : (1) "Minimiser sur $w$

tant que $\mathcal{D}$ est fixé" est un problème convexe (i.e., codage parcimonieux). (2) "Minimiser l'Eq. (5) sur $\mathcal{D}$ tant que $w$ est fixé" est également un problème convexe.
Dans le cas de l'espace des formes, $\mathcal{D} = \{\bar{d}_1, \bar{d}_2, ..., \bar{d}_N\}$ devient un dictionnaire dans $\mathcal{S}$ et $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_m\}$ est un ensemble d'apprentissage dans $\mathcal{S}$. Comme dans le cas du codage parcimonieux, nous introduisons dans l'Eq. (5) la distance géodésique : $d_{\mathcal{S}}(\bar{Z}, \bar{d}) = \| \log_{\bar{Z}}(\bar{d}) \|_{\bar{Z}}$. Par conséquent, le problème d'apprentissage du dictionnaire dans l'espace de Kendall est formulé comme suit :

$$\min_{\mathcal{D},w} \sum_{i=1}^{m} \left\| \sum_{j=1}^{N} [w_i]_j \log_{\bar{Z}_i} \bar{d}_j \right\|^2_{\bar{Z}_i} + \lambda f(w_i), \qquad (6)$$

avec la contrainte affine importante $\sum_{j=1}^{N} [w]_j = 1$. Comme dans le cas Euclidien, ce problème d'optimisation peut être résolu en alternant d'une manière itérative entre : le codage parcimonieux en fixant $\mathcal{D}$, et l'optimisation de $\mathcal{D}$ en fixant les codes.
Pour l'initialisation du dictionnaire et dans le but d'accélérer la convergence de son apprentissage, nous proposons de l'initialiser par une approche qui se résume en deux étapes : (1) Regroupement des formes squelettiques tel que le nombre de clusters est déterminé automatiquement ; (2) Générer des atomes de chaque cluster de manière à ce qu'ils décrivent bien la variabilité dans le cluster. Pour plus de détails sur cette étape d'initialisation, nous invitons le lecteur à consulter [3]. Les étapes de l'apprentissage du dictionnaire dans l'espace de Kendall sont résumées dans l'algorithme 1.

**Algorithm 1** Apprentissage de dictionnaire dans $\mathcal{S}$

**Entrée :** Ensemble d'apprentissage $\mathcal{Z} = \{\bar{Z}_i\}_{i=1}^{m}$, où $\bar{Z}_i \in \mathcal{S}$ ; *nIter* : nombre d'itérations
**Sortie :** Dictionnaire $\mathcal{D} = \{\bar{d}_j\}_{j=1}^{N}, \bar{d}_j \in \mathcal{S}$

1: Initialisation du dictionnaire
2: **for** $k = 1$ to *nIter* **do**
3: Codage parcimonieux dans $\mathcal{S}$ en fixant $\mathcal{D}$, $\{w_i^*\}_{i=1}^{m}$ sont les codes.
4: Mise à jour des atomes en utilisant un algorithme de *recherche linéaire* pour résoudre l'Eq. (6) quand $\{w_i^*\}_{i=1}^{m}$ est fixé.
5: **end for**

# 4 Modélisation temporelle et classification

Soit $\{\bar{Z}_1, \bar{Z}_2, ..., \bar{Z}_L\}$ une séquence de squelettes qui représente une trajectoire dans $\mathcal{S}$. Comme il a été décrit dans la section 3, chaque squelette $\bar{Z}_i$ est codé en un vecteur de codes parcimonieux $w_i \in \mathbb{R}^N$ à l'aide d'un dictionnaire $\mathcal{D}$ (nous donnons à $\mathcal{D}$ une structure particulière qui sera décrite par la suite). Par conséquent, chaque trajectoire est transformée en une fonction de codes parcimonieux de dimension $N$. Le problème de classification des trajectoires dans $\mathcal{S}$ revient ainsi à classifier les fonctions de codes dans un espace Euclidien où toute opération classique sur des séries temporelles Euclidiennes (e.g., techniques d'apprentissage automatiques standards) peut être directement appliquée. Plusieurs méthodes dans la littérature ont traité et classifié des séries temporelles [1, 2, 30, 31]. Dans notre travail, nous adoptons deux schémas de classification différents : (1) Le premier, qui est utilisé dans [30] applique l'algorithme Dynamic Time Warping (DTW), une pyramide temporelle de Fourier (FTP) et un SVM linéaire de type "one-vs-all". Ceci permet de gérer la variation temporelle, gérer le bruit et classifier les descripteurs finaux, respectivement. (2) Long short-term memory (LSTM) [7], qui est une variante des réseaux de neurones récurrents et qui a l'avantage d'apprendre les dépendances temporelles à long-terme. De plus, nous avons exploré l'utilisation d'un LSTM bi-directionnel (Bi-LSTM) qui est une extension de LSTM traitant chaque séquence de l'avant et l'arrière dans deux réseaux de neurones séparés, ce qui fournit un contexte futur et passé, respectivement [12].
**Structure du dictionnaire** : Dans le contexte de la classification, on peut exploiter l'importante information sur les labels pour construire des descripteurs plus pertinents. Pour se faire, nous proposons de construire des dictionnaires par classes, tel qu'il a été fait dans [13]. Formellement, soit $S$ un ensemble de séquences labellisées dans $\mathcal{S}$ qui appartiennent à $q$ classes différentes $\{c_1, c_2, ..., c_q\}$. Notre objectif est de construire $q$ dictionnaires $\{D_1, D_2, ..., D_q\}$ dans $\mathcal{S}$ tel que chaque $D_j$ est appris en utilisant les squelettes de la même classe $c_j$. Dans ce scénario, le codage d'une squelette $\bar{Z} \in \mathcal{S}$ est effectué par rapport à chaque $D_{j,1 \leq j \leq q}$, indépendamment. Ainsi, $q$ vecteurs de codes sont obtenus. Ces vecteurs sont finalement concaténés pour former un vecteur descripteur global $W$. Comme il a été mentionné dans la section 5, cette structure génère des descripteurs plus discriminants pour la classification d'actions.

# 5 Expérimentations

Afin de valider notre méthode, une évaluation a été menée sur trois bases de données qui représentent des *challenges* différents, à savoir, Florence3D-Action [27], UTKinect-Action [36] et MSR-Action 3D [23].
**Florence3D-Action** [27] est constitué de 9 actions effectuées par 10 sujets différents. Chaque sujet effectue l'action 2 ou 3 fois pour un total de 215 séquences d'actions.
**UTKinect-Action** [36] comporte 10 actions effectuées deux fois par 10 sujets pour un total de 199 séquences.
**MSR-Action 3D** [23] comporte 20 actions effectuées 2 ou 3 fois par 10 sujets pour un total de 557 séquences.

## 5.1 Protocoles et paramètres expérimentaux

Pour les trois bases de données, nous avons utilisé les paramètres expérimentaux de [33], où une moitié des sujets est utilisée pour l'apprentissage et la deuxième moitié servira pour la phase de test. Chaque résultat affiché représente la moyenne des taux de classification correspondant

à 10 combinaisons différentes de sujets. Pour Florence3D, nous avons utilisé un deuxième protocole où un seul sujet est utilisé pour le test et le modèle est appris par le reste [27, 32]. Pour UTKinect, nous avons appliqué un second protocole où une seule action est utilisée pour le test et le reste sert pour l'apprentissage [36]. Pour MSR-Action3D, nous avons aussi appliqué le protocole de [23] qui divise la base en 3 groupes d'actions : AS1, AS2 et AS3 où chacun contient 8 actions. Dans toutes les expériences, la reconnaissance est effectuée en appliquant les deux classifieurs décrits dans la section 4. Dans toutes les expériences, nous avons utilisé une FTP de six niveaux et la valeur du paramètre C de SVM a été fixée à 1. Dans le deuxième schéma de classification, le réseau est entraîné avec une couche Bi-LSTM et la minimisation est effectuée en utilisant *Adam*. En raison des variations du nombre des joints et la longueur des séquences pour les différentes bases de données, la valeur de la taille des neurones a été choisie par validation croisée pour chaque base.

## 5.2 Résultats et discussions

**Comparaison aux méthodes Riemanniennes existantes.** Les taux de reconnaissance obtenus par les différentes méthodes sont affichés dans le Tableau 1. Conformément aux autres méthodes, nous avons appliqué le protocole expérimental de [33] pour les trois bases de données, en plus du protocole de [23] pour MSR-Action3D. A partir du tableau, nous remarquons que nos résultats dépassent ceux des autres méthodes Riemanniennes pour les trois bases de données. Nous rappelons que ces méthodes présentent une limite commune qui est celle de projeter toutes les trajectoires de la variété vers un espace tangent commun, ce qui engendre des distorsions. Par contre, notre méthode évite ce problème non-trivial vu que chaque point de la variété est codé sur son propre espace tangent. Dans un premier temps, nous discutons les résultats que nous avons obtenus avec le premier schéma de classification qui est également utilisé dans [1, 30, 31]. Pour les trois bases de données, il est clair que notre approche dépasse les autres en utilisant le même classifieur, ce qui montre l'apport de notre représentation squelettique. Par exemple, nous marquons une amélioration de $1.73\%$ sur MSR-Action 3D et $1.45\%$ sur Florence3D.

Nous analysons maintenant les résultats obtenus par le second classifieur (Bi-LSTM). Dans ce cas, notons que même si aucun pré-traitement n'a été effectué sur les séquences de codes, notre approche dépasse les autres de $1.64\%$ sur Florence3D. Cependant, elle est moins bonne sur UTKinect où on affiche un taux de reconnaissance de $96.89\%$ comparé au meilleur résultat de $97.08\%$ obtenu par [30]. Sur MSR-Action3D, notre approche est plus performante que celle de [1] en utilisant le premier protocole. Notons que dans [1], les résultats ont été moyennés sur toutes les 242 combinaisons possibles. Par contre, le taux de reconnaissance que nous avons obtenu est inférieur à celui des autres approches en utilisant les deux protocoles sur cette base. (environ $3.5\%$ pour le premier et $0.62\%$ pour le second). Ici, il est important de mentionner que plusieurs séquences présentes dans MSR-Action3D sont bruitées [26]. Par conséquent, utiliser Bi-LSTM sans aucun pré-traitement qui gère le bruit (telle que FTP) ne suffit pas pour atteindre les résultats de l'état de l'art sur cette base.

TABLE 1 – Comparaison aux méthodes Riemanniennes.

| Méthode | MSR3D[1] | Florence | UTK | MSR3D[2] |
|---|---|---|---|---|
| T-SRVF Lie group [1] | 85.16 | 89.67 | 94.87 | – |
| T-SRVF on $\mathcal{S}$ [2] | 89.9 | – | – | – |
| Lie Group [30] | 89.48 | 90.8 | 97.08 | 92.46 |
| Rolling rotations [31] | – | 91.4 | – | – |
| **Ours** (FTP-SVM) | **90.01** | 92.85 | **97.39** | **94.19** |
| **Ours** (Bi-LSTM) | 86.18 | **93.04** | 96.89 | 91.84 |

[1] Protocole de [34].
[2] Protocole de [23].

**Comparaison aux méthodes récentes.** Dans ce paragraphe, nous discutons nos résultats par rapport aux méthodes non-Riemanniennes récentes. Pour toutes les bases, nos résultats sont compétitifs.

**Florence3D-Action** – Le Tableau 2 montre que notre approche dépasse toutes les autres en utilisant le protocole de [27, 32]. Cependant, notre résultat est inférieur à [21] de $2.19\%$. Dans [21], les auteurs combine deux représentations de noyau : SCK et DCK qui ont atteint, séparément, $92.98\%$ et $92.77\%$, respectivement. Notre approche a réussi à bien classifier la plupart des actions. Toutefois, les confusions majeures correspondent à des actions très similaires telles que : *répondre au téléphone* et *boire un verre*.

TABLE 2 – Florence3D : comparaison avec l'état de l'art.

| Méthode | prot. de [27, 32] | prot. de [33] |
|---|---|---|
| Graph-based [35] | 91.63 | – |
| T-Forest [11] | 94.16 | – |
| SCK+DCK [21] | – | **95.23** |
| **Ours** (FTP-SVM) | 92.27 | 92.85 |
| **Ours** (Bi-LSTM) | **94.48** | 93.04 |

**UTKinect** – Les résultats sont affichés dans le Tableau 3. En utilisant le protocole de [36], notre approche obtient le meilleur taux de reconnaissance avec chacun des deux classifieurs marquant ainsi une amélioration de $2.49\%$ comparée à la méthode de [24], qui est basée sur une version étendue de LSTM. Pour le protocole de [33], notre meilleur résultat est compétitif à celui de $98.2\%$ obtenu dans [21]. En prenant en compte le *challenge* majeur de cette base de données qui est la variation du point de vue, notre approche prouve l'importance de considérer les propriétés d'invariance offertes par la représentation des squelettes dans l'espace des formes de Kendall.

**MSR-Action 3D** – Le Tableau 4 montre une comparaison des résultats. En appliquant le protocole de [23], notre meilleur résultat est compétitif à celui des approches récentes. En particulier, pour AS3, nous soulignons un taux de reconnaissance de $100\%$. Ce résultat montre l'efficacité

TABLE 3 – UTKinect : comparaison avec l'état de l'art.

| Méthode | prot. de [36] | prot. de [33] |
|---|---|---|
| ST-LSTM [24] | 97.0 | 95.0 |
| JLd+RNN [38] | – | 95.96 |
| Graph-based [35] | – | 97.44 |
| SCK+DCK [21] | – | **98.2** |
| **Ours** (FTP-SVM) | 97.50 | 97.39 |
| **Ours** (Bi-LSTM) | **98.49** | 96.89 |

de notre approche dans la reconnaissance des actions complexes, vu que celles-ci sont regroupées dans AS3. Pour AS1, nous avons atteint l'un des meilleurs taux de reconnaissance ($95.87\%$). Cependant, le résultat obtenu sur AS2 est de $8.9\%$ inférieur au meilleur résultat de l'état de l'art. Ceci montre que notre méthode est moins efficace quand il s'agit de reconnaître des actions similaires, vu que AS2 regroupe des actions similaires ensemble. Même si notre meilleur résultat est légèrement supérieur à celui de [21], il est inférieur à celui de la même méthode quand nous suivons le protocole de [34], ce qui montre que notre approche est plus performante quand le problème contient moins de classes.

TABLE 4 – MSR-Action : comparaison avec l'état de l'art.

| Méthode | AS1 | AS2 | AS3 | Moy | Prot. [34] |
|---|---|---|---|---|---|
| SCK+DCK [21] | – | – | – | 93.96 | **91.45** |
| HBRNN-L [9] | 93.33 | **94.64** | 95.50 | 94.49 | – |
| T-Forest [11] | **96.10** | 90.54 | 97.06 | **94.57** | – |
| **Ours** (FTP-SVM) | 95.87 | 86.72 | **100** | 94.19 | 90.01 |
| **Ours** (Bi-LSTM) | 92.72 | 84.93 | 97.89 | 91.84 | 86.18 |

**Expériences supplémentaires.** Pour évaluer quelques propriétés de notre approche, nous avons effectué les expériences suivantes.

**Parcimonie** – Dans cette expérience, nous évaluons l'effet du paramètre $\lambda$ sur les taux de reconnaissance obtenus en utilisant chacun des deux classifieurs. Pour se faire, une moitié de l'ensemble de l'apprentissage a été utilisé pour apprendre le dictionnaire et entraîner les classifieurs et la deuxième moitié a été utilisée pour la validation de l'apprentissage. Le premier graphe de la Fig. 2 montre l'effet d'augmenter la valeur de $\lambda$ de $10^{-4}$ à $1$ avec un pas de $10^{-2}$. De plus, nous affichons le pourcentage de parcimonie (i.e., nombre de codes non-nuls divisé par le nombre total de codes) pour quelques valeurs de $\lambda$ pour montrer la cohérence des codes obtenues avec la théorie. Nous remarquons que le pourcentage de parcimonie augmente quand la valeur de $\lambda$ augmente. On remarque que le taux de reconnaissance atteint un maximum pour les valeurs de $\lambda = 0.01$ ($37\%$ de parcimonie) et $\lambda = 0.02$ ($49\%$ de parcimonie) pour SVM et Bi-LSTM, respectivement. Notons que dans toutes les expériences précédentes, la valeur de $\lambda$ a été choisie empiriquement de manière à ce qu'elle correspond à ces derniers taux de parcimonie.

**Structure du dictionnaire** – Comme il a été décrit dans la section 4, nous construisons des dictionnaires par classes. Pour évaluer la pertinence de cette structure dans le contexte de la classification, nous la comparons à la structure classique, qui consiste en un dictionnaire global qui ne prend pas en compte les labels. Les taux de reconnaissance obtenus sont respectivement $94.48\%$ et $91.53\%$. La structure adoptée permet donc de mieux classifier les actions.

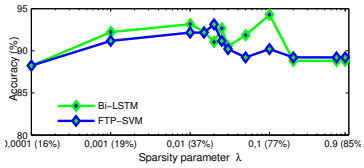


FIGURE 2 – Taux de reconnaissance en fonction du paramètre de régularisation $\lambda$ (les valeurs % dans l'axe des abscisses représentent les pourcentages de parcimonie).

# 6 Conclusion

Dans cet article, nous avons représenté un squelette 3D comme étant un point dans l'espace de Kendall, ainsi une action comme une trajectoire dans le même espace, dans le but d'exploiter des propriétés d'invariance importantes pour l'analyse de formes. Pour gérer la non-linéarité inhérente de cette variété, nous avons proposé de coder chaque forme squelettique sur l'espace tangent qui lui est attaché en se basant sur un dictionnaire qu'on apprend, évitant ainsi le problème de projeter tous les points dans un seul espace tangent attaché à la variété à un point de référence. L'apprentissage du dictionnaire s'est fait en respectant la géométrie de l'espace des formes. Notre approche de codage parcimonieux a permis de transformer les trajectoires initiales en des fonctions de codes parcimonieux permettant de les classifier dans un espace vectoriel. Nous avons appliqué notre approche au problème de la reconnaissance d'actions 3D en utilisant deux classifieurs différents obtenant des résultats compétitifs par rapport à l'état de l'art.

# Références


[1] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. Elastic functional coding of human actions : From vector-fields to latent variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] B. Ben Amor, J. Su, and A. Srivastava. Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1) :1–13, 2016.

[3] A. Ben Tanfous, H. Drira, and B. Ben Amor. Coding Kendall's Shape Trajectories for 3D Action Recognition. In *IEEE Computer Vision and Pattern Recognition*, Salt Lake City, United States, June 2018.

[4] D. Bryner, E. Klassen, H. Le, and A. Srivastava. 2d affine and projective shape analysis. *IEEE transactions on pattern analysis and machine intelligence*, 36(5) :998–1011, 2014.

[5] H. E. Cetingül and R. Vidal. Sparse riemannian manifold clustering for hardi segmentation. In *Biomedical Imaging : From Nano to Macro, 2011 IEEE International Symposium on*, pages 1750–1753. IEEE, 2011.

[6] A. Cherian and S. Sra. Riemannian dictionary learning and sparse coding for positive definite matrices. *IEEE Transactions on Neural Networks and Learning Systems*, PP, 2017.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] I. Dryden and K. Mardia. *Statistical shape analysis*. Wiley, 1998.

[9] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, June 2015.

[10] A. Efros and A. Torralba. Guest editorial : Big data. *International Journal of Computer Vision*, 119(1) :1–2, 2016.

[11] G. Garcia-Hernando and T. Kim. Transition forests : Learning discriminative temporal transitions for action recognition. *CoRR*, abs/1607.02737, 2016.

[12] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5) :602–610, 2005.

[13] T. Guha and R. K. Ward. Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8), 2012.

[14] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3) :113–136, 2015.

[15] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning : Kernels to the rescue. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3926–3935, June 2015.

[16] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson. Sparse coding on symmetric positive definite manifolds using bregman divergences. *IEEE transactions on neural networks and learning systems*, 27(6) :1294–1306, 2016.

[17] M. T. Harandi, C. Sanderson, R. I. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices : A kernel approach. *CoRR*, abs/1304.4344, 2013.

[18] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *International conference on machine learning*, pages 1480–1488, 2013.

[19] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on lie groups for skeleton-based action recognition. *arXiv preprint arXiv :1612.05877*, 2016.

[20] D. G. Kendall. Shape manifolds, Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2) :81–121, 1984.

[21] P. Koniusz, A. Cherian, and F. Porikli. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, pages 37–53. Springer, 2016.

[22] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *2013 IEEE International Conference on Computer Vision*.

[23] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *IEEE Inter. Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB)*.

[24] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*. Springer, 2016.

[25] Y. M. Lui. Advances in matrix manifolds for computer vision. *Image Vision Comput.*, 30(6-7) :380–388, June 2012.

[26] L. L. Presti and M. L. Cascia. 3d skeleton-based human action classification : A survey. *Pattern Recognition*, 53(Supplement C) :130 – 147, 2016.

[27] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*.

[28] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1297–1304, 2011.

[29] J. Su, S. Kurtek, E. Klassen, and A. Srivastava. Statistical analysis of trajectories on riemannian manifolds : Bird migration, hurricane tracking, and video surveillance. *Annals of Applied Statistics*, 2013.

[30] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[31] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4471–4479, 2016.

[32] C. Wang, Y. Wang, and A. L. Yuille. Mining 3d key-pose-motifs for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2639–2647, 2016.

[33] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3D action recognition with random occupancy patterns. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, pages 872–885, 2012.

[34] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1297, 2012.

[35] P. Wang, C. Yuan, W. Hu, B. Li, and Y. Zhang. Graph based skeleton motion representation and similarity measurement for action recognition. In *European Conference on Computer Vision*. Springer, 2016.

[36] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012.

[37] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo. *Human Action Recognition under Log-Euclidean Riemannian Metric*, pages 343–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[38] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157, March 2017.

[39] H. E. Çetingül, M. J. Wright, P. M. Thompson, and R. Vidal. Segmentation of high angular resolution diffusion mri using sparse riemannian manifold clustering. *IEEE Transactions on Medical Imaging*, 33(2) :301–317, Feb 2014.